WebSphere® Adapters

IBM

Version 6.0.2

Adapter for Siebel Business Applications User Guide

Note
Before using this information, be sure to read the general information in "Notices" on page 155.

22December2006

This edition applies to version 6, release 0, modification 2 of WebSphere Adapter for Siebel Business Applications (product number 5724-L80) and to all subsequent releases and modifications until otherwise indicated in new editions.

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

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

Contents

Chapter 1. About this information

This documentation is for integration developers who implement, configure, and deploy WebSphere Adapter for Siebel Business Applications. To use it, you should understand business integration concepts and possess certain technical skills.

Integration developers design, assemble, test, and deploy business integration solutions. This information is for those who are deploying WebSphere Adapter for Siebel Business Applications in a solution that requires data exchange between enterprise information systems (EIS) and J2EE applications. To use it, you should understand and have experience with the following concepts, standards, and tools:

- The business solution and environment.
- Knowledge on Siebel Business Applications, how to write server scripts on business components, and limited server administration is required.
- Business integration mechanisms, including the Service Component Architecture (SCA) programming model and the Service Data Objects (SDO) data model.
- The J2EE standard and J2EE applications.
- The capabilities and requirements of WebSphere Process Server or WebSphere Enterprise Service Bus, depending on the host used in the environment. You should know how to configure and administer the host server and how to use the administrative console.
- The tools and capabilities provided by WebSphere Integration Developer. You should know how to use these tools to wire components and complete other integration tasks.

To complete the deployment, you should know how to perform the following tasks:

- Create required scripts, tools, and templates for both testing and deployment
- Resolve interdependencies between entities such as enterprise beans, workflows, and Web pages
- Build data models for external data access tools
- Implement security measures

Chapter 2. What's new

WebSphere Adapter for Siebel Business Applications, version 6.0.2 provides enhancements to version 6.0 of the adapter.

New in this release

WebSphere Adapter for Siebel Business Applications, version 6.0.2, has new and updated features.

New in version 6.0.2:

- Siebel business object and business component support
- High availability support for inbound processing. For more information, see "WebSphere Adapters in clustered environments" "WebSphere Adapters in clustered environments" on page 35.

Updated in version 6.0.2:

- Quick start tutorials for inbound and outbound processing
- Bidirectional language text support

Release notes

The release notes for WebSphere Adapter for Siebel Business Applications, version 6.0.2. summarize new features and functions in this release and document any known workarounds.

You can see release notes for WebSphere Adapter for Siebel Business Applications by using the following link: http://publib.boulder.ibm.com/infocenter/dmndhelp/v6rxmx/to

Chapter 3. Introduction to WebSphere Adapters

IBM WebSphere Adapters make it possible for Java 2 Platform, Enterprise Edition (J2EE) components, such as new e-business applications, to communicate with resources on an enterprise information system (EIS). An EIS is the information infrastructure for an enterprise (for example, an enterprise resource planning [ERP] system).

A WebSphere adapter acts as an intermediary between the J2EE component and the EIS, so that the J2EE component does not need to understand the low-level API or data structures of the EIS.

WebSphere Adapters can be one of two types: application or technology.

- Application adapters connect to existing packaged applications (such as SAP Software, Siebel, PeopleSoft Enterprise, and JD Edwards EnterpriseOne) so that you can make use of data and services specific to the applications.
- Technology adapters provide connectivity to data through such technologies and protocols as relational databases, flat files, e-mail messages, and FTP.

As part of the WebSphere family of products, WebSphere Adapters work with WebSphere Integration Developer and either WebSphere Process Server or WebSphere Enterprise Service Bus.

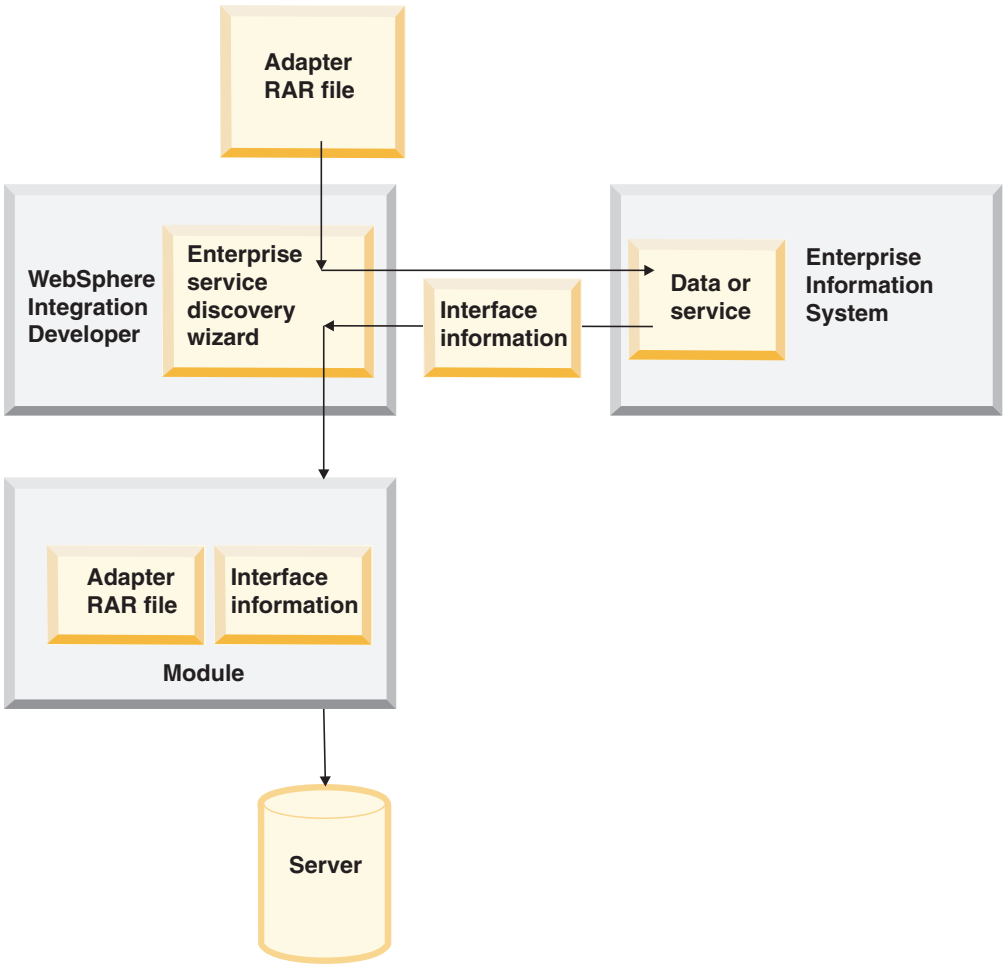
- WebSphere Integration Developer is the tooling environment for the WebSphere adapters.

  You use WebSphere Integration Developer to assemble a module that is deployed on WebSphere Process Server or WebSphere Enterprise Service Bus. From within WebSphere Integration Developer, you import the adapter (which is packaged as a resource adapter [RAR] file) and connect to the EIS. The enterprise service discovery wizard of WebSphere Integration Developer looks for data and services on the EIS and creates the interface information needed to gain access to the data and services. Finally, WebSphere Integration Developer generates a module that includes the adapter and the interface information.
- WebSphere Process Server or WebSphere Enterprise Service Bus is the runtime environment for the WebSphere adapters.

  You deploy the module generated by WebSphere Integration Developer to one of the servers.

The generation and deployment of the module is illustrated in the following figure.

Figure 1. How a module is generated and deployed


Adapter
RAR file
WebSphere
Integration
Developer
Enterprise
service
discovery
wizard
Interface
information
Data or
service
Enterprise
Information
System
Adapter
RAR file
Interface
information
Module
Server

Chapter 4. Introduction to WebSphere Adapter for Siebel Business Applications

IBM WebSphere Adapter for Siebel Business Applications connects Java 2 Platform, Enterprise Edition (J2EE) components running on WebSphere Process Server or WebSphere Enterprise Service Bus with remote file systems running on an enterprise information system (EIS). The adapter provides a means for the J2EE component and the remote file system to interact. For example, the J2EE application can be configured to update an account record in the Siebel system.

Hardware and software requirements

Before installing Adapter for Siebel Business Applications, you must verify that your environment meets the necessary requirements. These requirements fall into two categories: supported platforms for running the adapter installer, and hardware and software requirements for configuring, deploying, and running the adapter.

Supported platforms for running the adapter installer

The supported platforms for running the adapter installer are located in the ″Installing″ section of Installing IBM WebSphere Adapters.

Hardware and software requirements for configuring, deploying, and running the adapter

The hardware and software requirements for configuring, deploying, and running the adapter are located at the following Web site: IBM WebSphere Adapters and IBM WebSphere Business Integration Adapters: software requirements. From the IBM WebSphere Adapters list, select the link for the Adapter for Siebel Business Applications, Version 6.0.2.

Standards compliance

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

Accessibility

IBM® strives to provide products with usable access for everyone, regardless of age or ability. The WebSphere® Adapters software is fully accessible and section 508-compliant. Accessibility features enable users with physical disabilities, such as restricted mobility or limited vision, to operate software products successfully. These features are built into the installation and administration features of WebSphere Adapters.

Installation

You can install WebSphere Adapters either through a graphical user interface or silently through a script. The silent installation method is recommended for users with accessibility needs.

Administration

The administrative console of either WebSphere Process Server or WebSphere Enterprise Service Bus is the primary interface for deployment and administration of the 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 ViaVoice®, to enter data and to navigate the user interface
- Operate features by using the keyboard instead of the mouse

You can configure and use product features by using standard text editors and 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.

Enterprise service discovery wizard

The enterprise service discovery wizard is the primary component used to create enterprise applications with the adapters. This wizard is implemented as an Eclipse plug-in that is available through WebSphere Integration Developer and is fully accessible.

Keyboard navigation

This product uses standard Microsoft Windows® navigation keys.

IBM and accessibility

See the *IBM Accessibility Center* for more information about the commitment that IBM has to accessibility.

Internet Protocol Version 6.0

IBM WebSphere Process Server relies on WebSphere Application Server for Internet Protocol Version 6.0 compatibility.

IBM WebSphere Application Server Version 6.0 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 www.ipv6.org.

Technical overview of the Adapter for Siebel Business Applications

IBM WebSphere Adapter for Siebel Business Applications provides support for Siebel business objects, Siebel business components, and Siebel business services. The adapter supports both inbound and outbound operations.

Siebel business applications allows data exchange and business process execution using business services, business components, and objects. These entities provide business workflow and process execution to help companies manage their customer relations. The Siebel system supports the integration of its business services and business objects using the Siebel Java Data Bean.

The Siebel Java Data Bean interacts with the following types of application objects:

- Siebel business objects and business components
- Siebel business services with Siebel integration objects and components

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

The following figure shows the adapter architecture including arrows that represent the processing flow for both inbound and outbound operations.

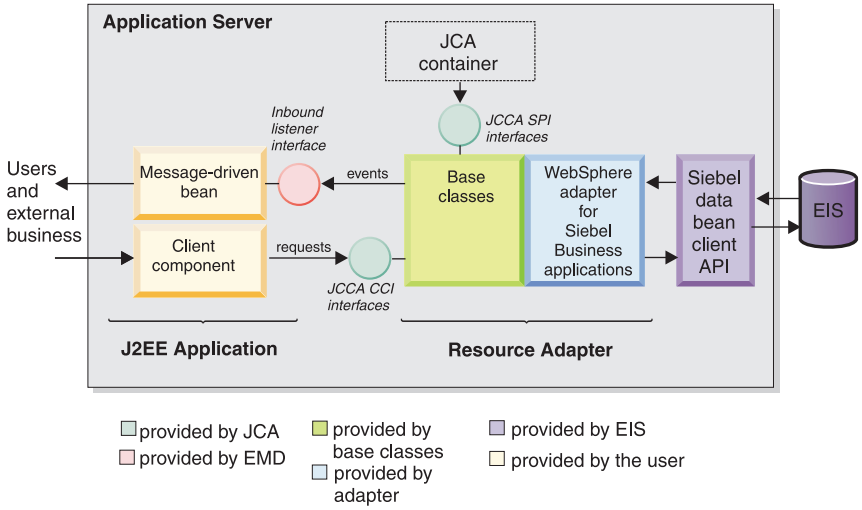


Figure 2. Adapter architecture diagram

Business objects

The Adapter for Siebel Business Applications supports two types of business objects: Siebel business services with Siebel integration objects and Siebel business objects and components.

Siebel business objects and components

Siebel business objects and components are object-oriented building blocks of Siebel applications and can be customized.

How the adapter processes Siebel business objects and components

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

Siebel business objects are object-oriented building blocks of Siebel applications and can be customized. The business objects define the relationships between business component objects (BusComps) and contain semantic information about sales, marketing, and service-related entities. A Siebel business object groups one or more business components into a logical unit of information. Some examples of Siebel business objects include: Opportunity, Quote, Campaign, and Forecast. An opportunity business object might consist of opportunity, contact, and product business components. The opportunity business component dictates the information of the other business components in a parent-child relationship.

A business component defines the structure, the behavior, and the information displayed by a particular subject such as a product, contact, or account. Siebel business components are logical abstractions of one or more database tables. The information stored in a business component is usually specific to a particular subject and is typically not dependent on other business components. Business components can be used in one or more business objects.

The Adapter for Siebel Business Applications is metadata driven. The adapter supports hierarchical business objects. The information about the processed object is stored in the application-specific information for the object and each of its attributes. The adapter supports the following operations:

- **ApplyChanges** Updates the business component based on the delta information.
- **Create** Creates the business object.
- **Delete** Deletes the business object and its children from Siebel (a physical delete). By default it is a cascade delete in Siebel.
- **Exists** Checks for the existence of the incoming business objects in Siebel. The output business object, *ExistsResult*, is returned with the Boolean value populated.
- **Retrieve** Shows the values of the business object.
- **RetrieveAll** Retrieves multiple instances of the same business object and populates it as the container BG (business graph) and return it back.
- **Update** Updates the Siebel application with the incoming object.

Business object naming convention for business objects representing Siebel business objects:

The naming convention for the business objects representing Siebel business objects includes the concatenation of several words.

The naming convention for business objects representing Siebel business objects are the same for both inbound and outbound processing as seen in the following example:

```
<Prefix><BO><Business Object Name> + <BC><Business Component Name>
```

Table 1. Business object naming formats and descriptions

Business object naming format	Description
Prefix	An optional prefix, which is placed at the start of the business object name.
Business Object Name	The Siebel business object name under which the business components are grouped.
Business Component Name	The Siebel business component name.

Note: The Siebel business object and component names are stripped of all non-alpha-numeric characters before being added to the WebSphere Business Integration business object name. If the resultant names have uniqueness issues, a counter is suffixed to the names. For example, if two WebSphere Business Integration business objects end up having the name 'SiebelBOAccountBCBusinessAddress', the names are suffixed with the counter to get unique values, for example, SiebelBOAccountBCAddress1, SiebelBOAccountBCAddress2.

The top-level business object will also have a business graph generated. The business graph name has the following format:

```
<Prefix><BO>+<Business Object Name>+<BC>+<Business Component Name>+BG
```

There is also a container business object generated for supporting the RetrieveAll operation. The complex type within the container object is the top-level business graph. The container business object has the following format:

```
<Prefix><BO>+<Business Object Name>+<BC>+<Business Component Name>+Container
```

The business object generated for the Exists operation has the following format:

```
Business Object:
<Prefix><ExistsResult>
```

The business object will contain one attribute, Status, of the type, Boolean, showing the return status.

Note: A business graph is generated for the corresponding ExistsResult business object because there is no need.

Business object attribute properties:

Business object attributes store important Name, Type, and Key information about the business object. They are set during enterprise service discovery and can be reset in the WebSphere Process Server administrative console.

