





**WebSphere Adapter for Siebel Business Applications User Guide
Version 6 Release 2**

Note

Before using this information and the product it supports, read the information in "Notices" on page 99.

December 2008

This edition applies to version 6, release 2, modification 0 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 International Business Machines Corporation 2006, 2008. All rights reserved. US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

© **Copyright International Business Machines Corporation 2006, 2008.**

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 of the Adapter for Siebel Business Applications	3
Outbound processing	3
Inbound processing	7
Business objects	10
Java data bindings	11
J2C Bean wizard	11
Standards Compliance	12

Chapter 2. Planning for adapter implementation 13

Before you begin	13
Security	13
Support for protecting sensitive user data in log and trace files	13
User authentication	14
Deployment options	15
WebSphere Adapters in clustered environments	17

Chapter 3. Configuring the module for deployment. 19

Configuring the Siebel application to work with the adapter	19
Launching the J2C Bean wizard	21
Configuring the connector dependencies.	22
Configuring the module for outbound processing	23
Setting connection properties for the J2C Bean wizard	23
Selecting business objects and services	24
Configuring the selected objects	26
Setting deployment properties and generating artifacts.	27
Generating the EJB or JSP project	29
Configuring the module for inbound processing	31
Setting connection properties for the J2C Bean wizard	31
Selecting business objects and services	32
Configuring the selected objects	34
Setting deployment properties and generating artifacts.	35

Chapter 4. Configuring the application on WebSphere Application Server . . . 37

Changing configuration properties for embedded adapters	37
Setting resource adapter properties for embedded adapters	37
Setting managed (J2C) connection factory properties for embedded adapters	38

Setting activation specification properties for embedded adapters.	39
Changing configuration properties for stand-alone adapters	40
Setting resource adapter properties for stand-alone adapters	40
Setting managed (J2C) connection factory properties for stand-alone adapters	41
Setting activation specification properties for stand-alone adapters	42
Adding dependency libraries to the deployed resource adapter.	43
Standalone deployment	43
EAR deployment	43

Chapter 5. Troubleshooting and support 45

Support for the Log and Trace Analyzer.	45
Configuring logging and tracing	45
Configuring logging properties	46
Changing the log and trace file names	47
First-failure data capture (FFDC) support	48
Solutions to some common problems.	49
XAResourceNotAvailableException	51
Self-help resources	52

Chapter 6. Reference information . . . 53

Business object information	53
Application-specific information	53
Supported operations	55
Naming conventions	55
Outbound configuration properties	58
Connection properties for the wizard.	60
Resource adapter properties	66
Managed connection factory properties	71
Interaction specification properties.	75
Inbound configuration properties	76
Connection properties for the wizard.	77
Resource adapter properties	84
Activation specification properties.	88
Globalization	96
Globalization and bidirectional data transformation	96
Properties enabled for bidirectional data transformation	97

Notices 99

Programming interface information	101
Trademarks and service marks	101

Index 103

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 application, without special coding.

WebSphere Adapter for Siebel Business Applications provides a way to create integrated processes that exchange information with a Siebel application. With the adapter, an application can send requests to the Siebel Business Applications server or receive notifications of changes from the server.

The adapter creates a standard interface to the applications and data on the Siebel Business Applications server, so that the application does not have to understand the lower-level details (the implementation of the application or the data structures) on the Siebel Business Applications server. An application, for example, can send a request to the Siebel Business Applications 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, to be notified that a customer record has been updated. This provides you with improved business workflow and processes to help manage your customer relations.

WebSphere Adapter for Siebel Business Applications complies with the Java™ 2 Platform, Enterprise Edition (J2EE) Connector Architecture (JCA). JCA standardizes the way application components, application servers, and Siebel applications, such as Siebel Business Applications server, interact with each other. WebSphere Adapter for Siebel Business Applications makes it possible for JCA-compliant application servers to connect to and interact with the Siebel server in a standard way by using JavaBeans™.

The following example assumes 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, needs to first obtain a list of customers.

The Siebel Business Applications function does not have a Web service interface, however, so the application component used by the promotions department needs to understand the low-level API and data structures of the Siebel Business Applications function in order to make the call to the function. Information technology resources are needed to create the links between the component and the Siebel Business Applications function.

With WebSphere Adapter for Siebel Business Applications, 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 either 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 does not have to deal directly with the Siebel Business Applications function; it is the Siebel Business Applications 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 Adapter for Siebel Business Applications. In return, the application component receives the results (the list of customers) in the form of a standard business object. The application component does not need to 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 notifies 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 Web site.

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 and click **Compatibility Matrix** beneath the **Related** heading in the **Additional support links** section: <http://www.ibm.com/software/integration/wbiadapters/support/>.
- Technotes for WebSphere Adapters provide workarounds 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 of the Adapter for Siebel Business Applications

WebSphere Adapter for Siebel Business Applications supports the exchange of information between your existing 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 application), and inbound processing (event notification from a Siebel application server to an application).

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

Adapter for Siebel Business Applications is metadata driven, which means that 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 new business objects without additional coding or customization.

Outbound processing

WebSphere Adapter for Siebel Business Applications supports synchronous outbound processing. This means that when 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 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 perform 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, Delete), the adapter populates that Siebel object (business or integration component hierarchy) with data from the hierarchy of WebSphere business objects.

Supported outbound operations

WebSphere Adapter for Siebel Business Applications supports the following outbound operations: Apply Changes, 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
ApplyChanges	Updates the business component based on the delta information.
Create	Creates the business component.
Delete	Deletes the business component and its children.
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 need to invoke operations such as data retrieval from a database. An operation is the action that an adapter can perform on the database during outbound processing. The name of the operation indicates the type of action that the adapter takes.

The 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 children is created first and then the parent is created. After all of the children are created, the parent is saved. By saving the parent after creating all the children, you can roll back all of the children 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. Siebel 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 new child business object. Instead, set the value on the parent. For example, if you pick an existing

Business Address instance under Account, set the PrimaryAddressId instead of creating a Business Address instance. However, to create a new child instance, you must populate the non-key attributes in the child business component so that the adapter will try 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 Picklist's list of values, the adapter tries to populate a value. If the list is unbounded and Insert is enabled, the operation will be successful. If the adapter is unable to populate the value by itself, the operation will fail.
- With multi value links, the child record is created. If the child Create operation fails, the child is rolled back. If the parent Create operation fails, all of the children and the parent are rolled back.
- The adapter attempts to select an existing child business component instance (for example, Association) based on the value of all of 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 is rolled back. If the parent Create operation fails, only the parent is rolled back. Because association components can also be individual components, as well as children of other components, a child or parent cannot be rolled back once an associated child has been created.

Delete operation:

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

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 will fail.

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 children 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. Before the adapter retrieves the child, the From attribute value from the parent must be set against the To attribute in the child. If the From attribute is not set, the adapter retrieves the child without setting any values on the To attribute. For example, only executeQuery is called.
- With multi-value link and association, if the child record is found, the attribute value is set on the parent.

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 have key and non-key attributes set. All of 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, it will be a 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 will be applied to the operation if it has already been set.

The steps involved in processing the RetrieveAll operation is similar to the Retrieve operation, except the adapter does not check to make sure that all of the keys have been 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 no records were 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 children will be processed and updated. If the child Create operation fails because the record already 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. This means that the adapter polls the Siebel Business 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 for the Siebel event business component or integration component 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. To keep track of inbound events as they make their way through the system, the adapter uses an event store.