The business object attribute properties table shown below describes the attributes used by the adapter.

Table 2. Business object properties

Property	Description
Cardinality	For simple attributes, 1 is used. For container properties, depending on the method requirements, n is used.
Key and foreign key	For business components, the ID is the primary key. For business services and integration objects, these attributes are not used.
Name	Contains the name of the attribute.
Required	This attributes is not used by the adapter.
Special	None.

Table 2. Business object properties (continued)

Property	Description
Type	Business services, integration objects, and business components, are supported by multiple types of properties, for example, int, double, string or complex types representing an integration object. Complex types can be either an integration component or a Siebel business component.

Business object metadata:

Business object metadata provides the adapter with application-dependent instructions on how to process business objects.

Table 3. Business object component metadata

Parameter	Description
ComponentName	The name of the Siebel business component corresponding to the WebSphere Business Integration adapter business object.
ObjectName	The name of the Siebel business object corresponding to the WebSphere Business Integration adapter business object.

Table 4. Property metadata

Parameter	Description	Globalized
FieldName	The field name of the field in the Siebel business component corresponding to this attribute.	Yes
PickListKey	If a simple attribute is a PickList, a PickListKey is specified.	Yes
Restrict=<attrname name>,<attr name>,<attr name>	Restrict is additional search criteria that a user can set to limit the retrieved records.	No

Table 5. Container attribute-level application text

Parameter	Description	Globalized
FieldName	Field Name representing the multi value Not used with simple links.	Yes
MultiValueLink	MVL set to Active specifies a one to many relationship. Setting this to Inactive indicates that there is an inactive multi value link relationship between parent and child object (ie there is no multi value field on the parent object).	No

Table 5. Container attribute-level application text (continued)

Parameter	Description	Globalized
PickList	Set to true indicates a many to one relationship.	No
Association	Set to true specifies that the relationship is a many to many relationship through an intersection table.	No
From=...,;To=...	Preprocessing instructions to the adapter to set the To attribute to the value of the From attribute. The From attribute must be populated. The To attribute is set only if it is null. The objects containing the attributes must have a one to one relationship. This is used in a Retrieve operation and used to specify which child record needs to be fetched.	No
SourceField=...;Destination Field...	This is used with simple links. The source field (SF) corresponds to the key attribute (Id) in the WebSphere Business Integration adapter parent business object. The destination field (DF) is the foreign-key field in the Siebel child business component. This corresponds to the child attribute with the field name set to the foreign-key field in the Siebel business component. If SourceField and DestinationField are null in the Siebel repository, the enterprise metadata discovery (EMD) ignores the corresponding attribute for simple link purposes, for example, it will not generate it as a simple link attribute. In such a case, there is an intersection table and the corresponding attribute becomes an association, for example, covered by the association type application-specific information (ASI). In addition, if the SourceField is null (but not the DestinationField), the SourceField is set as Id.	No

Table 5. Container attribute-level application text (continued)

Parameter	Description	Globalized
KeepRelations	This is a Boolean application-specific information (ASI). This information decides whether a particular child object needs to be deleted from the enterprise information system or not. This is used only for Siebel business objects (BO) and business components (BC), for example, not for business services and inputOutput (IO).	No

Supported verbs:

Verbs are used in the inbound support interactions.

Outbound supported verbs for Siebel business objects include the following:

- Create
- Delete
- Update
- UpdateWithDelete

Note: For outbound interactions, verbs are ignored.

For inbound interactions the following verbs are supported:

- Create
- Delete
- Update
-

Siebel business services

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

How the adapter processes Siebel business services

A Siebel business service is an entity in Siebel that encapsulates and simplifies the use of some sets of functionality, such as synchronizing between the Siebel application and external applications. With Siebel business services, developers can encapsulate business logic in a central location, abstracting the logic from the data it might act upon.

A business service is like an object in an object-oriented programming language, it has properties and methods and maintains a state. Methods take arguments that can be passed into the object programmatically or declaratively by way of workflows.

A service component architecture (SCA) module, as a client of the adapter, executes a request using the connection specification of the adapter. The adapter uses the input business graph to determine which Siebel business service method to invoke.

The adapter creates a copy of the input business graph, which is populated with results as the output business graph. The adapter builds the required Siebel property set based on the input business graph and invokes the business service's method. The Siebel property set output is then populated in the output business graph. The output business graph is converted to a Common Client Interface record implementation, which is returned to the calling client.

The adapter supports methods on the generic business services, custom business services, and application services interfaces. The adapter also supports the Siebel enterprise application integration (EAI) adapter and the Siebel application services built-in interfaces.

The adapter only supports business services that are of the class type CSSEAIDataSyncService and CSSEAIDTEScriptService, and CSSEAISiebelAdapter and CSSService.

The adapter supports EAI Siebel Adapter, a built-in Siebel business service is a general purpose business service that allows for data synchronization, Siebel Application Services, and any custom business services based on the supported class types.

Business object naming convention for business objects representing Siebel business services:

The naming convention for business objects representing Siebel business services are the same for both inbound and outbound processing.

The naming convention for the business objects includes the concatenation of several words, as seen in the following table:

Table 6. Business object naming formats and descriptions

Business object naming format	Description
Prefix	The *Prefix* property value is prefixed to the names of the WebSphere business objects generated against the following: • Siebel business objects and components, for both inbound and outbound processing. • Siebel Business services, outbound processing only.
Business Service Name	The business service for the business object.
Integration Object	The underlying integration object for the Siebel message container business object.
Integration Component	The underlying Integration component for the Siebel Message container business object.

Outbound

The integration object, top-level business object, has the following naming convention:

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

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

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

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

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

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

```
<Top Level business object Name> + BG.
```

Examples

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

```
IBMEAISiebelAdapterInsertAccountInterfaceBusinessAddressInterface
```

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

```
 IBMEAISiebelAdapterInsertAccountInterfaceBusinessAddressInterfaceBG
```

The integration-component level for the outbound objects generated against integration components, have the following naming convention:

```
'IO' + <Name of Integration Object> + 'IC' + <Name of integration component>
```

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

Inbound

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

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

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

Business object structure:

The following business object structures show a method with the same integration object as the input and output, an inbound event business object based on the account interface integration object, and a method with a different integration object as the input and output.

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

The following naming conventions apply to the diagrams showing methods, inputs, and outputs.

- <Prefix> - Prefix as stated in enterprise metadata discovery (EMD)
- <BSN> - The business service name for the business object
- <Method> - The method the business object was generated against
- <IO> - The underlying integration object is chosen to be used in the method

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

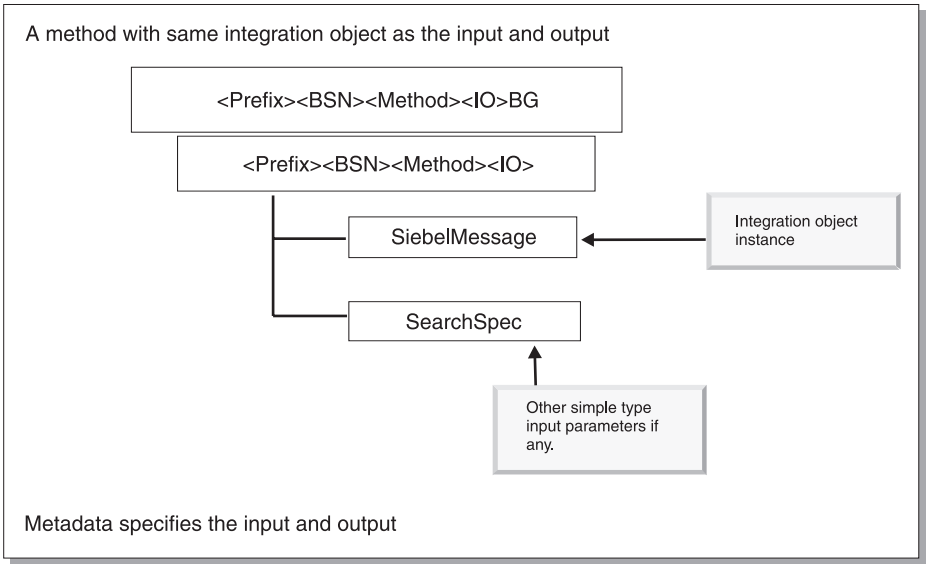


Figure 3. A method with the same integration object as the input and output

An inbound business object based on the account interface integration object

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

The following diagram represents a business object for an inbound event. The top-level data represents both input and output arguments with the Siebel message as a container. This same business object can be used for both request and response to and from the adapter to interact with the underlying Siebel enterprise information system (EIS). This means that the same business object type that you send in as a request is returned as the result of the execution.

The Siebel message is a wrapper similar to the wrapper the Siebel EIS uses to wrap integration objects (IO) and their respective fields and components within business services, as shown below.

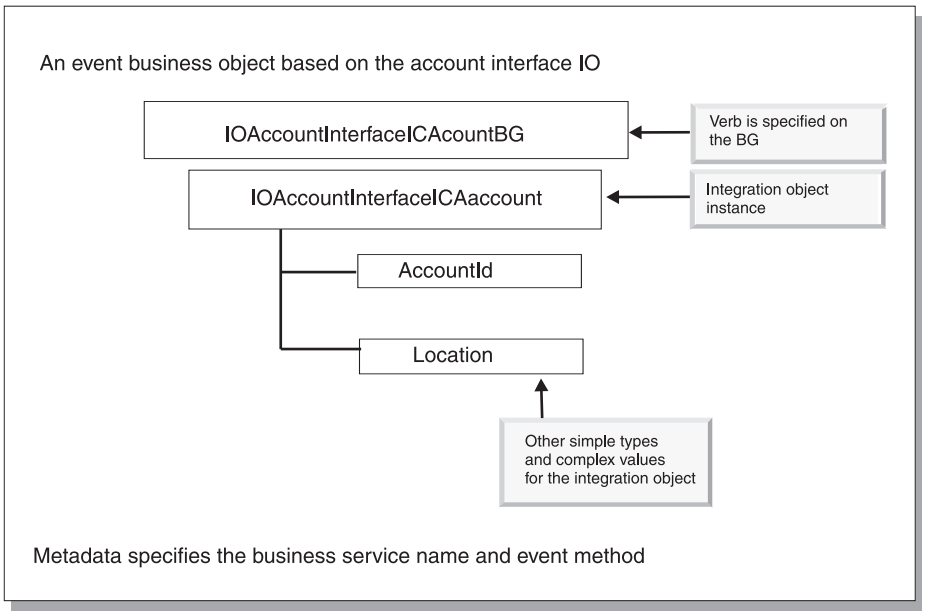


Figure 4. An event business object based on the account interface IO

A method with a different integration object as the input and output

The following diagram corresponds to a custom business service with a different integration object as the method's input and output.

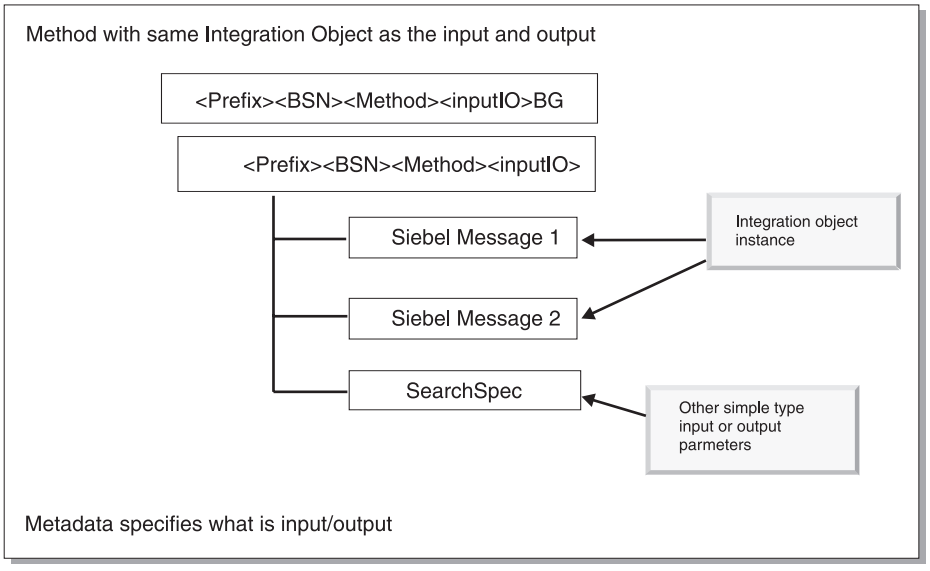


Figure 5. A method with the same integration object as the input and output

Business object attribute properties:

Business object attributes store important Name, Type, and Key information about the business object. They are set during enterprise service discovery and can be reset in the WebSphere Process Server administrative console.

The business object attribute properties table shown below describes the attributes used by the adapter.

Table 7. Business object properties

Property	Description
Cardinality	For simple attributes, 1 is used. For container properties, depending on the method requirements, n is used.
Key and foreign key	For business components, the ID is the primary key. For business services and integration objects, these attributes are not used.
Name	Contains the name of the attribute.
Required	This attributes is not used by the adapter.
Special	None.
Type	Business services, integration objects, and business components, are supported by multiple types of properties, for example, int, double, string or complex types representing an integration object. Complex types can be either an integration component or a Siebel business component.

Business object metadata:

Business object metadata provides the adapter with application-dependent instructions on how to process business objects.

Table 8. Business object application-specific text

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

Table 9. Property metadata

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

Note: There is no verb metadata.

Business object metadata schema

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

Supported verbs:

Outbound and inbound verbs are supported by the adapter.

Outbound supported verbs for Siebel business objects include the following:

- Create
- Delete
- Update
- UpdateWithDelete

For inbound interactions, the following verbs are supported on the integration-object Siebel business objects.

- Create
- Delete
- Update

Enterprise service discovery

The enterprise service discovery wizard consists of two components, the enterprise metadata discovery implementation in the enterprise application integration (EAI) tooling and the adapter. After an adapter is imported into the EAI tooling

environment, such as WebSphere Integration Developer, you can run the enterprise service discovery wizard and automatically generate the service component artifacts and business objects needed for an integration solution that runs on WebSphere Process Server.

By selecting enterprise information system entities from the metadata tree structure nodes, you can generate business objects for enterprise information system or database entities. You can browse the metadata information of an enterprise information system or database, select the entities of interest, and generate deployable service objects and descriptions. The metadata is transformed into service data objects consisting business graphs and business objects.

Note: Service component artifacts are a physical piece of information that is used or produced by a software development process. Examples of artifacts include models, source files, scripts, and binary executable files.

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

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

Note: The description of a Web service can be defined in any format such as WSDL, UDDI, or HTML.

Outbound processing

For outbound request processing, an external J2EE client invokes the adapter through the Service Component Architecture (SCA). The client then requests an execution, which in turn is passed from the adapter to the enterprise information system (EIS). With outbound processing, the client can make calls to the adapter to perform specific operations in the EIS file system.

The Adapter for Siebel Business Applications is metadata driven for outbound processing for business objects and business components. The adapter supports hierarchical business objects. The information about the object about to be processed is stored in the application-specific information for the object and each of its attributes.

Processing business objects modeled on Siebel business objects

The adapter supports the Create, Update, Delete, RetrieveAll, Retrieve, Exists, and ApplyChanges operations. The Delete operation is a physical delete. The adapter uses the Retrieve operation for inbound operations to retrieve the objects from the Siebel application. The adapter foundation classes command pattern is used to perform the operations. The execution order name is AFTER_PARENT.

Processing business objects modeled on Siebel Business services:

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

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

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

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

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

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

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

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

Supported outbound operations

For business object and business components, the adapter supports the ApplyChanges, Create, Delete, Exists, Retrieve, RetrieveAll, and Update operations.