The creation, update, or deletion of an event record in the Siebel business application 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 continually 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 just been added to it, this signals an update. If the adapter is configured to receive the events about the new update, there will be 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 was 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. Do this in the `ActivationSpec` 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 of 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 will hold 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. This column 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 will be auto-generated by Siebel.	
Event timestamp	The time stamp for the event. The format is <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 will remove the dependency on the EDT. For more information on XID, check the AFC FS.	
CONNECTOR ID	The ID set for the Siebel 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 will hold 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 'Acocunt (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. 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. This column 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 will be 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 will remove the dependency on the EDT. For more information on XID, check the AFC FS.	
CONNECTOR ID	The ID set for the Siebel adapter	Siebel001

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 or an employee record, or 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 children of it. Information about the processed object is stored in the application-specific information for the object and each of its attributes.

Java data bindings

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

A java data binding 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 java data bindings by using the J2C Bean wizard, launched from Rational Application Developer Connector Tools. The wizard connects to the Siebel Business Applications server, discovers data structures in the EIS, and generates data bindings 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.

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 Java data bindings.

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, including the objects associated with any faults that might be generated during processing.

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 utilizing standard text editors and scripted 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 web site <http://www.ibm.com/able/> for more information about the commitment that IBM has to accessibility.

Internet Protocol, Version 6 (IPv6)

IBM 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>.

Chapter 2. Planning for adapter implementation

Before you configure WebSphere Adapter for Siebel Business Applications Software, consider whether you will set up the adapters in a clustered environment, in which the workload of the server is distributed across multiple machines. Also, if you are migrating from an earlier version of WebSphere Adapter for Siebel Business Applications Software, perform any migration tasks.

Before you begin

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

To configure and use WebSphere Adapter for Siebel Business Applications, you should 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.
- The capabilities provided by the integration development tools you will use to build the solution. You should 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 will use for the integration solution. You should 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 should 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 authentication alias, feature of Java 2 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

The adapter provides the ability to prevent sensitive or confidential data in log and trace files from being seen by those 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 must 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 situations like this, the adapter lets you specify whether or not 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 71 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 the 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 the Siebel application to process outbound requests and inbound events.

Authentication in the wizard

The J2C Bean wizard asks for 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 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 lets you 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 completes.

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. The J2C Bean wizard lets you configure the adapter to get the user information using 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 as well as 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 2 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 can 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 will affect the type of deployment option you choose.

The deployment options are described below:

- 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.

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 will affect how the adapter is used in the runtime environment, as well as 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 on running 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 (product number 5724-T74). This provides the RAR file that you install following instructions supplied with WebSphere Application Server.

Considerations for embedding an adapter in the application

Take into consideration the following items if you plan on embedding 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 runtime environments. *Class loader isolation* means 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

Take into consideration the following items if you plan on using 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 given 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 given adapter share the same 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.

WebSphere Adapters in clustered environments

You can improve adapter performance and availability by deploying the module to a clustered server environment. The module is replicated across all servers in a cluster, regardless of whether you deploy the module using a stand-alone or embedded adapter.

WebSphere Application Server, WebSphere Application Server Network Deployment, and WebSphere Extended Deployment support clustered environments. Clusters are groups of servers that are managed together to balance workloads and to provide high availability and scalability. When you set up a server cluster, you create a Deployment Manager profile. The HAManager, a subcomponent of the Deployment Manager, notifies the Java 2 Platform, Enterprise Edition (J2EE) Connector Architecture (JCA) container to activate the adapter instance. The JCA container provides a runtime environment for adapter instances.

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 instead of a static workload manager, which is used by WebSphere Application Server Network Deployment. The dynamic workload manager can optimize the performance of adapter instances in the cluster by dynamically balancing the load of the requests. This means that application server instances can be automatically stopped and started based on the load variations, allowing machines with different capacities and configurations to evenly handle load variations.

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 2 Platform, Enterprise Edition (J2EE) 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.

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. Accordingly, 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, as long as 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 those 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 to retrieve Siebel business objects and services.

Also, because you will perform this task on the Siebel 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 will need to open Siebel Sales Enterprise (or the Siebel application you are using) on your local database, and to do this, you must have administrative privileges.

About this task

To configure the Siebel application, you will create an event table and 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 in which 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

Table 4. Event table details (continued)

Column Name	Type	Length	Data Type	Required	Nullable	Status
STATUS	Data (public)	20	Varchar	Yes	No	Active
XID	Data (public)	255	Varchar	Yes	No	Active

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

- b. Create a new business component called IBM Event.
 - c. Create a new 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 new 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 new 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_Q' 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 new 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 new 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 new 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 on 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 server, and replace the original repository file with the newly created one.
14. Restart the Siebel 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

Make sure 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 needed 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**.
2. Click **Next**.
3. 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, for example, 6.2.0.0
4. Click **Next**.
5. In the window, accept the default project name in the **Connector project** field or type a different name.
6. In the **Target server** field, select the type of server where you will deploy the module. The wizard creates the artifacts that are appropriate to that server.

7. Click **Next**. The Connector Settings window is displayed.

Results

A new connector project is created, which contains the adapter RAR file. The project is listed in the Enterprise Explorer view of the Java EE perspective.

What to do next

Continue working in the J2C Bean wizard. The next step is to add database-specific files to the project.

Configuring the connector dependencies

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

Before you begin

You should 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 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. External software 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, and 8.0)
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. From the Connector Settings window in the wizard, specify the location of the files by clicking **Browse** for each property and selecting the location of the file. For example, if you selected the Siebel.jar file, then the location would be: C:\downloads\Siebel8\Siebel.jar.
4. Click **Next**.

Results

The wizard has the files it needs to communicate with the Siebel server.

What to do next

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

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 business object definitions and related artifacts.

Setting connection properties for the J2C Bean wizard

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

Before you begin

Make sure you have successfully added the external software dependency files.

About this task

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

To specify the connection properties, use the following procedure.

Procedure

1. On the Adapter Style window, select **Outbound** (to send data to the Siebel server from the adapter), and click **Next**.
2. In the Discovery Configuration window, specify the configuration properties:
 - a. In the **Connection URL** field, type the URL needed to connect to the Siebel server.
 - b. If necessary, change the default setting (ENU - English) for **Language code** by selecting another language from the drop down list.
 - c. Type the username and password you use to access the Siebel server. The password is case-sensitive.
 - d. If necessary, change the default setting for **Type of Siebel objects to discover** (Siebel Business Services) by selecting Siebel Business Objects from the drop down list.
 - e. If necessary, change the default setting for **Siebel repository name** by typing in the name of the repository from which the objects are to be discovered.
 - f. If necessary, type a prefix in the **Prefix for business object names** field to have it attached to the front of the business object name.
3. To set additional advanced properties, click **Advanced**. The related properties are displayed.
4. If you need to set bidirectional properties, perform the following steps:
 - a. Expand **Bidi properties** and select the **Bidi transformation** check box.

- b. Set the properties, ordering schema, text direction, symmetric swapping, character shaping, and numeric shaping to control how bidirectional transformation is performed.
5. To change the location of the wizard's log files or the amount of information included in the logs, click the **Specify the level of the logging desired** check box. Provide the following information:
 - a. In the **Log file output location** field, specify the location of the log file for the wizard. Click **Browse** if you want to select a different location.
 - b. In the **Logging Level** field, specify the severity of errors that you want logged. This log information is for the wizard only; at run time, the adapter writes the error messages and the trace information into the standard log and trace files for the server.

Note: This log pertains to the J2C Bean wizard only, not to the operation of the adapter.