The Adapter for Siebel Business Applications is metadata driven. For outbound processing for business objects and business components, the adapter supports

hierarchical business objects. The information about the object about to be processed is stored in the application-specific information for the object and each of its attributes.

ApplyChanges:

The ApplyChanges operation sends the Create, Update, and Delete business objects to the adapter for processing.

By using the ApplyChanges operation, the verbs can be set to either Empty, Create, Update, Delete, or Updatewithdelete. After the verb is set, the adapter processes the verb.

If the verb is not set, the adapter checks the ChangeSummary record and performs the required operation. The ChangeSummary record includes business object-level creates and deletes.

The ChangeSummary record is needed only when the top-level verb is not set and the requested operation is ApplyChanges. The only exception to this process occurs when the top-level verb is UpdateWithDelete and the operation is ApplyChanges. In this case, the ChangeSummary record is set at the top-level to Update, otherwise, the adapter throws an error. In addition, if there are any failures, the exception, SiebelApplyChangesFailedException, is thrown.

Create:

The Create operation creates a new entry for the business component in the Siebel application.

Each of the child container attributes are processed as follows:

- The keys are not set on the parent business component. Siebel generates the keys on the creation of a record.
- A check is made to look for the type of relationship between the parent and the child. If the child container is a PickList business component and the keys are set, the adapter tries to fetch the record. When the record in the Siebel application is not found, the adapter errors out the operation saying the pick record is not found. If the non-keys are populated, the adapter tries to create the record and pick the record.
- If the PickList is static, the corresponding simple attribute in the WebSphere Business Integration business object is a required attribute. If a value is not listed in the PickList values sent, the adapter tries to set the value. If the value is successfully set the list is unbounded and insert is enabled. This new value is then picked. If the value is not successfully set, the verification fails and the adapter fails the create operation.

  In the following PickList properties table, the Create operation fails if the PickList property is bounded or unbounded, but insert is enabled. An error from the Siebel application is logged, the adapter fails the Create operation, and the SiebelCreateFailedException is thrown.

  Using the PickList property combinations shown in the PickList properties table, you can create, update, and delete a record from a Siebel PickList.

Table 10. PickList properties

PickList property	Insert	Update	Delete
UnBounded	Allowed if "No Insert" property is not set.	Allowed if "No Update" property is not set.	Allowed if "No Delete" property is not set.
Bounded	Not allowed.	Not allowed.	Not allowed.

- With a multi-value link, the child record is created. Explicit writeRecord() on the child is not required in case of a multi-value link. If the child fails, the child is rolled back. If the parent fails, all the children and the parent are rolled back.
- With association, a check based on all the attributes filled in is made to see if the record exists in the association business component. If the record is found, the record is associated with the parent. If the record is not found, the record is created in the association business component and associated with the parent business component.

  If the child creation fails, the child is rolled back. If the parent creation fails, the children that are already created are left as is and the parent alone is rolled back. This is because association components can be treated as individual components.
- In the case of a simple link, the child gets created before the parent. If the child or the parent creation fails the entire transaction is rolled back.

  In addition:
  - The field values are set for simple attributes.
  - Saving needs to be done for each of the business component records.
  - If any of the attributes are required and they are not set in the incoming business object, the adapter throws the exception, SiebelRequiredAttributeNotFoundException.
  - If there is any failure, the exception, SiebelCreateFailedException, is thrown.
  - The output business object should be populated with the keys.
  - Each of the children are created first, and then the parent is created. The saving of the parent is done after all the children are created. Saving after creating all the children has the advantage of rolling back all the child creates in case the top-level create fails.

Delete:

The Delete operation performs a physical delete and the record is removed from the underlying database.

Only the parent needs to be deleted and Siebel cascade deletes all the children. If any of the key attributes are missing from the incoming WebSphere Business Integration adapter business object, the delete fails.

Exists:

For the Exists operation, you must check for the existence of the primary business component. The keys are set on this business component and the query is executed.

If the Exists operation record exists, the special business object, ExistsResult, is returned containing the status of the Exists operation. For any failure, the exception, BusinessProcessingFailedException, is thrown.

For the Exists operation, the following apply:

- The keys are not set on the parent business component. Siebel generates the keys on the creation of a record.
- A check is made to look for the type of relationship between the parent and the child. If the child container is a PickList business component and the keys are set, the adapter tries to fetch the record. When the record in the Siebel application is not found, the adapter errors out the operation saying the pick record is not found. If the non-keys are populated, the adapter tries to create the record and pick the record.
- If the PickList is static, the corresponding simple attribute in the WebSphere Business Integration business object is a required attribute. If a value is not listed in the PickList values sent, the adapter tries to set the value. If the value is successfully set the list is unbounded and insert is enabled. This new value is then picked. If the value is not successfully set, the verification fails and the adapter fails the create operation.

  In the following PickList properties table, the Create operation fails if the PickList property is bounded or unbounded, but insert is enabled. An error from Siebel application is logged and the adapter fails the Create operation and SiebelCreateFailedException is thrown.

  Using the combinations of properties shown in the PickList properties table, you can create, update, and delete a record from a Siebel PickList.

Table 11. PickList properties

PickList property	Insert	Update	Delete
UnBounded	Allowed if "No Insert" property is not set.	Allowed if "No Update" property is not set.	Allowed if "No Delete" property is not set.
Bounded	Not allowed.	Not allowed.	Not allowed.

- With a multi-value link, the child record is created. Explicit writeRecord() on the child is not required in case of a multi-value link. If the child fails, the child is rolled back. If the parent fails, all the children and the parent are rolled back.
- With association, a check based on all the attributes filled in is made to see if the record exists in the association business component. If the record is found, the record is associated with the parent. If the record is not found, the record is created in the association business component and associated with the parent business component.

  If the child creation fails, the child is rolled back. If the parent creation fails, the children that are already created are left as is and the parent alone is rolled back. This is because association components can be treated as individual components.
- In the case of a simple link, the child gets created before the parent. If the child or the parent creation fails the entire transaction is rolled back.

  In addition:

  – The field values are set for simple attributes.
  – Saving needs to be done for each of the business component records.
  – If any of the attributes are required and they are not set in the incoming business object, the adapter throws the exception, SiebelRequiredAttributeNotFoundException.
  – If there is any failure, the exception, SiebelCreateFailedException, is thrown.
  – The output business object should be populated with the keys.

– Each of the children are created first, and then the parent is created. The saving of the parent is done after all the children are created. Saving after creating all the children has the advantage of rolling back all the child creates in case the top-level create fails.

Retrieve:

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

For the Retrieve operation, the following applies:

- The primary business component is pruned for its children. Pruning trims off the child containers from the incoming WebSphere Business Integration adapter business object and sets the container attribute to null.
- The keys are set on the top level business component for searching the records.
- The children are obtained based on the type of relationship.
- The From and To application-specific information is used as adapter preprocessing commands to set the keys. In the case of a static PickList, verification as to whether the sent-in value is part of the existing PickList is made. In the case of multi-field PickLists, the ID key specified in the application-specific information at the container level is used to get the PickList business component, pick the records, and set the records in the parent business object. If the record is not found, an error message is logged and the retrieve fails with an exception. With multi-value link and association, if the child record is found, the child record is set on the parent.

RetrieveAll:

The adapter supports the RetrieveAll operation request, even when non-keys are set on the primary business component. When none of the attributes are set, an asterisk (*) is selected.

If the values of the attributes, which are part of the search criteria, contain an asterisk (*), it is treated as a RetrieveAll for that attribute. Any additional search criteria is applied if it is present.

The returned business object is a container business object of the business graphs.

The steps involved in processing the RetrieveAll operation is similar to the Retrieve operation, except the check is not made to see whether all the keys are set in the incoming WebSphere Business Integration adapter business object.

Note:

The adapter can process the values of the attributes contained in the special characters, which include ‘(‘, ‘)’, “””, “’”, “,”,”-“,”<”,”>”. Only these special characters are processed by the adapter. This applies to all the operations.

It is important to note that the number of records that the adapter can return cannot exceed the MaxRecords property value of the WBIInteractionSpec property instance. If the number of records retrieved from the enterprise information system exceeds the MaxRecords property

value, the exception, MatchesExceededLimitException, is thrown. In addition, if no records are retrieved, the exception, RecordNotFoundException, is thrown.

Business service support

The business object is processed during outbound calls by processing the properties of the top-level object. Each business object property that is marked as an input property is used in the method invocation on the Siebel business service, if that property is set in the business object instance.

The Siebel message container, which represents an integration object, is also processed. This object consists of the attributes and metadata necessary to create the Siebel property sets (PropertySets) that are required during business service execution.

The results of the business service execution are Siebel property sets (PropertySets) and, or, simple types. These values are then placed in their corresponding output business object properties in the top-level object. The Siebel message-container business object is populated by the corresponding child property set (PropertySet). The child property set is traversed, and the object is populated, by using the metadata and field names specified in the result. The Siebel message-container business object is populated with the obtained property set. The returned output property set is based on the status keys defined in the integration object. Only those particular fields are present in the output property set.

For Example, in the case of an integration object account interface and the account integration component beneath, one status key is defined and the fields for that key are defined. The defined fields include Account Id, Integration Id, and operation. The output property set contains values for only those fields. The entire business graph is then returned to the client after being populated with the results from the business service execution.

Update:

The Update operation involves comparing the business object retrieved from the Siebel application with the incoming WebSphere Business Integration adapter business object.

The Update operation process involves setting the correct verb on the child objects and processing the object.

For the Update operation, the following applies:

1. The default behavior compares the object retrieved from the Siebel application with the incoming adapter business object.
2. The process of creating, updating, and deleting children is done based on the comparison above. As a result of the comparison, all the children in the Siebel application are processed and made the same as the inbound adapter business object. If the process to create the child fails because the record already exists, the exception, RecordAlreadyExistsException, is thrown.
3. In addition, for all failures, the exception, SiebelUpdateFailedException, is thrown.

KeepRelations support

When an incoming *after-image* business object has an update verb, the adapter does not generate delete commands for the missing child business objects (child business objects that exist in the target application, but not in the incoming after-image). If a source application provides an incomplete *after-image*, one or more missing children are kept.

The adapter supports the application-specific information (ASI) for the attribute that represents the child or array of children. The Boolean ASI tag, KeepRelations, needs to be set to true. Instead of creating delete commands for such children, the adapter generates *nooperation* command instances.

The adapter processes an *after-image* business object structure in the following way:

1. It determines whether the top-level object is an update.
2. If so, for each missing child object, the adapter determines whether the *KeepRelations* ASI is set to true in the attribute's ASI container
3. If so, the adapter generates a *nooperation* command instead of a delete command.

Note: There are no specific code changes in the adapter, except to add the ASI tag to the metadata schema. The adapter foundation classes provide the actual code support for the metadata schema.

Inbound processing

The adapter for Siebel Business Application supports inbound request processing. Inbound events are supported by a Siebel event component and the adapter polls the event component at regular intervals. Messages are propagated to endpoints that register for the events.

Inbound events for Siebel business objects

For inbound events, the Siebel event store component lists all the events. When these events are retrieved by the adapter, the business component that is represented by the event is retrieved using the Retrieve operation. The business component values are populated in the business graph and are published to the registered endpoints. The supported event types are Create, Update, and Delete.

Inbound events for Siebel business services

Inbound processing consists of the following steps:

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

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

Event store

An event store, also known as an event component, is a business component in Siebel where entries for each event are stored until they are processed by the adapter.

The back-end enterprise information system generates the events in the form of event records. These records are stored in the Siebel event component configured earlier. This event information is used by the adapter during event subscription to build the corresponding business objects and send them to their registered endpoints.

With inbound processing, the adapter polls the event records from the event component at regular intervals. In each poll call, a number of events are processed by the adapter. The order of event processing is based on the ascending order of priority and the ascending order of the event time stamp. The events with the status, Ready for poll, are picked up for polling in each poll cycle. The adapter uses the object name and object key to retrieve the integration object or business object. The business graph is created from the retrieved information and is published to the endpoints.

If an event is successfully posted, the entry is deleted from the event table. For failed events, the entries remain in the event table. For the event type, Delete, the keys are set on the data object, the business graph is created and published to the endpoints, and the object is not retrieved from Siebel.

The structure of the Siebel event table, used internally by the adapter, is as follows.

Table 12. Event table structure

Field	Description	Example
Description	Any comment associated with the event.	Account Interface event
Event ID	The ID of the event row.	Automatically generated unique ID in Siebel
Event time stamp	The time stamp for the event. The format is *mm/dd/yyy hh:mm:ss*.	02/24/2005 11:37:56
Event type	The type of event.	Create
Object key	A unique identifier of the business object row for which the event was created. It is a name value pair consisting of the name of the property and the value. The object key format for the Siebel business object is id=XXXX.	ID=1-CT8
Object name	The business graph for which the event was detected.	EAIAccountInterfaceBG
Priority	The event priority.	1
Status	The event status. This is initially set to READY_FOR_POLL.	0
XID	The transaction ID.	None.

As events are retrieved and processed from the event table, the status of the event changes to reflect the cycle, as shown in the table below.

Table 13. Event status values

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

Globalization and bidirectional transformation

This adapter is globalized to support single- and multi-byte character sets and deliver message text in the specified language. The adapter also performs bidirectional transformation, which refers 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.

Globalization

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

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

Bidirectional 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, standards are used to display and process it. WebSphere Process Server uses the Windows standard format, but an enterprise information system exchanging data with WebSphere Process Server can use a different format. WebSphere Adapters transform bidirectional script data passed between the two systems so that it is accurately processed and displayed on both sides of a transaction.

WebSphere Process Server bidirectional format

WebSphere Process Server and WebSphere Enterprise Service Bus 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 Process 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.

Bidirectional format attributes

Letter position	Purpose	Values	Description	Default setting
1	Order schema	I or V	Implicit (Logical) or Visual	I
2	Direction	L R C D	Left-to-Right, Right-to-Left Contextual, Left-to-Right Contextual, Right-to-Left	L
3	Symmetric Swapping	Y or N	Symmetric Swapping is on or off	Y
4	Shaping	S N I M F B	Text is shaped Text not shaped Initial shaping Middle shaping Final shaping Isolated shaping	N
5	Numeric Shaping	H C N	Hindi Contextual Nominal	N

The adapter transforms data into a logical, left-to-right format before sending the data to WebSphere Process Server.

Using bidirectional properties

You can use multiple bidirectional properties to control the transformation of both content data and metadata. You can set special bidirectional properties to exclude either content data or metadata from bidirectional transformation, or to identify data that requires special treatment during a transformation.

The following table describes four types of bidirectional properties.

Bidirectional property types

Property type	Data transformations
EIS	Controls the format for content data, or data that is sent by the enterprise information system.
Metadata	Controls the format for metadata, or data that provides information about the content data.
Skip	Identifies content or metadata to exclude from transformation.

Property type	Data transformations
Special Format	Identifies certain text, such as file paths or URLs, that require different treatment during the transformation process. Can be set for either content data or metadata.

You can set properties that control bidirectional transformation in three areas.

- **Resource adapter properties:** These properties store default configuration settings, including the TurnBiDiOff property, which controls whether the adapter instance performs bidirectional transformation or not. Use theWebSphere Process Server administrative console to configure these properties.
- **ManagedConnectionFactory (MCF) properties:** These properties are used at run time to create an outbound connection instance with an enterprise information system. Once the MCF properties are created, they are stored in the deployment descriptor.
- **Activation Specification properties:** These properties hold the inbound event processing configuration information for a message endpoint. Set them as you perform enterprise service discovery, or use the WebSphere Process Server administrative console.