6. Click **Next**.

Results

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

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

Selecting business objects and services

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

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

Selecting business objects

1. In the Discovery Configuration window, select **Siebel Business Objects** as the **Type of Siebel Objects to discover**. The check box **Prompt for additional configuration settings when adding a business object** is displayed.
2. Select the check box **Prompt for additional configuration settings when adding a business object**. You need to select this if you want to select the attributes of the business objects.
3. Click **Next**.
4. 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.
- 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 check box, **Prompt for additional configuration settings when adding a business object** 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. Also, if you add a primary business component, it is displayed in the **Objects to be imported** pane. If you add a dependent business component, the primary business component is also added but it is not displayed in the **Objects to be imported** pane. You can select more than one object at a time.

- h. Click **Next**.

Selecting business services

1. In the Discovery Configuration window, select **Siebel Business Services** from the **Type of Siebel Objects to discover** list. Click **Next**.
2. 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 Properties window is displayed.

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

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

Note:

- 1) When a business service has only 1 complex type Input/Output argument, the Configuration Properties 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: 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 so you do not need to 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 you do not need to select the **Integration Object** and it is not added as a business object.
 - 6) Siebel Enterprise Information System (EIS) has restriction that it is not possible to change the existing input and output argument names for any method of a standard business service or create new argument names as they are hard coded internally in the underlying class of the business service along with the logic. So make sure that you are working on a custom Siebel business service if you want to use custom Siebel message.
- i. Click **Next**.

Results

You have selected the Siebel business object or 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 should be stored.

Before you begin

Make sure 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 of 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** 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 could 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.
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 new Java 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 new project into which the J2C Bean is generated.
 - To select an existing project, click **Browse**. If the desired project name appears in the **Project Selection** list, select its name.
 - Otherwise, create a new project:
 - a. Click **New**.
 - b. In the New Source Project Creation window, select **Java project**.

- c. In the Create a new project window, type a name for the project. For example, MyApdapterOutbound.
 - d. Accept the default values for the other fields.
 - e. 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 desired package name appears in the **Project Selection** list, select its name.
 - Otherwise, create a new package:
 - a. Click **New**.
 - b. In the Java Package window, click **Java project**.
 - c. In the Create a new project window, type a name for the package. For example, myadapteroutboundpkg.
 - d. Accept the default values for the other fields.
 - e. 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: Select the **Enable generate Command Bean** check box, to generate a command bean for every operation that you have selected. 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.
 - To obtain the connection through JNDI, select the **Managed Connection (recommended)** check box. This type of connection is managed by the application server.
 - To obtain the connection directly from the resource adapter, select the **Non-managed Connection** check box.
6. If you select **Managed Connection (recommended)** check box, specify how you want the adapter to specify the connection properties.
 - To select an existing name, click **Browse**.
 - Otherwise, create a new name.
 - a. Click **New**.
 - b. In the Server selection window, specify the server on which to deploy the adapter and click **Next**.
 - c. In the New J2C Connection Factory window, specify the name in the **JNDI Name** field. For example, com/eis/AdapterCF.
 - d. Click **Finish**.
7. If you select the **Non-managed Connection** check box, the Database system connection information area is expanded to show the connection information. Review the connection information and change the values if required. You can change the values to specify a different user name and password at run time. You can also connect to an alternate database, although the schema names must be the same in both databases. The format of the connection properties is database-specific.
8. Optional: Specify advanced properties by clicking **Advanced**. Expand each of the advanced sections to review the properties.

9. When you are finished setting properties, 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 Java data bindings directly, in the non-managed mode or generate JSP or EJB that uses the J2C Bean. Rational Application Developer 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 documentation.

Generating the EJB or JSP project

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

About this task

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 Enter.
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** or **Sample JSP** and click **Next**.

Note: In the Deployment Information window, the **Check 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 EJB Creation wizard is displayed. This wizard creates the Java project as an EJB project.
 - To select an existing project, click **Browse**. If the desired 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 an EJB project name. For example, MyAdapterOutboundEJB.
 - c. In the EAR Membership area, click **New** to create a new ear project name.
 - d. In the EAR Application Project window, type an EAR project name. For example, MyAdapterOutboundEJB EAR.
 - e. Click **Next**.

- f. In the Enterprise Application window, select the Siebel adapter RAR dependency such as CWYEB_SiebelAdapter.
- g. Select the **Generate Deployment Descriptor** check box.
- h. Click **Finish** to return to the EJB Project window.
- i. Click **Finish** to return to the EJB Creation window.
- j. Click **Next**.

Note: In the EJB Creation window, the **Next** button is available for selection only if you have selected the **Configure Resource Adapter Deployment** check box when specifying the deployment information. After you create the EJB and EAR project and return to EJB Creation window, you can directly click **Finish** to complete the process. The adapter is deployed as stand alone by default.

- k. In Resource Adapter Deployment window, select **Deploy within EAR** and select the EJB EAR Project you created.
 - l. Click **Finish**.
9. If you select **Sample JSP**, the Simple JSP Creation wizard is displayed. .
- To select an existing project, click **Browse**. If the desired project name appears in the Project Selection list, select its name.
 - Otherwise, create a new project.
 - a. Click **New**.
 - b. In Dynamic Web Project window, type a project name. For example, MyAdapterOutboundWEB.
 - c. In EAR Membership area, click **New** to create a new ear project name.
 - d. In EAR Application Project window, type an EAR project name. For example, MyAdapterOutboundWEBEAR.
 - e. Click **Next**.
 - f. In the Java EE Module dependencies area of the Enterprise Application window, select the Siebel adapter RAR dependency such as CWYEB_SiebelAdapter.
 - g. Select the **Generate Deployment Descriptor** check box.
 - h. Click **Finish** to return to the Dynamic Web Project window.
 - i. Click **Finish** to return to the Simple JSP Creation window.
 - j. Click **Next**.

Note: In the EJB Creation window, the **Next** button is available for selection only if you have selected the **Configure Resource Adapter Deployment** check box when specifying the deployment information. After you create the WEB and EAR project and return to Simple JSP Creation window, you can directly click **Finish** to complete the process. The adapter is deployed as stand alone by default.

- k. In Resource Adapter Deployment window, select **Deploy within EAR** and select the Web EAR Project you created.
 - l. Click **Finish**.
10. 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.

Setting connection properties for the J2C Bean wizard

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

Before you begin

Make sure you have successfully added the external software dependency files.

About this task

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

To specify the connection properties, use the following procedure.

Procedure

1. On the Adapter Style window, select **Inbound** (to send data to the Siebel server from the adapter), and click **Next**.
2. In the Discovery Configuration window, specify the configuration properties:
 - a. In the **Connection URL** field, type the URL needed to connect to the Siebel server.
 - b. If necessary, change the default setting (ENU - English) for **Language code** by selecting another language from the drop down list.
 - c. Type the user name and password you use to access the Siebel server. The password is case-sensitive.
 - d. If necessary, change the default setting for **Type of Siebel objects to discover** (Siebel Business Services) by selecting Siebel Business Objects from the drop down list.
 - e. If necessary, change the default setting for **Siebel repository name** by typing in the name of the repository from which the objects are to be discovered.
 - f. If necessary, type a prefix in the **Prefix for business object names** field to have it attached to the front of the business object name.
3. To set additional advanced properties, click **Advanced**. The related properties are displayed.
4. If you need to set bidirectional properties, perform the following steps:
 - a. Expand **Bidi properties** and select the **Bidi transformation** check box.
 - b. Set the properties, ordering schema, text direction, symmetric swapping, character shaping, and numeric shaping to control how bidirectional transformation is performed.
5. To change the location of the wizard's log files or the amount of information included in the logs, click the **Specify the level of the logging desired** check box. Provide the following information:

- a. In the **Log file output location** field, specify the location of the log file for the wizard. Click **Browse** if you want to select a different location.
- b. In the **Logging Level** field, specify the severity of errors that you want logged. This log information is for the wizard only; at run time, the adapter writes the error messages and the trace information into the standard log and trace files for the server.