Business object annotations

Some adapters allow you to annotate bidirectional properties within a business object. Do this to add information that specifically controls the transformation of a business object or part of a business object. Use business object editor, a tool within WebSphere Integration Developer, to add annotations at these levels:

- Business object
- Business object application-specific attribute
- Business object attribute
- Business object attribute application-specific attribute

Property scope and lookup mechanism

After you set values for bidirectional properties for an adapter and annotate business objects where appropriate, the adapter performs bidirectional transformations. It does so by using logic that relies on a hierarchical inheritance of property settings and a lookup mechanism.

Properties defined within the resource adapter are at the top of the hierarchy, while those defined within other areas or annotated within a business object are at lower levels of the hierarchy. So for example, if you only set values for EIS-type bidirectional properties for the resource adapter, those values are inherited and used by transformations that require a defined EIS-type bidirectional property whether they arise from an inbound (Activation Specification) transaction or an outbound (MCF) transaction.

However, if you set values for EIS-type bidirectional properties for both the resource adapter and the Activation Specification, a transformation arising from an inbound transaction uses the values set for the activation specification.

The processing logic uses a lookup mechanism to search for bidirectional property values to use during a transformation. The lookup mechanism begins its search at the level where the transformation arises and searches upward through the

hierarchy for defined values of the appropriate property type. It uses the first valid value it finds. It searches the hierarchy from child to parent only; siblings are not considered in the search.

Chapter 5. Planning for adapter implementation

Before you begin your installation, you must consider several factors, such as your adapter environment, security and performance needs, and whether you need locale or globalization support.

Security

The adapter is Java 2 security enabled and features user name and password authentication. In addition, you can configure additional security permissions by altering the application server's WAS.policy file and storing it in the meta-inf folder. For more details on configuring security details, see the security documentation for WebSphere Process Server.

WebSphere Adapters in clustered environments

You can improve adapter performance and availability by deploying the WebSphere adapter enterprise archive (EAR) module to a clustered server environment. The adapter instance within the EAR module is replicated across federated servers.

WebSphere Process Server and WebSphere Application Server Network 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 JCA container to activate the adapter instance. The JCA container provides a runtime environment for adapter instances. For more information about clustered environments, see http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/trun_wlm_cluster_v61.html.

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

High availability for inbound operations

Inbound operations are based on events triggered as a result of updates to data in the enterprise information system (EIS) application. The adapter is configured to detect updates through event listeners or by polling an event table. The adapter then publishes the event to its endpoint.

In a clustered environment, two or more adapter instances might detect the same event. This scenario raises the possibility of duplicate event processing or data infidelity. For example, if two adapter instances are simultaneously polling the same event table with the same event type filter, one may alter data that the other adapter instance depends on, or it might fail. There is a parallel risk for event-listening adapter architectures in a clustered environment.

To avoid this condition, the HAManager for the inbound adapter instances enforces a singleton behavior. Even though all the adapter instances are started, only one of the instances detects and publishes an event to the endpoint for each type of EIS application.

When you deploy an adapter module to a cluster, the JCA container checks the enableHASupport property of the ResourceAdapter bean. If the value for the enableHASupport property is true, the JCA container registers all of the adapter instances with HAManager with a policy 1 of N. This policy means that only one of the clustered servers starts event polling (or listening) for this adapter instance. 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.

High availability for outbound operations

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

WebSphere Application Server Network Deployment has a workload management capability that distributes the outbound processing among the adapter instances. As a result, outbound operations in a clustered environment are similar to those in a single server environment: one adapter instance processes only one outbound request at a time. For more information on workload management, see http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/trun_wlm.html.

Note: Adapter instances are replicated in a clustered server environment. When the enableHASupport property is set to true, which is the default setting, only one of the replicated adapter instances actively polls for events while other instances are in standby mode. If the enableHASupport property is set to false, all of the adapter instances replicated on cluster members actively poll for events. This may result in event duplication. Do not change the value of enableHASupport to false for single server environments. For information on changing the value of this property, see the Resource adapter properties section in this documentation. To determine whether adapter replication is supported in a clustered environment, see the software and hardware requirements section of this documentation.

Roadmap for installing, configuring, and deploying the adapter

Before you can use the adapter in a runtime environment, you must install, configure, and deploy it. Understanding these tasks at a high level helps you perform the steps that are needed to accomplish each task.

After successfully installing the WebSphere Adapter, you configure it using WebSphere Integration Developer. You then deploy it as an enterprise archive (EAR) file to WebSphere Process Server or WebSphere Enterprise Service Bus. The following figure illustrates this flow of tasks, and the steps that follow the figure describe each task at a high-level. For detailed instructions on installing, see *Installing IBM WebSphere Adapters*. For information about configuring and deploying the adapter, see the adapter documentation.

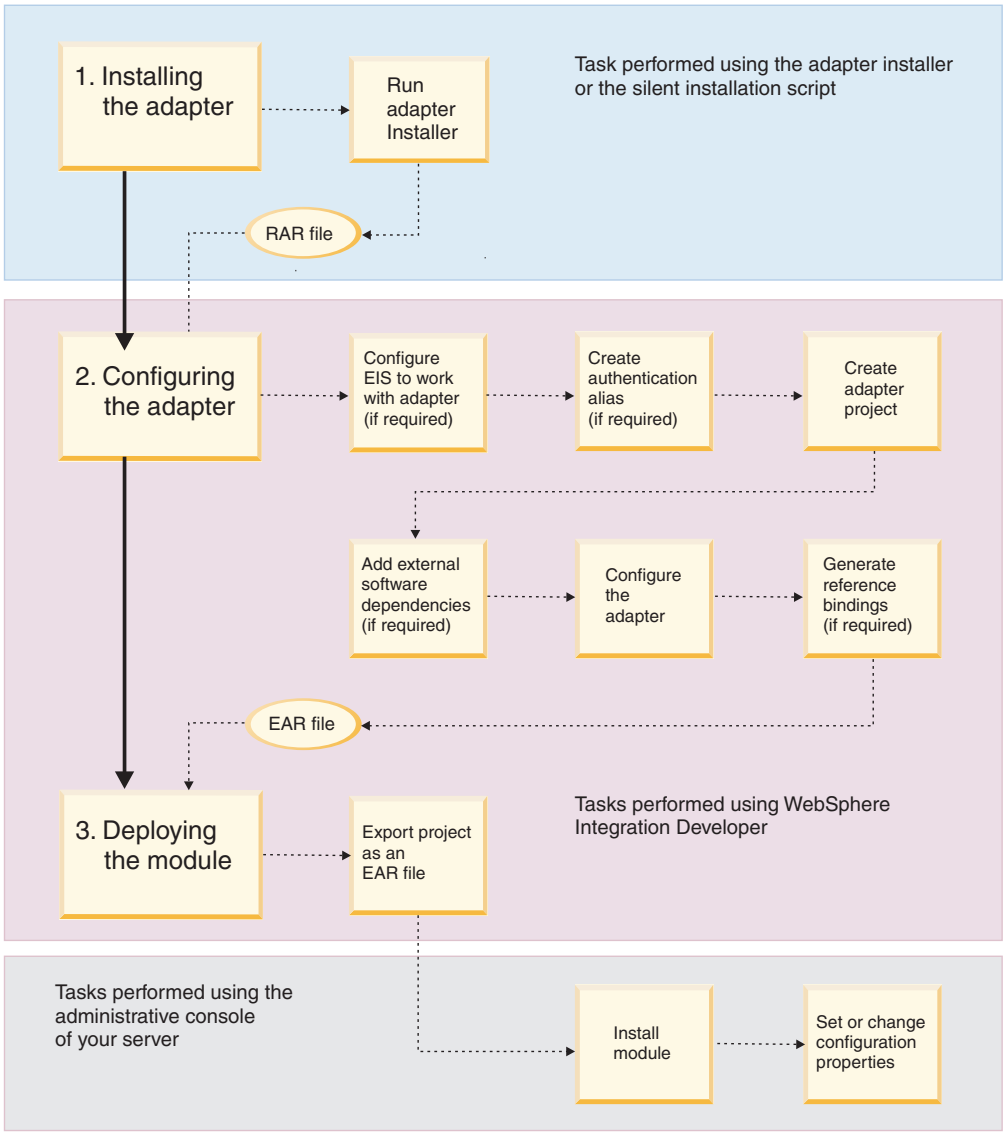


Figure 6. Roadmap for installing, configuring, and deploying the adapter

1. **Installing the adapter**
   a. Use the installer (a graphical user interface) or a script that runs a silent installation. Either method installs a resource adapter archive (RAR) file on your workstation. You use this RAR file to configure the adapter.
2. **Configuring the adapter**
   a. (If required) Configure the enterprise information system (EIS) to work with your adapter. You perform this step from within the EIS application.
   b. (If required) Create an authentication alias to access the application.
   c. Create an adapter project in WebSphere Integration Developer (J2EE Perspective) by importing the adapter RAR file.
   d. (If required) Using WebSphere Integration Developer, add any external dependencies required by your adapter to the adapter project. These dependencies are also required as part of the bundled EAR file, which is exported when you deploy the adapter.

e. To configure the adapter, run the enterprise service discovery wizard from the Business Integration Perspective of WebSphere Integration Developer. The enterprise service discovery wizard generates business integration components and allows you to enter all the information necessary to configure the adapter for the first time. The output from the enterprise service discovery tool is saved to a business integration module project, which contains the business object, or objects, and the import or export file.

f. (If required) Use WebSphere Integration Developer to generate reference bindings for the component created by the enterprise service discovery wizard.

3. **Deploying the module**

   a. From the J2EE perspective in WebSphere Integration Developer, export a business integration module project as an EAR file.

   b. Install the module on WebSphere Process Server or WebSphere Enterprise Service Bus.

   c. (If required) In the server administrative console, set (or change) the following properties:

      - Resource adapter properties
      - Managed (J2C) connection factory properties
      - Activation specification properties for the EIS

Chapter 6. Installing the adapter

To install the adapter, you must check system prerequisites, perform migration steps, then perform the installation steps common to all adapters.

Installation prerequisites

Before installing Adapter for Siebel Business Applications, you must verify that your environment meets all of the necessary hardware and software requirements. These requirements fall into two categories: supported platforms for running the adapter installer, and hardware and software requirements for configuring, deploying, and running the adapter.

Supported platforms for running the adapter installer

The supported platforms for running the adapter installer are located in the ″Installing″ section of Installing IBM WebSphere Adapters.

Hardware and software requirements for configuring, deploying, and running the adapter

The hardware and software requirements for configuring, deploying, and running the adapter are located at the following Web site: IBM WebSphere Adapters and IBM WebSphere Business Integration Adapters: software requirements. From the IBM WebSphere Adapters list, select the link for the Adapter for Siebel Business Applications, Version 6.0.2.

Additional jar files

If you are using WebSphere Integration Developer, version 6.0.1.1 or earlier, you must manually add three additional jar files to the classpath of the adapter project. For more information about how to do this, see “Adding jar files to WebSphere Integration Developer versions 6.0.1.1 and earlier” on page 152 in the Reference section.

Performing the installation

The steps for installing the adapter are the same for all WebSphere Adapters. You can install the adapter either by using a graphical user interface or by performing a silent installation.

Before you begin

Review the installation prerequisites.

How to perform this task

After performing the basic installation steps, you can configure the adapter.

Result

The resource adapter archive (RAR) file is copied to the workstation where the adapter is installed. If you accepted the default installation location, the RAR file is

placed in the following directory: C:\Program Files\IBM\ResourceAdapters\Siebel\adapter\Siebel\deploy\CWYEM_Siebel.rar.

What to do next

If you have no need to perform migration, go on to configure the adapter.

Migrating to version 6.0.2

When you migrate to version 6.0.2, you must pay special attention to backward compatibility prior to performing the migration.

Backward compatibility

The older business object definitions work as they are, no changes are needed. To update previous versions of the adapter, replace the RAR file, no further changes in the configuration or existing artifacts are required.

Performing the migration

Migrate to the latest version of the adapter by adding properties to the inbound service description EAR file and deploying the EAR file onto WebSphere Process Server.

About this task

To migrate to the latest version of the adapter, you must update the inbound service description with the latest event filtering properties to enable inbound filtering.

How to perform this task

1. In the imported EAR file, find the inbound service description, and add the following properties:
   - `<assuredOnceDelivery>true</assuredOnceDelivery>`
   - `<filterFutureEvents>false</filterFutureEvents>`
   - `<eventTypeFilter>xyz</eventTypeFilter>`

     Note: The xyz represents the actual object name on which the filtering must be done.

   The properties must specify a value, otherwise, there an error is thrown when you deploy the EAR file. You must add the *EventTypeFilter* property with a valid value. If you do not want to enable filtering on an event type, do not add it to the export.
2. Export the application as an EAR file by selecting **File → Export → EAR file**.
3. Type the default name of the EAR file.
4. Deploy the EAR file onto WebSphere Process Server.

Result

You have completed all the necessary steps for migration.

Uninstalling the adapter

The steps for uninstalling the adapter are the same for all WebSphere Adapters. You can uninstall the adapter either by using a graphical user interface or by performing a silent uninstallation.

About this task

Uninstalling the adapter may be a required task for troubleshooting an installation problem. The steps for uninstalling the adapter are located in the ″Uninstalling″ section of Installing WebSphere Adapters.

Note: If you need to uninstall an adapter that is already deployed, refer to the ″Additional adapter-related information you might need″ section of “Related product information” on page 152.

Chapter 7. Configuring the adapter for deployment

To configure WebSphere Adapter for Siebel Business Applications so that it can be deployed on WebSphere Process Server or WebSphere Enterprise Service Bus, use WebSphere Integration Developer to create an adapter project, add required files to the project, and specify the business objects you want to discover and the system on which you want to discover them.

Configuration prerequisites

Before you begin configuring the adapter, make sure the password for the authentication alias has been set in the server for processing outbound requests, and make sure that an event table in the Siebel application has been created for processing inbound events.

Creating the authentication alias

Create the authentication alias on the server by using the WebSphere Process Server administrative console. From the administrative console, configure the global security and set the password for the authentication alias, which is used to process inbound and outbound events.

Before you begin

If you have not done so already, create an authentication alias on the server for both inbound and outbound processing.

About this task

To create an authentication alias, use the following procedure.

How to perform this task

1. On the WebSphere administrative console ″Welcome page,″ click **Security** → **Global security**.
2. Under the Authentication heading, click **JAAS Configuration** → **J2C Authentication data**.
3. Click **New**.
4. Type the required information in the **Alias**, **User ID**, **Password**, and **Description** fields.

   Note: The user ID and password that you type are used to establish a connection to the enterprise information system for outbound processing.
5. Click **OK**, click **Save**, and then click **Save** again.

Creating an event component in the Siebel application

Create an event component, also known as an event table, in the Siebel application to process inbound operations.

About this task

If the deployed enterprise application is configured for inbound processing, it cannot be started unless the event component exits in the Siebel application. The

procedure shown for the event component uses the Siebel Sales Enterprise application as an example. Substitute all references to Siebel Sales Enterprise with the name of the Siebel application in use. For this event table example, the name, IBM2 Events, has been used. Based on your needs, this event component name can be changed.

Creating event table columns

Create the event table columns in the Siebel event table by using the event wizard in the Siebel application.

Before you begin

You must be familiar with the tools found in the Siebel application to create the event table.

About this task

An event table must be created in the Siebel application event component to track the events that occur in the Siebel enterprise information system.

How to perform this task

1. Create an IBM project and lock your project, according to the Siebel instructions. Except for event triggers, you must complete all Siebel customization under the new project.
2. If you are installing multiple connectors, create multiple tables with different names.
3. Using the new object wizard, create a standalone table named `CX_IBM2_Events`.
4. Create columns in your new table with the following names:
   - Column Name/User Name
   - Type
   - Length
   - Physical type
   - Required
   - Nullable
   - Status

The example below shows some typical column titles and along with their associated information.