Note: This log pertains to the J2C Bean wizard only, not to the operation of the adapter.

6. Click **Next**.

Results

The J2C Bean wizard contacts the Siebel application, using the information you provided (such as Connection URL, user name and password) to log in. You will see the Object Discovery and Selection window.

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

Selecting business objects and services

Select business objects and services through the J2C Bean wizard to specify the business functions that you want to call and business objects 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 for and select business objects or business functions, use the following procedure.

Selecting business objects

1. In the Discovery Configuration window, select **Siebel Business Objects** as the **Type of Siebel Objects to discover**. The check box **Prompt for additional configuration settings when adding a business object** is displayed.
2. Select the check box **Prompt for additional configuration settings when adding a business object**. You need to select this if you want to select the attributes of the business objects.
3. Click **Next**.
4. 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.
- 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 check box, **Prompt for additional configuration settings when adding a business object** 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. Also, if you add a primary business component, it is displayed in the **Objects to be imported** pane. If you add a dependent business component, the primary business component is also added but it is not displayed in the **Objects to be imported** pane. You can select more than one object at a time.

- h. Click **Next**.

Selecting business services

1. In the Discovery Configuration window, select **Siebel Business Services** from the **Type of Siebel Objects to discover** list. Click **Next**.
2. 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 Properties window is displayed.

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

Note:

- 1) When a business service has only 1 complex type Input/Output argument, the Configuration Properties 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: 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 so you do not need to 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 you do not need to select the **Integration Object** and it is not added as a business object.
- 6) Siebel Enterprise Information System (EIS) has restriction that it is not possible to change the existing input and output argument names for any method of a standard business service or create new argument names as they are hard coded internally in the underlying class of the business service along with the logic. So make sure that you are working on a custom Siebel business service if you want to use custom Siebel message.

i. Click Next.

Results

You have selected the Siebel business object or business service you want to work with and selected a name for 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 should be stored.

Before you begin

Make sure 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 of 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 **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** 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 could 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.
6. If you want the business object or business objects to be generated with shorter names, select **Generate business objects with shorter names**.
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 new Java 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 desired 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, MyAdapterInboundEJB EAR
 - 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 desired 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, specify the JNDI name, which JNDI data source for an existing activation specification in WAS server or you can create it later. For example, com/eis/AdapterAS.
 5. Click **Finish**.

Chapter 4. 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.

Changing configuration properties for embedded adapters

To change 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).

Setting resource adapter properties for embedded adapters

To set resource adapter properties for your adapter after it has been deployed as part of a module, 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 module must be deployed on WebSphere Application Server.

About this task

Custom properties are default configuration properties shared by all WebSphere adapters.

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

Procedure

1. Start the administrative console.
2. Under **Applications>Application Types**, select **WebSphere enterprise applications**.
3. From the **Enterprise Applications** list, click the name of the adapter module whose properties you want to change. The **Configuration** page is displayed.
4. Under **Modules**, click **Manage Modules**.
5. Click **IBM WebSphere Adapter for Siebel Business Applications**.
6. From the **Additional Properties** list, click **Resource Adapter**.
7. On the next page, from the **Additional Properties** list, click **Custom properties**.
8. For each property you want to change, perform the following steps.

Note: See “Resource adapter properties” on page 66 for more information about these properties.

- a. Click the name of the property. The **Configuration** page for the selected property is displayed.
- b. Change the contents of the **Value** field or type a value, if the field is empty.

- You can change the number in the **Value** field and add a description of the property.
- c. Click **OK**.
9. Click the **Save** link in the **Messages** box at the top of the window.

Results

The resource adapter properties associated with your adapter module are changed.

Setting managed (J2C) connection factory properties for embedded adapters

To set managed connection factory properties for your adapter after it has been deployed as part of a module, 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 module must be deployed 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. Start the administrative console.
2. Under **Applications>Application Types**, select **WebSphere enterprise applications**.
3. In the **Enterprise Applications** list, click the name of the adapter module whose properties you want to change.
4. Under **Modules**, click **Manage Modules**.
5. Click **IBM WebSphere Adapter for Siebel Business Applications**.
6. In the **Additional Properties** list, click **Resource Adapter**.
7. On the next page, from the **Additional Properties** list, click **J2C connection factories**.
8. Click the name of the connection factory associated with your adapter module.
9. In the **Additional Properties** list, click **Custom properties**.

Custom properties are those J2C connection factory properties that are unique to 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 71 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. Click the **Save** link in the **Messages** box at the top of the window.

Results

The managed connection factory properties associated with your adapter module are changed.

Setting activation specification properties for embedded adapters

To set activation specification properties for your adapter after it has been deployed as part of a module, 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 module must be deployed 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. Start the administrative console.
2. Under **Applications>Application Types**, select **WebSphere enterprise applications**.
3. From the **Enterprise Applications** list, click the name of the adapter module whose properties you want to change.
4. Under **Modules**, click **Manage Modules**.
5. Click **IBM WebSphere Adapter for Siebel Business Applications**.
6. From the **Additional Properties** list, click **Resource Adapter**.
7. On the next page, from the **Additional Properties** list, click **J2C activation specifications**.
8. Click the name of the activation specification associated with the adapter module.
9. From the **Additional Properties** list, click **J2C activation specification custom properties**.
10. For each property you want to change, perform the following steps.

Note: See “Activation specification properties” on page 88 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. Click the **Save** link in the **Messages** box at the top of the window.

Results

The activation specification properties associated with your adapter module are changed.

Changing configuration properties for stand-alone adapters

To set configuration properties after you install a stand-alone adapter, you use the administrative console of the runtime environment. You provide general information about the adapter and then set resource adapter properties (which are used for general adapter operation). If the adapter will be used for outbound operations, you create a connection factory and then set properties for it. If the adapter will be used for inbound operations, you create an activation specification and then set properties for it.

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 WebSphere adapters.

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

Procedure

1. Start the administrative console.
2. Click **Resources** → **Resource Adapters** → **Resource adapters**.
3. In the Resource adapters page, click **IBM WebSphere Adapter for Siebel Business Applications**.
4. In the **Additional Properties** list, click **Custom properties**.
5. 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. You can change the number in the **Value** field and add a description of the property.
 - c. Click **OK**.
6. Click **Save** in the **Messages** box at the top of the page.

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. Start the administrative console.
2. Click **Resources** → **Resource Adapters** → **Resource adapters**.
3. In the Resource adapters page, click **IBM WebSphere Adapter for Siebel Business Applications**.
4. In the **Additional Properties** list, click **J2C connection factories**.
5. If you are going to use an existing connection factory, skip ahead to select from the list of existing connection factories.

Note: If you selected **Use predefined connection properties** when you used 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 could type AdapterCF.
 - c. Type a value for **JNDI name**. For example, you could type com/eis/AdapterCF.
 - d. Optional: Select an authentication alias from the **Component-managed authentication alias** list.
 - e. Click **OK**.
 - f. Click **Save** in the **Messages** box at the top of the page.
The newly created connection factory is displayed.
6. In the list of connection factories, click the one you want to use.
 7. In the **Additional Properties** list, click **Custom properties**.

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

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

Note: See “Managed connection factory properties” on page 71 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**.
9. After you have finished setting properties, click **Apply**.
 10. Click **Save** in the **Messages** box at the top of the window.

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. Start the administrative console.
2. Click **Resources** → **Resource Adapters** → **Resource adapters**.
3. In the Resource adapters page, click **IBM WebSphere Adapter for Siebel Business Applications**.
4. In the **Additional Properties** list, click **J2C activation specifications**.
5. If you are going to use an existing activation specification, skip ahead to select from an existing list of activation specifications.

Note: If you selected **Use predefined connection properties** when you used the J2C Bean wizard to configure the adapter module, you do not need to 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 could type AdapterAS.
- c. Type a value for **JNDI name**. For example, you could 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.
6. In the list of activation specifications, click the one you want to use.
7. In the Additional Properties list, click **J2C activation specification custom properties**.
8. 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**.
9. After you have finished setting properties, click **Apply**.
10. Click **Save** in the **Messages** box at the top of the page.

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.

Standalone deployment

The dependency libraries may be added to the resource adapter deployed standalone either during initial deployment of the RAR file or by configuring the Resource Adapter properties after deployment.

About this task

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

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

Make sure dependent files are available on the server machine in the separate folder. If needed, copy 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 above step 1.
3. Deploy the EAR to the server.
4. Configure the EAR to reference defined shared library.

Chapter 5. Troubleshooting and support

Common troubleshooting techniques and self-help information help you identify and solve problems quickly.

Support for the 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 adapter's messages 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, make sure that the first seven characters of the adapter ID property are unique for each instance so that you can correlate 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 log and trace information to a particular instance of an adapter. To illustrate how the length of the adapter ID property affects the filtering of log and trace files, suppose 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. For example, suppose you set the adapter ID properties of two instances to Instance01 and Instance02. You will not be able to separately 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, to prevent 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 about the adapter ID property, see "Adapter ID (AdapterID)" on page 67.

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.

About this task

Configuring logging properties

Use the 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.

For more information about monitoring on an application server, including service components and event points, see the documentation for your application server.

You can change the log configuration statically or dynamically. Static configuration takes effect when you start or restart the application server. Dynamic, or runtime, 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 children of the log, which recursively propagate the change to their children, 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, click **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 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. 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, respectively.

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, in the Modules list, click **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. This method allows you to make changes that take effect right away.

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 thereby 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 located 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 workaround to some problems you may encounter while running Adapter for Siebel Business Applications with your database are provided. These problems and solutions are also documented as technotes on the Software support Web site.

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 will remain idle, without any transactions coming in for 60 minutes, increase the timeout value on the Siebel application to a higher number (for example, 1 1/2 hours). This ensures that the connections remain valid for a longer period of time. If you cannot increase the timeout value, then resend the failed transaction.

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 external service 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 external service 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 will be 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 should 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.

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

Problem

WebSphere Adapter for Siebel Business Applications generates multiple container attributes 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, if 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 instead of one: BusinessAddress, BusinessAddress', and BusinessAddress3. You can confirm that SimpleLink application-specific information is set against the three container attributes.

Solution and Workaround

Although the extra container attributes should not affect the transaction if they are left untouched, you can remove the unwanted attributes by using the business object editor in Rational Application Developer for WebSphere Software.

XAResourceNotAvailableException

When the application server log contains repeated reports of the `com.ibm.ws.Transaction.XAResourceNotAvailableException` exception, remove transaction logs to correct the problem.

Symptom:

When the adapter starts, the following exception is repeatedly logged in the application server log file:

```
com.ibm.ws.Transaction.XAResourceNotAvailableException
```

Problem:

A resource was removed while the application server was committing or rolling back a transaction for that resource. When the adapter starts, it tries to recover the transaction but cannot because the resource was removed.

Solution:

To correct this problem, use the following procedure:

1. Stop the application server.
2. Delete the transaction log file that contains the transaction. Use the information in the exception trace to identify the transaction. This prevents the server from trying to recover those transactions.

Note: In a test or development environment, you can generally delete all of the transaction logs. In Rational Application Developer for WebSphere Software, delete the files and subdirectories of the transaction log directory, `server_install_directory\profiles\profile_name\tranlog`.

In a production environment, delete only the transactions that represent events that you do not need to process. One way to do this is to reinstall the adapter, pointing it to the original event database used, and deleting only the transactions you do not need. Another approach is to delete the transactions from either the log1 or log2 file in the following directory:

```
server_install_directory\profiles\profile_name\tranlog\node_name\server_name\  
transaction\tranlog
```

3. Start the application server.

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 Web site

The WebSphere Adapters software support Web site at <http://www.ibm.com/software/integration/wbiadapters/support/> provides links to many resources to help you learn about, use, and troubleshoot WebSphere Adapters, including the following types of

- 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 should apply is available at the following location: <http://www.ibm.com/support/docview.wss?doc=aimadp&rs=695&uid=swg27010397>

Technotes

Technotes provide the most current documentation about the 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 Adapters, visit this address:

<http://www.ibm.com/support/search.wss?tc=SSMKUK&rs=695&rank=8&dc=DB520+D800+D900+DA900+DA800+DB560&dtm>

Plug-in for IBM Support Assistant

Adapter for Siebel Business Applications provides a plug-in for IBM Support Assistant, which is a free, local software serviceability workbench. The plug-in supports the dynamic trace feature. For information about installing or using IBM Support Assistant, visit this address:

<http://www.ibm.com/software/support/isa/>

Chapter 6. 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 should 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 6. 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 7. 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 8. 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 9. 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 objects and components. The following table describes the metadata elements located at the container level for Siebel business objects and Siebel business components.

Table 10. 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.
KeepRelations	Specifies whether or not the child objects needs 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.

Table 10. Container attribute-level metadata elements generated against Siebel business objects and Siebel business components (continued)

Metadata element	Description
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.

Supported operations

An operation is the action that an adapter can perform on the Siebel 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 11. Supported operations

Operation	Description
Create	Creates the business component
Delete	Deletes the business component and its children
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 12. 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 should set the 'Folder' property with a unique value. This will avoid overwriting 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 'EAISiebelAdapter.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 13. 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 14. 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>

Table 14. Business object naming conventions for Siebel business objects (continued)

Element	Naming convention
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 should set the 'Folder' property with a unique value. This will avoid overwriting 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, '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 15. 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

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 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 will process the field using its assigned default value, and that default value will also be displayed on the 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 will note 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 will state 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 the following:</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>
Example	<p>Provides sample property values, for example:</p> <p>"If Language is set to JA (Japanese), Codepage 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>

Row	Explanation
Bidi supported	Indicates whether the property is supported in bidirectional (bidi) processing. Bidirectional processing pertains to the task of processing data that contains both left-to-right (Hebrew or Arabic, for example) and right-to-left (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 server. The properties you configure in the J2C Bean wizard specify such things as connection configuration, bidirectional (bidi) properties, and logging and tracing options.

Once a connection between the J2C Bean wizard and the Siebel server is established, the J2C Bean wizard is able to access the metadata it needs from the Siebel 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 on how to read the property detail tables in the sections that follow, see “Guide to information about properties” on page 58.

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 16. 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.
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 server
Method name	The name of the business service method to be implemented
Password	The password for the corresponding user name

Table 16. J2C Bean wizard properties (continued)

Property name in the wizard	Description
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 server mode and controls the kind of data 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 server, and the value is set to true, the adapter takes advantage of the load balancing feature to connect to the Siebel server more efficiently.
User name	The user name that is used to log into the Siebel application

Adapter style (AdapterStyle)

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

Table 17. 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 18. 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 19. 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

Delimiter for keys in the event store (DelimiterForKeysInTheEventStore)

Table 20. 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 don't 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 21. 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

Table 21. Folder details (continued)

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 22. 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. Alpha-numeric 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 server.