Table 14. Event table column example

Column name/ user name	Type	Length	Physical type	Required	Nullable	Status
DESCRIP TION	Data (public)	255	Varchar	No	Yes	Active
EVENT_ ID	Data (public)	30	Varchar	Yes	N/A	Active
EVENT_ TIME STAMP	Data (public)	7	Date Time	Yes	N/A	Active
EVENT_ TYPE	Data (public)	20	Varchar	Yes	N/A	Active

Table 14. Event table column example (continued)

Column name/ user name	Type	Length	Physical type	Required	Nullable	Status
OBJECT_ KEY	Data (public)	255	Varchar	Yes	N/A	Active
OBJECT_ NAME	Data (public)	255	Varchar	Yes	N/A	Active
PRIORITY	Data (public)	10	Number	No	Yes	Active
STATUS	Data (public)	20	Number	Yes	N/A	Active
XID	Data (public)	255	Varchar	No	N/A	Active

What to do next

Create a business component in the event table.

Creating a business component

Create a business component called IBM2 Event in the event table.

1. Create a new Business Component (BC), **IBM2 Events**, based on your new table. All fields are single value fields.
2. Create a new business object (BO) named **IBM2 Events**.
3. Associate **IBM2 Event BC** to the **IBM2 Event BO**.
4. Create **IBM2 Event List View Applet** based on **IBM2 Event BC**.
5. Create **IBM2 Event List View** based on **IBM2 Event BO**.
6. Create **IBM2 Event Screen** and associate it to the **IBM2 Event List View** in screen view.
7. Create a page tab as follows:
   a. Access the **Application** → **Siebel Sales** → **Page tab**.
   b. Right-click and select **New Record** from the menu.
   c. Enter **IBM2 Event** as the screen name and **IBM2 Event** as the text name.
   d. For the sequence, enter a number greater than the rest of the sequence numbers. This selection determines where the tab is displayed in the application.
   e. Leave the inactive field unchecked.
8. Create a screen menu item as follows:
   a. From the menu, select **Application** → **Siebel Sales Enterprise** → **Screen menu**.
   b. Right-click and select **New Record**.
   c. Enter **IBM2 Event** as the screen name and **IBM2 Event** as the text name.
   d. For the sequence, enter a number greater than the rest of the sequence numbers. This selection determines where the tab is displayed in the application.
   e. Leave the inactive field unchecked.
   f. Go to the screen menu item, **Locale** and create a new record for **IBM2Event**.

What to do next

Apply a schema to the event table

Applying a schema to the event table

Apply the physical schema for the new tables to your local database.

1. Query for the new table, **CX_IBM2_EVENT**.
2. Select the current query to create a physical schema. Leave the table space and index space blank.
3. To activate the new schema, click **Activate**.
4. Add or modify the Siebel VB or e-scripts for the business component (BC) that corresponds to the business objects used at your site. The Siebel script triggers the event notification for business objects. The samples are in the Sample folder. If you are planning to use multiple connectors, make sure the correct name is specified in the Siebel script. In the Siebel script, make sure the event is not created for the adapter user name, it will create a repeat effect.
5. Compile the updated and locked projects on your local database to create a new Siebel repository (.srf) file.

What to do next

Create a new account in the IBM2 event table.

Creating a new account in the IBM2 event table

Create a new account in the IBM2 event table to view the status of the events that are processed.

Before you begin

You must have administrative privileges to your local database.

How to perform this task

1. Open Siebel Sales Enterprise on your local database.
   a. Create a new view called IBM2 Event List View. **Tip**: Copy the view name from tools and paste it into the **View Name** field.
   b. Create a new responsibility called IBM2 Responsibility for the IBM2 Event List View.
   c. Add the employees or teams who are responsible for reviewing events to the newly created IBM2 Responsibility.
   d. Create the `IBMCONN` (or your adapter user name) user and add it to IBM2 Responsibility and Administrative Responsibility.
2. Test the application in your local environment.
3. Ensure that you have visibility to the *IBM2 Event List View* and that an event is generated in the view after you create a record in **Supported object**. For example, you must create a new account in Siebel and ensure that a new account event appears in the *IBM2 Event List View* .
4. Check in the new and updated projects to your development server.
5. Activate the new tables in the development database.
6. Compile a new `Siebel.srf` file in the server.
7. Enable Enterprise Application Integration by selecting **Sitemap > Server Administration > Component Group** and then selecting **Enable**.

Result

An event table is now created in the Siebel application for processing inbound events.

Creating the adapter project in WebSphere Integration Developer

To begin the process of creating and deploying a module, you create an adapter project. The adapter project contains the adapter itself plus other related artifacts. You create the project by importing the RAR file, which was copied to your local file system during installation, into WebSphere Integration Developer.

Before you begin

Make sure you have installed WebSphere Adapter for Siebel Business Applications and that you have created an authentication alias on the Siebel server.

About this task

Create an adapter project (called a *connector project* in WebSphere Integration Developer) to contain the adapter (which you import from the adapter installation directory) as well as artifacts related to it. All projects are self-contained; they do not refer to objects outside of the project.

To create an adapter project, use the following procedure.

How to perform this task

1. If WebSphere Integration Developer is not currently running, start it now.
   a. Click **Start** → **Programs** → **IBM WebSphere** → **Integration Developer 6.0**.
   b. If you are prompted to specify a workspace, accept the default value.

      The workspace is a directory where WebSphere Integration Developer stores your project.
   c. When the WebSphere Integration Developer window is displayed, close the Welcome page.
2. Switch to the J2EE perspective:
   a. Click **Window** → **Open Perspective** → **Other**.
   b. Click **J2EE**.

      If **J2EE** is not displayed in the Select Perspective window, select the **Show all** check box, click **J2EE**, and click **OK**.
   c. If you see the Confirm Enablement window, select **Always enable capabilities and don't ask me again**.
   d. Click **OK**.
3. Import the RAR file by right-clicking **Connector Projects** and clicking **Import** → **RAR file**.

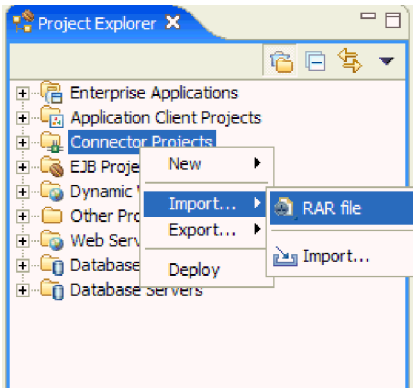


Figure 7. Importing the RAR file

4. From the Connector Import window, click **Browse** and navigate to the directory in which Adapter for Siebel Business Applications was installed.
5. Click **CWYEB_SiebelAdapter.rar**.

   The connector project has the same name as the RAR file.
6. **Optional:** In the **Connector project** field, type another name for the project or accept the default value.
7. **Optional:** In the **Target server** field, select the server to which the adapter will be deployed or accept the default value.
8. Clear the **Add module to an EAR project** check box.

   Notice that the EAR project field becomes unavailable after you remove the check mark.
9. Click **Finish**.

Result

A new J2EE Connector project is created. To see its contents, expand the project in Project Explorer. For example, if the connector project is named CWYEB_SiebelAdapter, expand **CWYEB_SiebelAdapter**.

What to do next

Add the required external dependencies to the project.

Adding external software dependencies

To add the required external dependency files to the adapter project, you first import the files to the connectorModule folder inside the adapter project, then you copy the files from the connectorModule folder of the adapter project to the project classpath.

Before you begin

Create the adapter project in WebSphere Integration Developer. Also, obtain the software dependency files.

About this task

The third party libraries that you add to the connector project are used when you deploy the project.

How to perform this task

The Siebel application requires that you add external software dependencies to the adapter project. These software dependencies enable the Adapter for Siebel Business Applications to communicate with the Siebel environment.

1. Import the Siebel Business Application software dependency files to the connectorModule folder inside the adapter project.
   a. In the J2EE perspective of WebSphere Integration Developer, expand the Connector Projects folder, then expand the adapter project.
   b. Right-click the **connectorModule** folder, then select **Import**.
   c. In the Import window, select **File system** from the list of import sources, then click **Next**.
   d. In the File system window, click **Browse**, then navigate to the directory that contains the Siebel Business Applications software dependencies and configuration files, then click **OK**.
   e. To select all of the files that are displayed in the right pan of the File system window, click **Select All**, then click **Finish**.

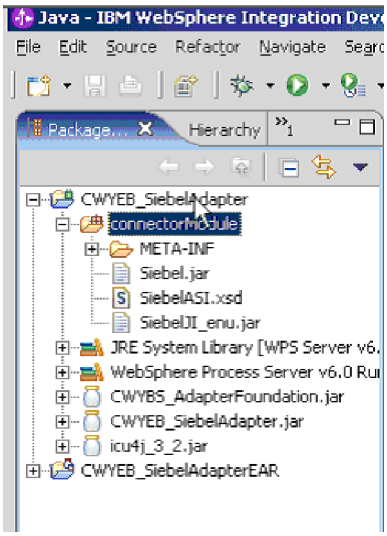


Figure 8. Viewing the dependency JAR files added to the connectorModule folder

2. Import the .jar files into the connectorModule folder.
   a. In the J2EE perspective of WebSphere Integration Developer, expand the Connector Projects folder, then expand the adapter project.
   b. Right-click the **connectorModule** folder, then select **Import**.
   c. In the Import window, select **File system** from the list of import sources, then click **Next**.
   d. In the File system window, click **Browse** to navigate to the following directory:*WebSphere_Integration_Developer_Installation_Directory*\runtimes\bi_v6\lib.

      Note: If you accepted the default installation directory when you installed WebSphere Integration Developer, the .jar files are located in the following directory: C:\Program Files\IBM\WebSphere\ID\6.0\runtimes\bi_v6\lib.

3. Add the software dependency files from the connectorModule folder to the adapter project classpath.
   a. Right-click the adapter project, then select **Properties**.
   b. In the Properties for CWYEB_Siebel Adapter window, select **Java Build Path** from the left pane.
   c. In the Libraries page of the right pane, click **Add JARs**.
   d. In the JAR Selection window, expand the adapter project folder (CWYEB_Siebel Adapter), then expand the connectorModule folder.
   e. Highlight all of the JAR files listed under the connectorModule folder, then click **OK**.

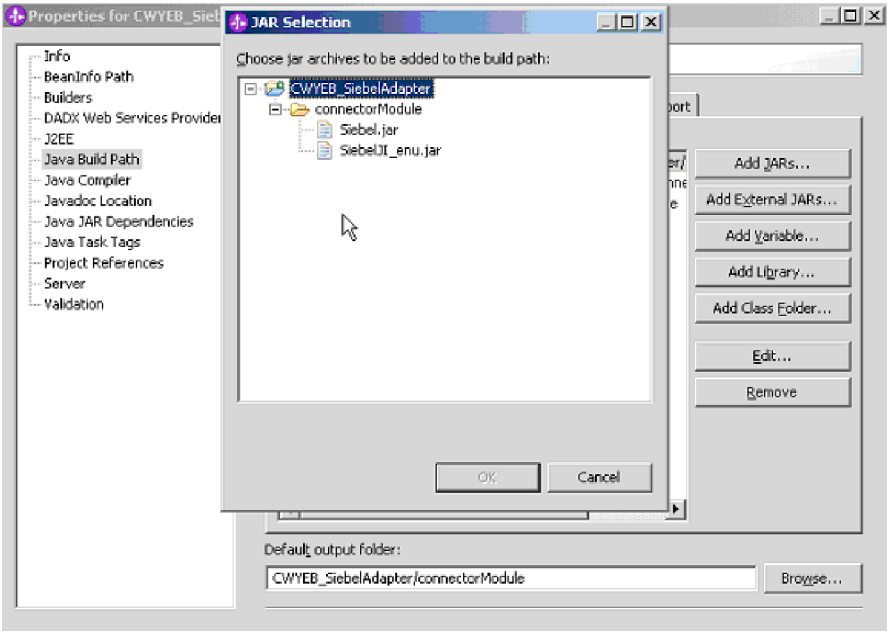


Figure 9. JAR files added to the connectorModule folder

   f. In the Properties window, click **OK**.

Result

The external dependencies are displayed in the adapter project folder.

What to do next

Configure the adapter.

Configuring the adapter for outbound processing

To configure WebSphere Adapter for Siebel Business Applications for outbound processing, use the enterprise service discovery wizard in WebSphere Integration Developer to set the connection properties for enterprise service discovery, select business objects or services that are in the enterprise information system, and generate business object definitions and related artifacts for outbound processing.

Generating artifacts using the enterprise service discovery wizard

Use the enterprise service discovery wizard to identify Siebel business objects and Siebel business services that can be used for outbound communication with the Siebel server.

Setting connection properties for enterprise service discovery

Use the enterprise service discovery wizard to set the outbound connection properties needed for the application and the adapter to communicate.

Before you begin

You must have created an adapter project with the external dependencies added to the adapter project.

About this task

To set the outbound connection properties, use the following procedure.

How to perform this task

1. From the Configure settings for Discovery Agent window, provide values for the following connection properties:
   - Connect String
   - Language Code
   - User Name
   - Password
2. From the **Siebel Metadata Type** pull-down menu, select either Siebel Business Services or Siebel Business objects.
3. In the **Siebel Repository** field, type a repository name. The default name is Siebel Repository.
4. Select the BiDi transformation check box, if bidirectional language text support is needed, and click **Next**.
5. **Optional:** To set the Log file and Prefix values, select **Advanced**.

Result

The outbound connection properties are set.

What to do next

Select the Siebel business objects or Siebel business services to use with the adapter.

Selecting business objects or services for use with the adapter

Browse the metadata information of the enterprise information system and select either the Siebel business objects or Siebel business services to use for outbound processing. The type of Siebel entities that are listed are based on the available selections for the Siebel metadata type.

Before you begin

The connection properties must be set for the enterprise service discovery wizard.

About this task

To select either the Siebel business objects or Siebel business services to use for inbound or outbound processing, use the following procedure.

How to perform this task

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

   The enterprise service discovery wizard fetches either the Siebel business objects or Siebel business services based on the value filled in against the filter property. The items shown depend on which Siebel metadata type was selected. For example, if you enter a value, such as *EAI*, all objects that start with EAI are retrieved by the enterprise service discovery wizard. To narrow the search parameters, you can enter the first few characters in the field and click **OK**. If there is no value provided and you run the query, all objects, for either business services or business objects are retrieved and listed in a tree structure, where each tree node represents an object.
3. From the Find and Discover Enterprise Services window, click **Run Query**.

   The query results are displayed in a tree structure.
4. From the search results discovered by the query, expand the node, and select the business service method or business component that you want, and click **Add**.
5. When you finish making your selections, click **Next**.

What to do next

Generate business object definitions and related artifacts for outbound processing. The type of Siebel entities listed are based on the available selections for the Siebel metadata type.

Generating artifacts

Use the enterprise service discovery wizard to generate artifacts for use with your adapter project. When you generate artifacts, you are adding instructions to the metadata that you extracted from the enterprise information file system. This process also bundles everything together to create an assembled adapter application, also known as an service component architecture (SCA) module.

Before you begin

WebSphere business objects or WebSphere business services must be added to the adapter project. In addition, an authentication alias must be created on the server.

About this task

To generate the artifacts for deployment to the server, use the following procedure.

How to perform this task

1. From the Configure Objects window, specify the properties for the objects that will be imported by the discovery agent:
   a. In the **Service Type** field, select **Outbound**.
   b. Accept the default in the **Namespace** field provided. The value for Namespace is initially set to the default for all business objects. Do not change the Namespace value.

c. Enter a **Business object location** and click **Next**. The business object location is where the generated WebSphere business objects are stored. This location is created as a folder under the top-level module folder.