Table 23. 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 server.
Globalized	No
Bidi supported	No

Method name (MethodName)

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

Table 24. Method name details

Required	Yes
Default	Query
Property type	String

Table 24. Method name details (continued)

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 25. 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 26. 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	Let's say, for 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 27. Siebel business object name for event store details

Required	Yes
Default	IBM_EVENT
Property type	String

Table 27. Siebel business object name for event store details (continued)

Usage	After clicking on 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 28. 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 server view mode and controls the data that can be retrieved and what actions can be performed on it.

Table 29. Siebel server view mode details

Required	Yes
Default	3
Property type	Integer
Usage	This property displays after clicking on 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 30. Type of Siebel objects to discover details

Required	Yes
Possible values	Siebel business objects and Siebel business services
Default	Siebel business objects
Property type	String

Table 30. Type of Siebel objects to discover details (continued)

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 server uses resonate support.

Table 31. 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 on 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 will take advantage of the load balancing feature to connect to the Siebel 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 into the Siebel application.

Table 32. 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 using the external service wizard when you configure the adapter. After deploying the adapter, use the administrative console to change these properties.

The following properties for logging and tracing are no longer required in version 6.1.x, 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 on how to read the property detail tables in the sections to follow, see “Guide to information about properties” on page 58.

Table 33. 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 strings of X's instead of user data in 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 server, and the value is set to true, the adapter will take advantage of the load balancing feature to connect to the Siebel server more efficiently
Siebel server view mode	viewMode	Specifies the Siebel view mode and controls the kind of data to be retrieved and what actions can 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 34. Adapter ID details

Required	Yes
Default	001
Property type	String

Table 34. Adapter ID details (continued)

Usage	<p>This property identifies the adapter instance in 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, make sure that the first seven characters of the adapter ID property are unique for each instance so that you can correlate 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 log and trace information to a particular instance of an adapter.</p> <p>To illustrate how the length of the adapter ID property affects the filtering of log and trace files, suppose 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. For example, suppose you set the adapter ID properties of two instances to Instance01 and Instance02. You will not be able to separately 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, can be set both at the resource adapter level and the managed connection factory level. After using 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) property

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 35. 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, can be set both at the resource adapter level and the managed connection factory level. After using 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 36. 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 37. Log file maximum size details

Required	No
Default	0
Property type	Integer
Usage	When the log file reaches it maximum size, the adapter start 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 38. 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 39. Log number of files details

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

Table 39. 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 server uses resonate support.

Table 40. Resonate support details

Required	No
Possible values	True False
Default	True
Property type	Boolean
Usage	If the check box is selected, the value for Resonate Support is set to true, and the adapter takes advantage of the load balancing feature to connect to the Siebel server more efficiently. If the check box is not selected, the value for Resonate Support is set to false.
Globalized	No

Siebel server view mode (SiebelServerViewMode)

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

Table 41. 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 42. 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 43. 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 44. 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 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 58

Adapter ID (AdapterID)

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

Table 45. Adapter ID details

Required	Yes
Default	001
Property type	String

Table 45. Adapter ID details (continued)

Usage	<p>This property identifies the adapter instance in 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, make sure that the first seven characters of the adapter ID property are unique for each instance so that you can correlate 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 log and trace information to a particular instance of an adapter.</p> <p>To illustrate how the length of the adapter ID property affects the filtering of log and trace files, suppose 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. For example, suppose you set the adapter ID properties of two instances to Instance01 and Instance02. You will not be able to separately 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, can be set both at the resource adapter level and the managed connection factory level. After using 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

Connection URL (ConnectionURL)

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

Table 46. Connection URL details

Required	Yes
Default	No default value
Property type	String
Usage	<p>Protocol://machinename:port/enterprisename/object manager/server name. 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 above (for Siebel versions 7.8 and 8), another port (2321) has been specified.</p>
Globalized	Yes
Bidi supported	Yes

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

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 47. 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, can be set both at the resource adapter level and the managed connection factory level. After using 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

Language code (LanguageCode)

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

Table 48. 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). This is used to log on to the Siebel server.
Globalized	No
Bidi supported	No

Password (Password)

This property specifies the password for the corresponding user name.

Table 49. Password details

Required	Yes
Default	No default value
Property type	String
Example	sadmin
Usage	This property will display after clicking on the advanced button on the connection properties window on the J2C Bean wizard. The password is saved in .import and .export files. This makes it possible for the adapter to connect to the Siebel application after it has been deployed. If a J2C Authentication Alias is used, then a password is not required.
Globalized	Yes
Bidi supported	Yes

Prefix (Prefix)

This property specifies the prefix for the business object name.

Table 50. 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.
Example	If you use the prefix, IBM, generate a business object for the EAI Siebel Adapter and the Insert method, and 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

Resonate support (ResonateSupport)

This property indicates whether the Siebel server uses resonate support.

Table 51. Resonate support details

Required	No
Possible values	True False
Default	True
Property type	Boolean
Usage	If the check box is selected, it will be set to true, and the adapter will take advantage of the load balancing feature to connect to the Siebel 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 into the Siebel application.

Table 52. User name details

Required	Yes
Possible values	None
Default	No default value
Property type	String
Usage	This property will display after clicking on the advanced button on the connection properties window on the J2C Bean wizard. The user name is saved in .import and .export files. This makes it possible for the adapter to connect to the Siebel application after it has been deployed. If a J2C Authentication Alias is used, then a password is not required.
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 53. View mode details

Required	Yes
Default	Although the adapter supports values 1 - 9, the default value is 3.
Property type	Integer
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. To change these properties after the application is deployed, use the assembly editor in Rational Application Developer for WebSphere Software. The properties reside in the method binding of the import.

Table 54 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 58.

Table 54. Interaction specification property for the Adapter for Siebel Business Applications

Property name		Description
In the wizard	In the assembly editor	
“Maximum records (MaximumRecords)”	MaximumRecords	Maximum number of records to return during a RetrieveAll operation

Maximum records (MaximumRecords)

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

Table 55. 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 error MatchesExceededLimitException. The adapter uses this property to help avoid out-of-memory issues.
Property type	Integer
Globalized	No

Table 55. Maximum records details (continued)

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 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 will process the field using its assigned default value, and that default value will also be displayed on the 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 will note 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 will state 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 the following:</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), Codepage 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 pertains to the task of processing data that contains both left-to-right (Hebrew or Arabic, for example) and right-to-left (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 server. The properties you configure in the J2C Bean wizard specify such things as connection configuration, bidirectional (bidi) properties, and logging and tracing options.

Once a connection between the J2C Bean wizard and the Siebel server is established, the J2C Bean wizard is able to access the metadata it needs from the Siebel 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 on how to read the property detail tables in the sections that follow, see "Guide to information about properties" on page 58.

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 56. 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.
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 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 server mode and controls the kind of data 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 server, and the value is set to true, the adapter takes advantage of the load balancing feature to connect to the Siebel server more efficiently.
User name	The user name that is used to log into the Siebel application

Adapter style (AdapterStyle)

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

Table 57. 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 58. 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 59. 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/sebldevl .• 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

Delimiter for keys in the event store (DelimiterForKeysInTheEventStore)

Table 60. 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.

Table 60. Delimiter for keys in the event store details (continued)

Examples	The default value for this property can be changed, however, if you remove the default value and don't 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 61. 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 62. 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. Alpha-numeric 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 server.