2. Create a new business integration module:
   a. From the Generate Artifacts window, click **New**.
   b. From the Integration Project window, select **Create a module project**, and click **Next**.
   c. In the **Module Name** field, provide a module name, and click **Finish**.
3. In the **Folder** field, type a folder name. The corresponding folder is created under the module. This is the folder where the generated *.import* and *.wsdl* files are stored.
4. Select the **Use Discovered Connections** property radio button.
5. Provide the authentication alias name that you created in the WebSphere Process Server for the password property.
6. Provide values for the properties. Example property values are shown in the table below.

Table 15. Property examples

Property	Possible value
Adapter ID	ResourceAdapter
Log file size	500000
Log file name	C:\BOOutboundlog.log
Log files	1
Trace file size	500000
Trace file name	C:\BOOutboundTrace.trc
Trace files	1

7. Select the **Resonate Support** check box if your Siebel server administrator supports resonate support.
8. Type a value for **Siebel View Mode**, and click **Finish**. Using the default (3) allows you to see all the views.

Result

The result of running the enterprise service discovery wizard is an SCA module that contains an EIS import. Install this SCA module in the WebSphere Integration Developer integration test client.

What to do next

You can either configure the adapter for inbound event processing or you can deploy the module.

Configuring the adapter for inbound processing

To configure WebSphere Adapter for Siebel Business Applications for inbound processing, use the enterprise service discovery wizard in WebSphere Integration Developer to set the connection properties for the adapter, select business objects or services that are in the enterprise information system, and generate business object definitions and related artifacts for inbound processing.

Generating artifacts using the enterprise service discovery wizard

Use the enterprise service discovery wizard to identify Siebel business objects and Siebel business services that can be used for inbound communication with the Siebel server.

Setting connection properties for enterprise service discovery

Use the enterprise service discovery wizard to set the inbound connection properties needed for the application and the adapter to communicate.

Before you begin

You must have created an adapter project with the external dependencies added to the adapter project.

About this task

To set the outbound connection properties, use the following procedure.

How to perform this task

1. From the Configure settings for Discovery Agent window, provide values for the following connection properties:
   - Connect String
   - Language Code
   - User Name
   - Password
2. From the **Siebel Metadata Type** pull-down menu, select either Siebel Business Services or Siebel Business objects.
3. In the **Siebel Repository** field, type a repository name. The default name is Siebel Repository.
4. Select the BiDi transformation check box, if bidirectional language text support is needed, and click **Next**.
5. **Optional:** To set the Log file and Prefix values, select **Advanced**.

Result

The inbound connection properties are set for enterprise service discovery.

What to do next

Select the Siebel business objects or Siebel business services to use with the adapter.

Selecting business objects or services for use with the adapter

Browse the metadata information of the enterprise information system and select the Siebel business objects or Siebel business services to use for inbound processing. The type of Siebel entities that are listed are based on the available selections for the Siebel metadata type.

Before you begin

Connection properties must be set for the enterprise service discovery wizard.

About this task

To select the Siebel business objects or services to use with the adapter, use the following procedure.

How to perform this task

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

   The enterprise service discovery wizard fetches either the Siebel business objects or Siebel business services based on the value filled in against the filter property. The items shown depend on which Siebel metadata type was selected. For example, if you enter a value, such as *EAI*, all objects that start with EAI are retrieved by the enterprise service discovery wizard. To narrow the search parameters, you can enter the first few characters in the field and click **OK**. If there is no value provided and you run the query, all objects, for either business services or business objects are retrieved and listed in a tree structure, where each tree node represents an object.
3. From the Find and Discover Enterprise Services window, click **Run Query**.

   The query results are displayed in a tree structure.
4. From the search results, under the objects discovered by the query, expand the business service node, select the business service method that you want, and click **Add**.
5. For complex arguments, under the business service, select the appropriate integration object.
6. For the inbound service type, enter an Event Method. For inbound Siebel business service, the usual value is *QueryByExample*. You can remove an object by selecting the object from the lower pane of the Find and Discover Enterprise Services window and clicking **Remove**.
7. When you finish making your selections, click **Next**.

What to do next

Generate business object definitions and related artifacts needed for inbound event processing. The type of Siebel entities listed are based on the available selections for the Siebel metadata type.

Generating artifacts

Use the enterprise service discovery wizard to generate artifacts for use with your adapter project. When you generate artifacts, you are adding instructions to the metadata that you extracted from the enterprise information file system. This process also bundles everything together to create an assembled adapter application, also known as an service component architecture (SCA) module.

Before you begin

WebSphere business objects or WebSphere business services must be selected to add to the adapter project and an authentication alias must be created in the server.

About this task

To configure the artifacts that are deployed to the server, use the following procedure.

How to perform this task

1. From the Configure Objects window, specify the properties for the objects that will be imported by the discovery agent:
   a. In the **Service Type** field, select **Inbound**.
   b. Accept the default in the **Namespace** field provided. The value for Namespace is initially set to the default for all business objects. Do not change the Namespace value.
   c. Enter a **Business object location** and click **Next**. The business object location is where the generated WebSphere business objects are stored. This location is created as a folder under the top-level module folder.
2. Create a new business integration module:
   a. From the Generate Artifacts window, click **New**.
   b. From the Integration Project window, select **Create a module project**, and click **Next**.
   c. In the **Module Name** field, provide a module name, and click **Finish**.
3. In the **Folder** field, type a folder name. The corresponding folder is created under the module. This is the folder where the generated *.import* and *.wsdl* files are stored.
4. Select the **Use Discovered Connections** property radio button.
5. Provide the authentication alias name that you created in the WebSphere Process Server for the password property.
6. Provide the values for the properties. Example values are shown in the table below.

Table 16. Property examples

Property	Possible value
Event component name	IBM2
Adapter ID	ResourceAdapter
Log file size	500000
Log file name	C:\BSInboundlog.log
Log files	1
Trace file size	500000
Trace file name	C:\BSInboundTrace.trc
Trace files	1

7. Select the **Resonate Support** check box if your Siebel server administrator supports resonate support.
8. Type a value for **Siebel View Mode**, and click **Finish**. Using the default (3) allows you to see all the views.

Result

The result of running the enterprise service discovery wizard is an SCA module that contains an EIS import. Install this SCA module in the WebSphere Integration Developer integration test client.

What to do next

Generate reference bindings for the test environment or deploy the adapter project.

Generating reference bindings (test environment only)

Generate reference bindings to create a reference in the assembly editor from the adapter project to a stand-alone reference for inbound event processing. The stand-alone reference represents an a generic J2EE component, such as the application server. By wiring the adapter project to the stand-alone reference, you link the adapter to other server processes.

Before you begin

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

About this task

To generate reference bindings to bind to the service component, use the following procedure.

How to perform this task

1. From the WebSphere Integration Developer main window, beneath **All Resources**, select the module name for your project.
2. Double-click on your project folder. The project module appears in the upper-right pane.
3. Select the **Import** icon. Hold your cursor over the icons on the left side of the pane until you find the icon that you want. When you select the **Import** icon, several more icons appear.
4. Double-click the **Standalone References** icon. The Standalone Reference box appears in the right pane.
5. Drag and drop the yellow bulb around the Standalone Reference box onto the Import Module to bind them.
6. From the Add Wire dialog box, click **OK**.
7. Right-click the **Standalone References** icon and select **Generate JAVA implementation**.
8. Save the inbound service type file.

What to do next

Deploy the module.

Chapter 8. Deploying the module

To deploy the module to the application server, export the adapter project as an enterprise archive (EAR) file, install the module, and add any configuration properties that were not set in the enterprise service discovery wizard.

Exporting the project as an EAR file

Using the enterprise service discovery wizard, export the adapter project that you have created as an EAR file. By creating an EAR file, you capture all of the contents of your adapter project in a format can be easily deployed to the application server.

Before you begin

Before you can export the project as an EAR file, you must have created your business objects, and have a configured project that is free of build errors.

About this task

To export the module as a resource adapter archive (RAR) file, perform the following procedure.

How to perform this task

1. Confirm that there are no errors by building the project.
2. Right-click on the project and select **Export → EAR file**.
3. In the EAR Export window, select the EAR project.
4. Provide the absolute path including the EAR file name for the Destination. Example of absolute paths, including the EAR file name, are C:\SiebelBuild\Siebel_BS_OutboundApp.ear and ear and C:\SiebelBuild\Siebel_BS_InboundApp.
5. Select the following options:
   - Export source files
   - Overwrite existing file
   - Include project build paths and metadata files
6. Click **Finish**.

Result

The adapter project is exported to an EAR file.

What to do next

Install the module in the server administrative console. This deploys the module to application server.

Installing the module

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

Before you begin

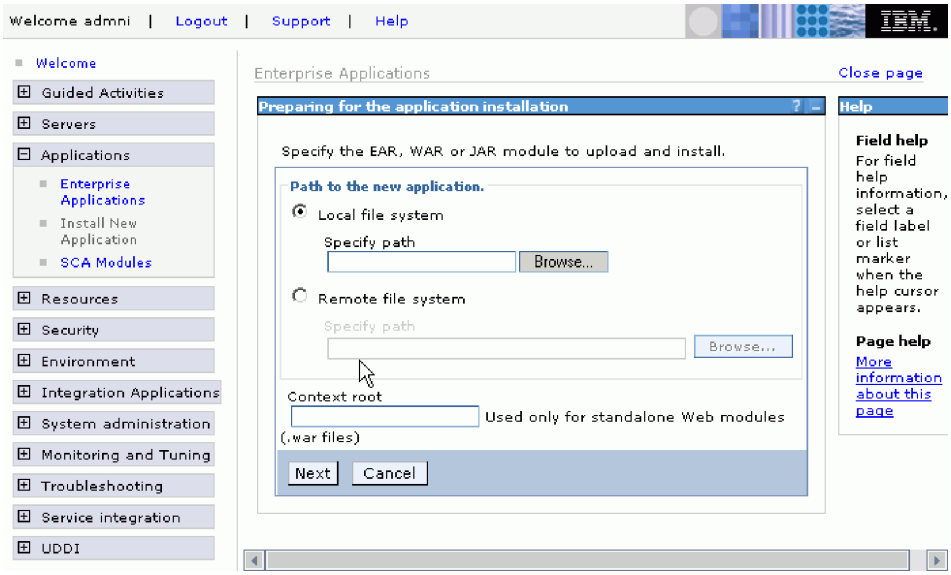
You must have exported your project module as an EAR file before installing the adapter project.

About this task

To install the adapter module, perform the following procedure. For more information on clustering adapter project applications, see http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp.

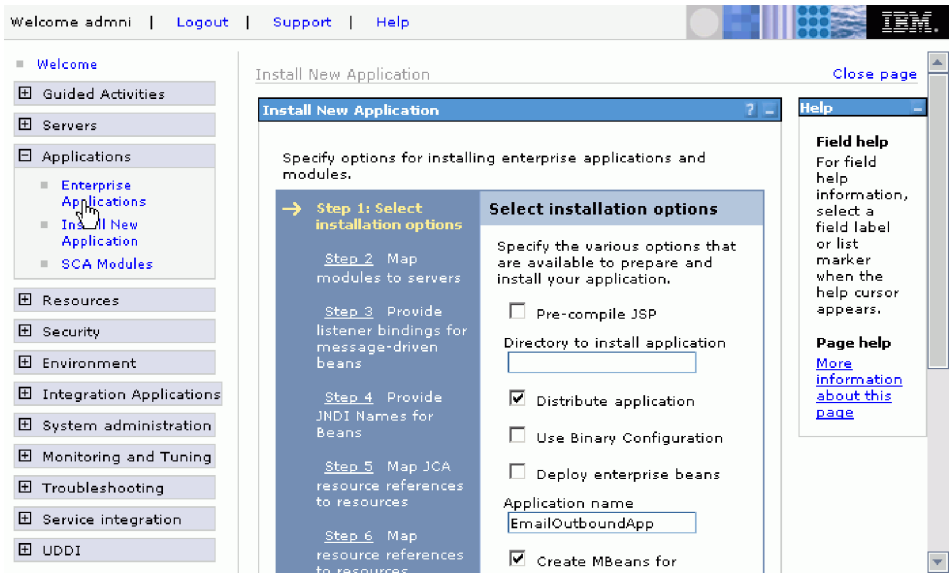
How to perform this task

1. Open the WebSphere Process Server administrative console by right-clicking your server instance and selecting **Run administrative console**.
2. In the administrative console window, click **Applications** → **Install New Applications**.



Preparing for the application installation window

3. Click **Browse** to locate your EAR file and click **Next**.
4. **Optional:** If you are deploying to a clustered environment, click **Next** until you reach Step 2: Mapping modules to servers, then select **Modules** and then the name of the server cluster and click **Apply**. Note: Adapter instances are replicated in a clustered server environment when enableHASupport is set to true. Do not change the value of enableHASupport for single server environments. **Note:** Adapter instances are replicated in a clustered server environment when **enableHASupport** is set to true. Do not change the value of **enableHASupport** for single server environments.
5. Click **Next** until you reach Step 6: Map resource reference to resources.



Install New Application window

6. Select **SCA Auth Alias** from the select authentication data entry list.
7. Select the check box for the module and click **Apply**.
8. Click **Next**. A summary of all of the installation options is displayed.
9. Verify that all options are correct and click **Finish**.
10. Confirm that the application was installed successfully.
11. Click the **Save to Master Configuration** link at the end of the list of installation messages.
12. Click **Save**.

Result

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

What to do next

If you want to set or reset resource adapter, managed connection factory, activation specification, or data transformation properties, or you would like to cluster adapter project applications, you should do that using the WebSphere Process Server administrative console before configuring troubleshooting tools.

Setting or changing adapter configuration properties

The adapter configuration properties are set using the enterprise service discovery wizard when you generate artifacts. However, after you deploy the adapter, you can use the WebSphere Process Server administrative console to reconfigure the adapter properties.

Setting resource adapter properties

Resource adapter properties include logging and tracing properties, bidirectional properties, and adapter-specific properties. Using the enterprise service discovery wizard when you first configure the adapter (and later, through the WebSphere Process Server administrative console), you can configure Resource Adapter properties.

Before you begin

Verify that WebSphere Process Server is properly working and currently running.

About this task

Resource adapter properties are set using the enterprise service discovery wizard when you create your business objects or services. You can reconfigure the resource adapter properties using the WebSphere Process Server administrative console.

How to perform this task

1. Start the WebSphere Process Server administrative console.
2. Under **Resources**, select **Resource Adapters**.
3. Under **Resource Adapters**, select **WebSphere Adapter for Siebel Business Applications**.
4. From the General Properties page, under Additional Properties, select **Resource Adapter Properties**.
5. Select the resource adapter property that you would like to configure from the resource adapter properties and set its value.

Setting managed (J2C) connection factory properties

Managed (J2C) connection factory properties affect outbound processing and correspond to the ManagedConnectionFactory interface of the J2EE Connector Architecture Specification. A J2C connection factory also manages connection pooling. It provides configuration information for outbound operations via the resource adapter. Managed (J2C) connection factory properties are set using the enterprise service discovery wizard when you create your business objects. You can reconfigure the Managed (J2C) connection factory properties using the WebSphere Process Server administrative console.

Before you begin

Verify WebSphere Process Server is properly working and currently running.

About this task

Managed (J2C) connection factory properties are set using the enterprise service discovery wizard when you create your business objects. You can reconfigure the managed (J2C) connection factory properties using the WebSphere Process Server administrative console.

How to perform this task

1. Start the WebSphere Process Server administrative console.
2. Under **Resources**, select **Resource Adapters**.
3. Under **Resource Adapters**, select **WebSphere Adapter for Siebel Business Applications**.
4. From the General Properties page, under Additional Properties, select managed (J2C) connection factory properties.
5. Select the managed (J2C) connection factory properties that you would like to configure from the Resource adapter properties and set their value.