Table 63. 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 server.
Globalized	No
Bidi supported	No

Method name (MethodName)

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

Table 64. 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 65. 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 66. 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.

Table 66. Prefix details (continued)

Example	Let's say, for 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 67. Siebel business object name for event store details

Required	Yes
Default	IBM_EVENT
Property type	String
Usage	After clicking on 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 68. 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 server view mode and controls the data that can be retrieved and what actions can be performed on it.

Table 69. Siebel server view mode details

Required	Yes
Default	3
Property type	Integer

Table 69. Siebel server view mode details (continued)

Usage	This property displays after clicking on 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 70. Type of Siebel objects to discover details

Required	Yes
Possible values	Siebel business objects and 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 server uses resonate support.

Table 71. 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 on 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 will take advantage of the load balancing feature to connect to the Siebel 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 into the Siebel application.

Table 72. User name details

Required	Yes
Default	No default value
Property type	String

Table 72. User name details (continued)

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 using the external service wizard when you configure the adapter. After deploying the adapter, use the administrative console to change these properties.

The following properties for logging and tracing are no longer required in version 6.1.x, 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 on how to read the property detail tables in the sections to follow, see “Guide to information about properties” on page 58.

Table 73. 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 strings of X's instead of user data in 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 server, and the value is set to true, the adapter will take advantage of the load balancing feature to connect to the Siebel server more efficiently
Siebel server view mode	viewMode	Specifies the Siebel view mode and controls the kind of data to be retrieved and what actions can be performed
(Not available)	TraceFileMaxSize	Deprecated

Table 73. Resource adapter properties (continued)

Property name		Description
In the wizard	In the administrative console	
(Not available)	TraceFileName	Deprecated
(Not available)	TraceNumberOfFiles	Deprecated

Adapter ID (AdapterID)

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

Table 74. Adapter ID details

Required	Yes
Default	001
Property type	String
Usage	<p>This property identifies the adapter instance in 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, make sure that the first seven characters of the adapter ID property are unique for each instance so that you can correlate 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 log and trace information to a particular instance of an adapter.</p> <p>To illustrate how the length of the adapter ID property affects the filtering of log and trace files, suppose 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. For example, suppose you set the adapter ID properties of two instances to Instance01 and Instance02. You will not be able to separately 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, can be set both at the resource adapter level and the managed connection factory level. After using 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) property

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

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

Table 75. Disguise user data as "XXX" in log and trace files details (continued)

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, can be set both at the resource adapter level and the managed connection factory level. After using 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

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 76. 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 77. Log file maximum size details

Required	No
Default	0
Property type	Integer
Usage	When the log file reaches it maximum size, the adapter start 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 78. 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 79. Log number of files details

Required	No
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 server uses resonate support.

Table 80. Resonate support details

Required	No
Possible values	True False
Default	True
Property type	Boolean
Usage	If the check box is selected, the value for Resonate Support is set to true, and the adapter takes advantage of the load balancing feature to connect to the Siebel server more efficiently. If the check box is not selected, the value for Resonate Support is set to false.
Globalized	No

Siebel server view mode (SiebelServerViewMode)

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

Table 81. 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.

Table 81. View mode details (continued)

Globalized	No
------------	----

Trace file maximum size (TraceFileMaxSize)

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

Table 82. 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 83. 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 84. 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 activation specification properties either through the J2C Bean wizard or the administrative console.

Activation specification properties are used during endpoint activation to notify the adapter of eligible listeners. During inbound processing, the adapter uses these event listeners to receive events before forwarding them to the endpoint.

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 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 58

Connection URL (ConnectionURL)

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

Table 85. Connection URL details

Required	Yes
Default	No default value
Property type	String
Usage	Protocol://machinename:port/enterprisename/object manager/server name. 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 above (for Siebel versions 7.8 and 8), another port (2321) has been specified.
Globalized	Yes
Bidi supported	Yes

Delivery type (DeliveryType)

This property specifies the order in which events are delivered by the adapter to the export.

Table 86. 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.
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 87. 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 88. Ensure once-only event delivery details

Required	Yes
Possible values	True False
Default	True
Property type	Boolean
Usage	When this property is set to True, the adapter provides assured once event delivery. This means that each event will be delivered once and only once. A value of False does not provide assured once event delivery, but provides better performance. 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. 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.
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 89. Event component name details

Required	Yes
Default	IBM2 (for Siebel version 7.x) and IBM Event (for Siebel version 8)
Property type	String
Usage	The default is IBM2, for Siebel version 7.x and IBM Event, for Siebel version 8. If you select one of these defaults to configure the event business component on the Siebel server, it will be 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.

Table 89. Event component name details (continued)

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 90. Event types to process details

Required	No
Possible values	A comma-delimited (,) list of business object types
Default	null
Property type	String
Usage	Events are filtered by business object type. 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.
Example	To receive only events relating to the Customer and Order business objects, specify this value: Customer,Order 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.
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 91. Retry limit for failed events details

Required	No
Possible values	Integers
Default	5
Property type	Integer
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

Table 91. Retry limit for failed events details (continued)

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

Interval between polling periods (PollPeriod)

This property specifies the length of time that the adapter waits between polling periods.

Table 92. 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 will occur immediately to make up for the lost time caused by the delay.
Globalized	No
Bidi supported	No

Language code (LanguageCode)

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

Table 93. 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 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 94. Maximum connections details

Required	No
Default	1
Property type	Integer
Usage	Only positive values are valid. The adapter considers any positive entry less than 1 to be equal to 1. Typing a negative value or 1 for this property may result in runtime errors.
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 95. 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 may 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 96. 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. Typing a negative value or 1 for this property may result in run time errors.
Globalized	No
Bidi supported	No

Number of times to retry the system connection (RetryLimit)

This property specifies the number of times the adapter tries to reestablish an inbound connection.

Table 97. Number of times to retry the system connection details

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	Yes
Bidi supported	No

Password (Password)

This property specifies the password for the corresponding user name.

Table 98. Password details

Required	Yes
Default	No default value
Property type	String
Usage	This property will display after clicking on the advanced button on the connection properties window on the J2C Bean wizard. The password is saved in .import and .export files. This makes it possible for the adapter to connect to the Siebel application after it has been deployed. If a J2C Authentication Alias is used, then a password is not required.
Example	sadmin
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 99. 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

Retry interval if connection fails (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 establish a new connection.

Table 100. Retry interval details

Required	Yes
----------	-----

Table 100. Retry interval details (continued)

Default	2000
Unit of measure	Milliseconds
Property type	Integer
Usage	Only positive values are valid. When the adapter encounters an error related to the inbound connection, this property specifies the length of time the adapter waits before trying to establish a new connection.
Globalized	Yes
Bidi supported	No

Stop the adapter when an error is encountered while polling (StopPollingOnError)

This property specifies whether the adapter will stop polling for events when it encounters an error during polling.

Table 101. 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 into the Siebel application.

Table 102. User name details

Required	Yes
Default	No default value
Property type	String
Usage	This property will display after clicking on the advanced button on the connection properties window on the J2C Bean wizard. The user name is saved in .import and .export files. This makes it possible for the adapter to connect to the Siebel application after it has been deployed. If a J2C Authentication Alias is used, then a password is not required.
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 prior to introducing the data to WebSphere Application Server.

Five attributes comprise bidirectional format. When you set bidirectional properties, you assign values for each of these attributes. The attributes and settings are listed in the following table.

Table 103. 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	
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. Do this by specifying values for each of the five bidirectional format attributes (listed in the preceding table) for the property. 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. Do this by specifying values for each of the five bidirectional format attributes (listed in the preceding table) for the property. 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. Do this 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. Once 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

A

- activation specification properties
 - list of 89
 - setting in administrative console 39, 42
 - setting with external service wizard 89
- adapter
 - project, create 21
- adapter architecture 3
- Adapter for Siebel Business Applications
 - administering 37
 - overview 1
- AdapterID property 66, 84
- AdapterStyle property 60, 77
- application-specific information
 - business-object-level metadata 53
 - container attribute-level metadata 53
 - property-level metadata 53
 - siebel business components 53
 - siebel business objects 53
 - siebel business services 53
- AssuredOnceDelivery property 89
- authentication
 - description 14
 - J2C Bean wizard 14
 - run time 15
- authentication alias 15

B

- bidirectional data transformation properties
 - activation specification properties 98
 - connection properties used in the external service wizard 98
 - list of 98
 - managed connection properties 98
- business graph 6, 7
- business object information 53
- business objects
 - application-specific information 53
 - configuring for inbound processing 34
 - configuring for outbound processing 26
 - creating
 - structure 11
 - naming conventions 55
 - selecting for inbound processing 32
 - selecting for outbound processing 24
 - supported data operations 55
- business-object-level metadata 53
- BusinessObjectNameSpace property 60, 77

C

- clustered environment
 - deploying in 17
 - description 17
 - inbound processes 17
 - outbound processes 17
- compatibility matrix 2
- confidential data, disguising 13
- confidential tracing 13

- configuring
 - logging 45
 - tracing 45
- connection properties, J2C Bean wizard 23, 31
- ConnectionURL property 60, 71, 77, 89
- connector project 21
- container attribute-level metadata 54
- Create operation 4
- custom properties
 - activation specification 39, 42
 - managed connection factory 38, 41
 - resource adapter 37, 40

D

- data operations 55
- debugging
 - self-help resources 52
 - XAResourceNotAvailableException exception 51
- Delete operation 5
- DelimiterForKeysInTheEventStore property 60, 77
- DeliveryType property 89
- deployment
 - options 15

E

- embedded adapter
 - activation specification properties, setting 39
 - considerations for using 16
 - description 15
 - managed connection factory properties, setting 38
 - resource adapter properties, setting 37
- enableHASupport property 17, 66, 84
- event business components 7
- event notification 3
- event record 7
- event store
 - event business components 7
 - event record 7
 - event triggers 7
 - polling 7
- event table
 - creating 19
- event triggers 7
- EventComponentName property 89
- EventDelimiter property 66, 84
- EventTypeFilter property 89
- exceptions
 - XAResourceNotAvailableException 51
- Exists operation 5
- external dependencies, adding 22
- external service wizard
 - overview 11
 - properties, connection 60, 77
 - setting connection properties 23

F

- FFDC (first-failure data capture) 48

files

SystemOut.log log file 47

trace.log trace file 47

FilterFutureEvents property 89

first-failure data capture (FFDC) 48

Folder property 60, 77

G

GenerateBusinessObjectsWithShorterNames property 60, 77

globalization and bidirectional data transformation

bidirectional properties that identify data for transformation 96

bidirectional script data formats 96

bidirectional script data transformation 96

H

hardware and software requirements 2

hardware requirements 2

high-availability environment

deploying in 17

description 17

inbound processes 17

outbound processes 17

I

inbound configuration properties 76

inbound processing

business graph 7

overview 3, 7

selecting business objects 32

interaction specification properties

list of 75

setting in assembly editor 75

setting with external service wizard 75

J

J2C Bean wizard

authentication in 14

JAR file, adding external 22

Java 2 security 15

L

LanguageCode property 60, 71, 77, 89

Log Analyzer 46

Log and Trace Analyzer, support for 45

log and trace files 45

log files

changing file name 47

disabling 46

enabling 46

level of detail 46

location 47

LogFileMaxSize property 66, 84

LogFilename property 66, 84

logging

configuring properties with administrative console 46

LogNumberOfFiles property 66, 84

M

managed (J2C) connection factory properties

list of 71

setting in administrative console 38, 41, 71

setting with external service wizard 71

matrix, compatibility 2

MaximumConnections property 89

MaximumRecords property 75

metadata 53

attribute-level 54

business-object-level 53

property-level 54

MethodName property 60, 77

MinimumConnections property 89

N

naming conventions for business objects

business graph 55

siebel business components 55

siebel business services 55

siebel integration components 55

O

outbound configuration properties 58

outbound operations 4

Create 4

Delete 5

Exists 5

Retrieve 5

RetrieveAll 6

Update 6

outbound processing

ApplyChanges operation 3

Create operation 3

Delete operation 3

Exists operation 3

overview 3

Retrieve operation 3

RetrieveAll operation 3

selecting business objects 24

supported outbound operations

business object hierarchy 3

Update operation 3

P

package files for adapters 46

Password property 60, 71, 77, 89

polling 7

PollPeriod property 89

PollQuantity property 89

Prefix property 71

PrefixForBusinessObjectNames property 60, 77

problem determination

self-help resources 52

solutions to common problems 49

XAResourceNotAvailableException exception 51

properties

activation specification 39, 42

configuration properties

inbound 76

outbound 58

external service connection 60, 77

- properties (*continued*)
 - inbound configuration 76
 - managed (J2C) connection factory 38, 41
 - outbound configuration 58
 - resource adapter 37, 40
- property-level metadata 54

R

- record, event 7
- requirements, hardware and software 2
- ResonateSupport property 66, 71, 84
- resource adapter properties
 - list of 66, 84
 - setting in administrative console 37, 40
 - setting with external service wizard 66, 84
- Retrieve operation 5
- RetrieveAll operation 6
- Retry limit property 93
- RetryInterval property 89
- RetryLimit property 89
- runtime environment
 - authentication in 15

S

- security
 - disguising sensitive data 13
- security, Java 2 15
- self-help resources 52
- sensitive data, disguising 13
- setting connection properties 23, 31
- SiebelBusinessObjectNameForEventStore property 60, 77
- SiebelRepositoryName property 60, 77
- SiebelServerViewMode property 60, 66, 77, 84
- software dependencies, adding external 22
- software requirements 2
- solutions to common problems
 - adapter does not generate an exception 49
 - adapter does not generate artifacts properly 49
 - adapter generates multiple container attributes 49
 - adapter may time out 49
- stand-alone adapter
 - activation specification properties, setting 42
 - considerations for using 16
 - description 15
 - managed connection factory properties, setting 41
 - resource adapter properties, setting 40
- StopPollingOnError property 89
- support
 - overview 45
 - self-help resources 52
- supported data operations 55
- SystemOut.log file 47

T

- technotes 2, 52
- trace files
 - changing file name 47
 - disabling 46
 - enabling 46
 - level of detail 46
 - location 47
- trace.log file 47
- TraceFileMaxSize property 66, 84

- TraceFilename property 66, 84
- TraceNumberOfFiles property 66, 84
- tracing
 - configuring properties with administrative console 46
- triggers, event 7
- troubleshooting
 - overview 45
 - self-help resources 52
 - XAResourceNotAvailableException exception 51
- TypeOfSiebelObjectsToDiscover property 60, 77

U

- Update operation 6
- UseResonateSupportForLoadBalancingOnSiebelServer property 60, 77
- UserName property 60, 71, 77, 89

V

- ViewMode property 71

W

- WebSphere Extended Deployment 17

X

- XAResourceNotAvailableException 51



Printed in USA