Setting activation specification properties for the EIS

Activation specification properties hold the inbound event processing configuration information for a message endpoint. You can reconfigure the activation specification properties using the WebSphere Process Server administrative console.

Before you begin

Verify WebSphere Process Server is properly working and currently running.

About this task

ActivationSpecification properties are set using the enterprise service discovery wizard when you create your business objects. You can reconfigure the managed ActivationSpecification properties using the WebSphere Process Server administrative console.

How to perform this task

1. Start the WebSphere Process Server administrative console.
2. Under **Resources**, select **Resource Adapters**.
3. Under **Resource Adapters**, select **WebSphere Adapter for Siebel Business Applications**.
4. From the General Properties page, under Additional Properties, select **ActivationSpecification** properties.
5. Select the managed ActivationSpecification properties that you would like to configure from the resource adapter properties and set their values.

Chapter 9. Configuring troubleshooting tools

Configure the troubleshooting tools to suit your requirements. Enable logging for the adapter to control the status of event processing. Enable the Common Event Infrastructure to collect diagnostic information about your adapter. Set tracing levels to determine the level of the information captured in the adapter log and trace files. Install IBM Support Assistant to gain quick access to support-related information along with serviceability tools for problem determination for IBM software products.

Enabling tracing with the Common Event Infrastructure (CEI)

Enable tracing and control the level of detail in the adapter trace by configuring the Common Event Infrastructure (CEI).

Before you begin

Before you enable tracing with CEI, complete the following tasks:

- Enable the diagnostic trace service.
- Publish the IBM WebSphere Adapters event definitions file to the CEI catalog before you can set these event definitions.

For instruction on how to do these tasks, refer to the CEI documentation located on the Web site for your server:

- For WebSphere Process Server: http://www.ibm.com/software/integration/wps
- For WebSphere Enterprise Service Bus: http://www.ibm.com/software/integration/wsesb

To enable tracing and control the level of trace detail, use the following procedure.

How to perform this task

1. In the administrative console, click **Troubleshooting**.
2. Click **Logs and Trace**.
3. In the list of servers, click the name of your server.
4. In the General Properties area, click **Change Log Detail Level** and then select **com.ibm.j2ca.*** for the adapter components. There is a subcomponent for each adapter type, as described in the following table.

Adapter	Package Name
WebSphere Adapter for Email	com.ibm.j2ca.email.*
WebSphere Adapter for Flat Files	com.ibm.j2ca.flatfile.*
WebSphere Adapter for FTP	com.ibm.j2ca.ftp.*
WebSphere Adapter for JDBC	com.ibm.j2ca.jdbc.*
WebSphere Adapter for JD Edwards EnterpriseOne	com.ibm.j2ca.jde.*
WebSphere Adapter for SAP Software	com.ibm.j2ca.sap.*
WebSphere Adapter for Siebel Business Applications	com.ibm.j2ca.siebel.*

5. Select the component that matches your adapter. Each adapter component has two subcomponents, one for logging and one for CEI. They are:

- *subcomponent_name*.log.*adapter_ID*
- *subcomponent_name*.cei.*adapter_ID*

For example, com.ibm.j2ca.siebel.cei.*adapter_ID1*. For each instance of a deployed adapter, the system shows a separate ID.

6. Select the CEI adapter ID that you want to enable.
7. From the list, choose the level of business object detail to capture in service component events:
   - **off**. Turn CEI off.
   - **fine**. Turn CEI on but publish none of the business object payload. This corresponds to the event control detail level of Empty in WebSphere Integration Developer.
   - **finer**. Turn CEI on and publish only the payload description for the business object. This corresponds to the event control detail level of Digest in WebSphere Integration Developer .
   - **finest**. Turn CEI on and publish all of the business object payload. This corresponds to the event control detail level of Full in WebSphere Integration Developer.
   - **all**. Same as **finest**.

   For information on what each event content level means (Empty, Digest and Full), and for more information on using the Common Base Event model and the Common Event Infrastructure, refer to the documentation for your process server.

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 process 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 a process server, including service components and event points, see the documentation for your process server.

You can change the log configuration statically or dynamically. Static configuration take 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.

How to perform this task

1. In the navigation pane of the administrative console, click **Servers** → **Application Servers**.
2. Click the name of the server that you want to work with.
3. Under **Troubleshooting**, click **Logs and trace**.
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. Select 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.base.***.
   - For a specific adapter, select its package name.

Adapter	Package Name
WebSphere Adapter for Email	com.ibm.j2ca.email
WebSphere Adapter for Flat Files	com.ibm.j2ca.flatfile
WebSphere Adapter for FTP	com.ibm.j2ca.ftp
WebSphere Adapter for JDBC	com.ibm.j2ca.jdbc
WebSphere Adapter for JD Edwards EnterpriseOne	com.ibm.j2ca.jde
WebSphere Adapter for SAP Software	com.ibm.j2ca.sap
WebSphere Adapter for Siebel Business Applications	com.ibm.j2ca.siebel

7. Click the package name and 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.

Logging Level	Description
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 process server.

Changing the log and trace file names

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

About this task

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

You can change the log configuration statically or dynamically. Static configuration changes affect applications when you start or restart the application server. Dynamic or run time configuration 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.

How to perform this task

1. In the navigation pane, click **Enterprise Applications**.
2. Click the name of the adapter application. This is the name of the EAR file for the adapter, without the .ear file extension. For example, if the EAR file is named Accounting_OutboundApp.ear, then click **Accounting_OutboundApp**.
3. Click **Connector Modules**.
4. Select the adapter by clicking the name of the RAR file for the adapter. The RAR files are listed in the following table.

Adapter	RAR File Name
WebSphere Adapter for Email	CWYEM_Email.rar
WebSphere Adapter for Flat Files	WYFF_FlatFile.rar
WebSphere Adapter for FTP	CWYFT_FTPFile.rar
WebSphere Adapter for JDBC	CWYBC_JDBC.rar
WebSphere Adapter for JD Edwards EnterpriseOne	CWYED_JDE.rar
WebSphere Adapter for SAP Applications	CWYAP_SAPAdapter.rar CWYAP_SAPAdapterTX.rar

Adapter	RAR File Name
WebSphere Adapter for Siebel Business Applications	CWYEM_Siebel.rar

5. Click the name of the resource adapter.
6. In the Custom Properties area, specify the file names:
   - To change the log file name, type the name in the **Value** field for **logFilename**. By default, this log is in the SystemOut.log file.
   - To change the trace file name, type the name in the **Value** field for **traceFilename**. By default, this log is in the trace.log file.
7. To have static configuration changes take effect, stop and then restart the process server.

Installing or upgrading IBM Support Assistant

IBM Support Assistant (ISA) is a free, local software serviceability workbench that helps you resolve questions and problems with IBM software products. Install plug-ins for the products you have installed. It provides quick access to support-related information along with serviceability tools for problem determination. Installing and upgrading it is simple and straightforward.

About this task

IBM Support Assistant provides the following services:

- Symptom-based data collection
- Access to IBM support information, IBM newsgroups, and other resources through a federated search interface (one search, multiple resources)
- Easy access to IBM educational materials
- Easy access to IBM product home pages, product support pages, and product forums or newsgroups through convenient links
- A tools framework and update manager to easily update and install ISA plug-ins and tools
- Fast resolution of problem management records through electronic submission of critical system data to IBM

You can install and run both version 2 and version 3 of IBM Support Assistant on a single computer, to get support for a broad range of IBM solutions.

To install and upgrade IBM Support Assistant, use the following procedure.

How to perform this task

1. Go to the IBM Support Assistant Web page at:
   http://www.ibm.com/software/support/isa/
2. Follow the directions on the Web page to download ISA version 3.0, and then to extract, install, and use the tool.
3. Start ISA.
4. Open the **Updater** component.
5. On the **Upgrades** tab, upgrade ISA to version 3.0.1 or higher.
6. On the **New Products and Tools** tab, install the plug-ins for your adapter. Select the plug-in for your adapter from the list for the WebSphere brand. There is an optional language pack plug-in for each adapter, which enables you to see adapter-specific information in languages other than English.

Chapter 10. Administering the adapter

Use the administrative console of the server to start, stop, and troubleshoot the adapter.

Starting the adapter

To start an adapter that has a status of Stopped, use the administrative console. By default, an adapter starts automatically when the server starts.

Before you begin

The administrative console of the server must be running in order to complete this task.

To start the adapter, use the following procedure.

How to perform this task

1. On the Enterprise Applications page, click **Applications** → **Enterprise Applications**.
2. Select the check box of the adapter that you want to start.
3. Click **Start**.

Result

The status of the adapter changes to Started and a message stating that the adapter started displays at the top the page.

Use the administrative console of the server to stop the adapter.

Stopping the adapter

Use the administrative console of the server to stop an adapter.

Before you begin

The administrative console of the server must be running in order to complete this task.

To stop the adapter, use the following procedure.

How to perform this task

1. On the Enterprise Applications page, click **Applications** → **Enterprise Applications**.
2. Clear the check box of the adapter you want to stop.
3. Click **Stop**.

Result

The status of the adapter changes to Stopped and a message stating that the adapter stopped displays at the top the page.

Use the administrative console of the server to troubleshoot the adapter.

Troubleshooting and support

Common troubleshooting techniques and self-help information help you identify and solve problems quickly. If necessary, follow the procedures for contacting IBM Software Support.

Exception: XAResourceNotAvailableException

When the process 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 process server log file:

com.ibm.ws.Transaction.XAResourceNotAvailableException

Problem:

A resource was removed while the process 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 process 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 WebSphere Integration Developer, 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*\wps*server_name*\transaction\tranlog
3. Start the process server.

Trouble connecting to Siebel

If you have trouble connecting to Siebel, check the *ConnecString* property, ensure that object manager is running, and check the dependency files.

Before you begin

The administrative console of the server must be running in order to complete this task.

About this task

To stop the adapter, use the following procedure.

How to perform this task

1. Check if the port mentioned in the **ConnectString** property is correct. For Siebel versions later than 7.7, the default port number is 2321. The ConnectString property should include the port number. For more information, refer to the properties section for the ConnectString format.
2. Check if the object manager being used is running.
3. Check if the right dependency files are being used.

The

Self help resources

Use the self help resources of IBM Software Support to get the most current support information, to obtain technical documentation, to download support tools and fixes, and prevent problems with WebSphere Adapter for Siebel Business Applications. The self help resources also help you diagnose problems with the adapter and contact IBM Software Support.

The software support Web site for WebSphere Adapters at http://www.ibm.com/software/integration/wbiadapters/supp provides the following resources:

- Flashes (alerts from technical support)
- Technotes

  You can get a list of technotes for WebSphere Adapters at http://www.ibm.com/support/search.wss?rs=695&tc=SSMKUK
- Authorized program analysis reports (APARs)
- Technical information including the product information center, manuals, IBM Redbooks™, and whitepapers.
- Educational offerings
- *IBM Software Support Handbook*

Register at the site to use My Support to create a customized support page for your use.

Contacting IBM Software Support

IBM Software Support provides support for WebSphere Adapters either online or by phone. Gathering information about the problem before you contact IBM Software Support can dramatically increase support responsiveness.

Before you begin

If you think your problem is defect-related, IBM Software Support provides assistance. Before contacting IBM Software Support, your company must have an active IBM software maintenance contract, and you must be authorized to submit problems to IBM. The type of software maintenance contract that you need depends on the type of product you have:

- For IBM distributed software products (including, but not limited to, Tivoli®, Lotus®, and Rational® products, as well as DB2® and WebSphere products that run on Windows, Linux®, or UNIX® operating systems), you must be enrolled in Passport Advantage®. You can enroll in one of the following ways:

  Online

  Go to the Passport Advantage Web page (http://www-306.ibm.com/software/support/pa.html), and click **How to Enroll**.

  By phone

  For the phone number to call in your country, go to the contacts page of the IBM Software Support Handbook on the Web (http://techsupport.services.ibm.com/guides/contacts.html), and click the name of your geographic region.
- For IBM eServer™ software products (including, but not limited to, DB2 and WebSphere products that run in zSeries®, pSeries®, and iSeries™ environments), you can purchase a software maintenance agreement by working directly with an IBM sales representative or an IBM Business Partner. For more information about support for eServer software products, go to the IBM Technical Support Advantage Web page (http://www-03.ibm.com/servers/eserver/techsupport.html).

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

About this task

The IBM Software Support Handbook contains detailed information about the service and support of your IBM products. Read the handbook at http://techsupport.services.ibm.com/guides/handbook.html.

To contact IBM Software Support, use the following procedure.

How to perform this task

1. Describe your problem and gather background information. When explaining a problem to a support specialist, be as specific as possible. Include all relevant background information so that the specialists can help you solve the problem efficiently. To save time, know the answers to these questions:
   - What software versions were you running when the problem occurred? Include the version of the operating system as well as related products.
   - Has the problem happened before, or is this an isolated problem?
   - What steps led to the failure?
   - Can the problem be recreated? If so, what steps led to the failure?
   - Have any changes been made to the system such as to the hardware, operating system, networking software, and so on?
   - Are you currently using a workaround for this problem? If so, be prepared to explain it when you report the problem.
   - Do you have logs, traces, and messages that are related to the problem symptoms? IBM Software Support is likely to ask for this information.

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

Table 17. Severity criteria for problem reporting

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

3. Submit your problem to IBM Software Support. You can submit your problem in the following ways:
   - **Online**. Go to the Submit and track problems page on the IBM Software Support site http://www.ibm.com/software/support/probsub.html Enter your information into the appropriate problem submission tool.
   - **By phone**. For the phone number to call in your country, go to the contacts page of the IBM Software Support Handbook on the Web (http://techsupport.services.ibm.com/guides/contacts.html), and click the name of your geographic region.

Result

If the problem you submit is for an unreported software defect or for missing or inaccurate documentation, IBM Software Support creates an Authorized Program Analysis Report (APAR). The APAR describes the problem in detail and tracks its resolution.

What to do next

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

Chapter 11. Quick start tutorials

To gain practical knowledge in setting up and deploying the adapter, complete one or more of the quick start tutorials. Everything you need to complete each tutorial is contained in the tutorial. If you have performed the prerequisite tasks (such as installing the adapter), you can complete each tutorial in under an hour.

Introduction

Each scenario provides a complete set of instructions for configuring the adapter so that it can be used by a J2EE component to send or receive requests to the enterprise information file system. Use the enterprise service discovery wizard in WebSphere Integration Developer to configure the adapter, connect to the enterprise information file system, and retrieve information about a Siebel business object or service. Enterprise service discovery creates the Siebel business object or service needed to create a deployable module.

In the tutorials, you use the enterprise service discovery wizard to configure the adapter, connect to the Siebel server, and retrieve information about a Siebel business object or service on the Siebel server. The enterprise service discovery wizard then creates the business objects and interface information needed to interact with the Siebel enterprise information system (EIS) using the Siebel business object or service. The business objects and interface information, along with the adapter, are built into a deployable module.

The adapter supports the exchange of data with the Siebel EIS through two types of Siebel objects: Siebel business services and Siebel business objects. The scenario examples show outbound and inbound operations using Siebel business services and business components.

There are a total of four scenarios:

- Outbound processing for Siebel business services
- Outbound processing for Siebel business objects
- Inbound processing for Siebel business services
- Inbound processing for Siebel business objects

Outbound processing for Siebel business services

The Insert and QueryByExample operations are demonstrated in this scenario. The Insert operation is used to add an instance hierarchy of the integration object, Account Interface, to the Siebel enterprise information system (EIS), using the Siebel business service, Siebel Account. The QueryByExample operation retrieves the created Account Interface instance from the Siebel EIS.

The outbound Siebel business services relationships demonstrated in this scenario include the following:

- For business service, Siebel Account is used
- For the integration object, Account Interface is used
- For operations, Insert and QueryByExample are used

Outbound processing for Siebel business objects

The Create and Exists operations are demonstrated in this scenario. The Create operation is used to add an instance hierarchy of the business object, Account – ESP, to the Siebel EIS. The Exists operation confirms that a particular Account instance exists in the Siebel EIS.

The outbound Siebel business object relationships demonstrated in this scenario include the following:

- For business object, Account – ESP is used
- For operations, Create and Exists are used

Inbound processing for Siebel business services

This scenario includes updating an instance hierarchy of Account Interface (not through the adapter) in the EIS. There is an event added automatically (through the Siebel script) to the event component for the corresponding update. The adapter polls the event from the event component and retrieves the updated instance hierarchy using the QueryByExample operation. This is done based on the information available in the event.

The inbound Siebel business service relationships demonstrated in this scenario include the following:

- For integration object, Account Interface is used
- For the operation, QueryByExample is used

Inbound processing for Siebel business objects

This scenario includes retrieving an updated instance hierarchy of Account - ESP. There is an event added automatically (through the Siebel script triggers) to the event component when the corresponding update happens. The adapter then polls the event from the event component and retrieves the updated instance hierarchy based on the information available in the event.

The inbound Siebel business object relationships demonstrated in this scenario include the following:

- For business object, Account – ESP is used
- For the operation, Retrieve is used

Learning objectives

After completing the tutorial, you should be able to perform the following tasks:

- Create an adapter project in WebSphere Integration Developer
- Discover services and associated business objects from the enterprise information file system and make them part of the adapter project
- Create a deployable module that you install on WebSphere Process Server or WebSphere Enterprise Service Bus
- Test the module to ensure that it operates correctly and to see the results of running the module

Time required

The time required to complete each tutorial is about one hour.

Audience

These tutorials are for integration developers who design, assemble, test, and deploy business integration solutions.

Prerequisites

To complete this tutorial, the following applications must be either installed or accessible:

- WebSphere Integration Developer, version 6.0.1 or later must be installed
- WebSphere Process Server or WebSphere Enterprise Service Bus must be installed
- WebSphere Adapter for Siebel Business Applications must be installed

Accessing the tutorial files

Run the installer to install the RAR file, which contains the adapter and the samples folder. These files are exact examples of the artifacts that you create when using the enterprise service discovery wizard in this scenario. These files available for reference so that you can verify the look of the files that you create.

1. Run the installer to install the RAR file, which contains the adapter and the sample files.
2. In the folder under **adapter/Siebel/samples/referencefiles**, extract the contents of the ZIP files.

Business objects used in the tutorial

The tutorials use Siebel business services and Siebel business objects for inbound and outbound processing.

Tutorial pre-configuration prerequisites

Before you begin working on the tutorials, set the password for the authentication alias, create an event table in the Siebel application, create an adapter project, and add external software dependencies.

Creating the authentication alias

Create the authentication alias on the server by using the WebSphere Process Server administrative console. From the administrative console, configure the global security and set the password for the authentication alias, which is used to process inbound and outbound events.

Before you begin

If you have not done so already, create an authentication alias on the server for both inbound and outbound processing.

About this task

To create an authentication alias, use the following procedure.

How to perform this task

1. On the WebSphere administrative console ″Welcome page,″ click **Security** → **Global security**.

2. Under the Authentication heading, click **JAAS Configuration → J2C Authentication data**.
3. Click **New**.
4. Type the required information in the **Alias**, **User ID**, **Password**, and **Description** fields.

   Note: The user ID and password that you type are used to establish a connection to the enterprise information system for outbound processing.
5. Click **OK**, click **Save**, and then click **Save** again.

Creating an event component in the Siebel application

Create an event component, also known as an event table, in the Siebel application to process inbound operations.

About this task

If the deployed enterprise application is configured for inbound processing, it cannot be started unless the event component exits in the Siebel application. The procedure shown for the event component uses the Siebel Sales Enterprise application as an example. Substitute all references to Siebel Sales Enterprise with the name of the Siebel application in use. For this event table example, the name, IBM2 Events, has been used. Based on your needs, this event component name can be changed.

Creating event table columns

Create the event table columns in the Siebel event table by using the event wizard in the Siebel application.

Before you begin

You must be familiar with the tools found in the Siebel application to create the event table.

About this task

An event table must be created in the Siebel application event component to track the events that occur in the Siebel enterprise information system.

How to perform this task

1. Create an IBM project and lock your project, according to the Siebel instructions. Except for event triggers, you must complete all Siebel customization under the new project.
2. If you are installing multiple connectors, create multiple tables with different names.
3. Using the new object wizard, create a standalone table named `CX_IBM2_Events`.
4. Create columns in your new table with the following names:
   - Column Name/User Name
   - Type
   - Length
   - Physical type
   - Required
   - Nullable

- Status

The example below shows some typical column titles and along with their associated information.

Table 18. Event table column example

Column name/ user name	Type	Length	Physical type	Required	Nullable	Status
DESCRIPTION	Data (public)	255	Varchar	No	Yes	Active
EVENT_ ID	Data (public)	30	Varchar	Yes	N/A	Active
EVENT_ TIMESTAMP	Data (public)	7	Date Time	Yes	N/A	Active
EVENT_ TYPE	Data (public)	20	Varchar	Yes	N/A	Active
OBJECT_ KEY	Data (public)	255	Varchar	Yes	N/A	Active
OBJECT_ NAME	Data (public)	255	Varchar	Yes	N/A	Active
PRIORITY	Data (public)	10	Number	No	Yes	Active
STATUS	Data (public)	20	Number	Yes	N/A	Active
XID	Data (public)	255	Varchar	No	N/A	Active

What to do next

Create a business component in the event table.

Creating a business component

Create a business component called IBM2 Event in the event table.

1. Create a new Business Component (BC), **IBM2 Events**, based on your new table. All fields are single value fields.
2. Create a new business object (BO) named **IBM2 Events**.
3. Associate **IBM2 Event BC** to the **IBM2 Event BO**.
4. Create **IBM2 Event List View Applet** based on **IBM2 Event BC**.
5. Create **IBM2 Event List View** based on **IBM2 Event BO**.
6. Create **IBM2 Event Screen** and associate it to the **IBM2 Event List View** in screen view.
7. Create a page tab as follows:
   a. Access the **Application** → **Siebel Sales** → **Page tab**.
   b. Right-click and select **New Record** from the menu.
   c. Enter **IBM2 Event** as the screen name and **IBM2 Event** as the text name.
   d. For the sequence, enter a number greater than the rest of the sequence numbers. This selection determines where the tab is displayed in the application.

e. Leave the inactive field unchecked.

8. Create a screen menu item as follows:
   a. From the menu, select **Application → Siebel Sales Enterprise → Screen menu**.
   b. Right-click and select **New Record**.
   c. Enter **IBM2 Event** as the screen name and **IBM2 Event** as the text name.
   d. For the sequence, enter a number greater than the rest of the sequence numbers. This selection determines where the tab is displayed in the application.
   e. Leave the inactive field unchecked.
   f. Go to the screen menu item, **Locale** and create a new record for **IBM2Event**.

What to do next

Apply a schema to the event table

Applying a schema to the event table

Apply the physical schema for the new tables to your local database.

1. Query for the new table, **CX_IBM2_EVENT**.
2. Select the current query to create a physical schema. Leave the table space and index space blank.
3. To activate the new schema, click **Activate**.
4. Add or modify the Siebel VB or e-scripts for the business component (BC) that corresponds to the business objects used at your site. The Siebel script triggers the event notification for business objects. The samples are in the Sample folder. If you are planning to use multiple connectors, make sure the correct name is specified in the Siebel script. In the Siebel script, make sure the event is not created for the adapter user name, it will create a repeat effect.
5. Compile the updated and locked projects on your local database to create a new Siebel repository (.srf) file.

What to do next

Create a new account in the IBM2 event table.

Creating a new account in the IBM2 event table

Create a new account in the IBM2 event table to view the status of the events that are processed.

Before you begin

You must have administrative privileges to your local database.

How to perform this task

1. Open Siebel Sales Enterprise on your local database.
   a. Create a new view called IBM2 Event List View. **Tip**: Copy the view name from tools and paste it into the **View Name** field.
   b. Create a new responsibility called IBM2 Responsibility for the IBM2 Event List View.
   c. Add the employees or teams who are responsible for reviewing events to the newly created IBM2 Responsibility.

d. Create the IBMCONN (or your adapter user name) user and add it to IBM2 Responsibility and Administrative Responsibility.

2. Test the application in your local environment.
3. Ensure that you have visibility to the *IBM2 Event List View* and that an event is generated in the view after you create a record in **Supported object**. For example, you must create a new account in Siebel and ensure that a new account event appears in the *IBM2 Event List View* .
4. Check in the new and updated projects to your development server.
5. Activate the new tables in the development database.
6. Compile a new Siebel.srf file in the server.
7. Enable Enterprise Application Integration by selecting **Sitemap > Server Administration > Component Group** and then selecting **Enable**.

Result

An event table is now created in the Siebel application for processing inbound events.

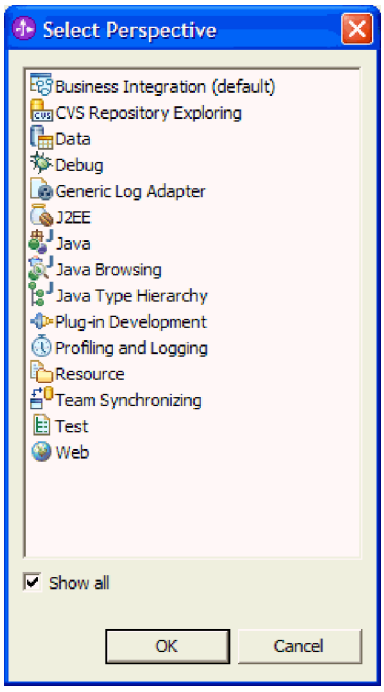
Creating the adapter project in WebSphere Integration Developer

All the tutorials require that you create an adapter project. The same project is used for all the tutorials. The adapter project contains the adapter itself plus other related artifacts. Create the project by importing the RAR file, which was copied to your local file system during installation, into WebSphere Integration Developer.

1. In WebSphere Integration Developer, switch to the J2EE perspective:
   a. Click **Window → Open Perspective → Other**.
   b. Click **J2EE**.

      If **J2EE** is not displayed, select the **Show all** check box, click **J2EE**, and click **OK**.

Figure 10. Selecting J2EE from the Select Perspective list



c. If you see the Confirm Enablement window, select **Always enable capabilities and don't ask me again**.

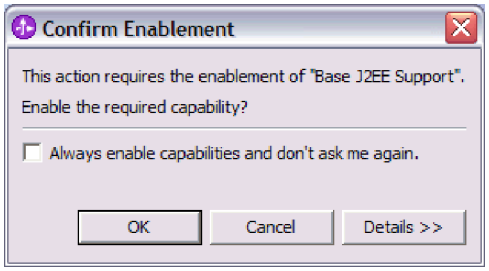
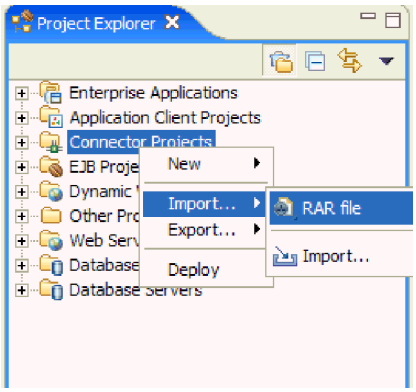


Figure 11. The Confirm Enablement window

d. Click **OK**.

2. Import the RAR file by right-clicking **Connector Projects** and clicking **Import → RAR file**.

Figure 12. Importing the RAR file



3. Find the RAR file on your local file system by clicking **Browse** and navigating to the directory in which Adapter for Siebel Business Applications was installed.

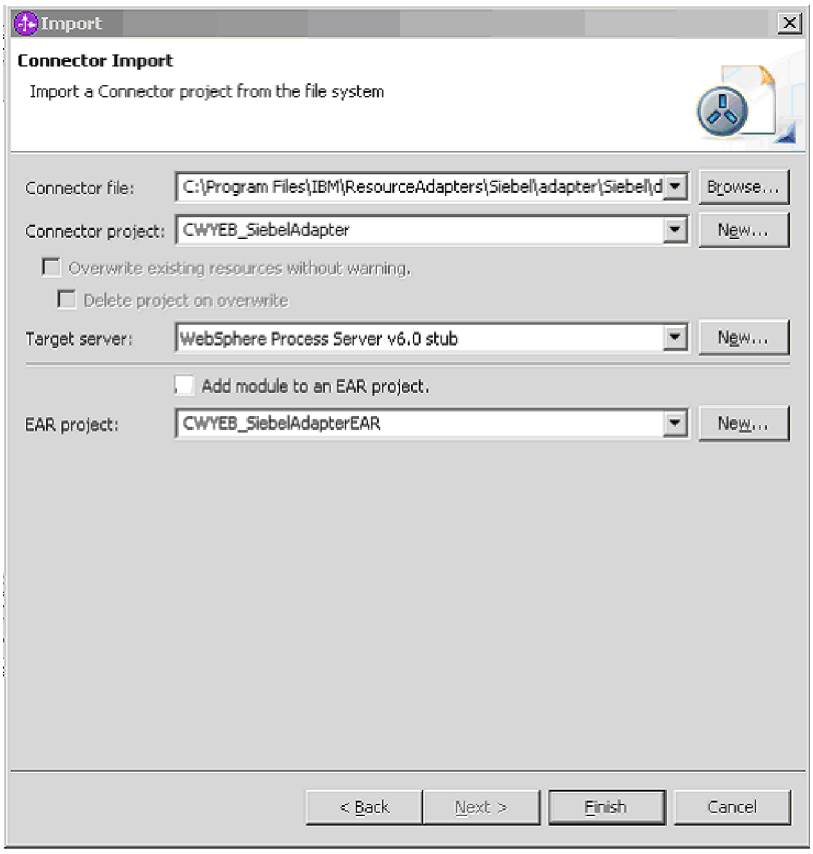


Figure 13. Importing a connector project

4. Accept the default setting (**CWYEB_SiebelAdapter.rar**) for **Connector project**. The connector project has the same name as the RAR file.
5. Accept the default value in the **Target server** field.
   The default value is the test environment for WebSphere Process Server, which is installed as part of WebSphere Integration Developer.
6. Clear the **Add module to an EAR project** check box.
7. Click **Finish**.

Result

A new J2EE connector project, named CWYEB_SiebelAdapter, is created, as shown below. To view the contents of the project, expand the **CWYEB_SiebelAdapter** node.

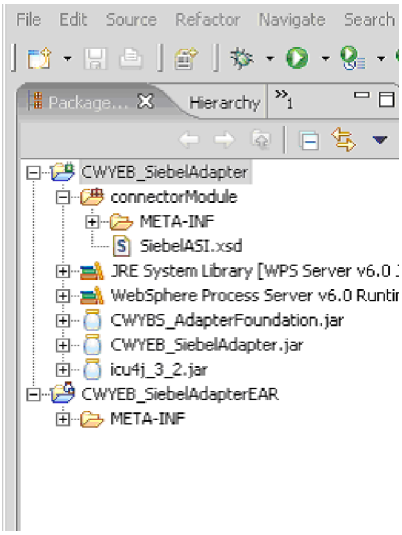


Figure 14. Viewing the contents of the project

What to do next

Add external software dependencies.

Adding external software dependencies

To add the required external dependency files to the adapter project, you first import the files to the connectorModule folder inside the adapter project, then you copy the files from the connectorModule folder of the adapter project to the project classpath.

Before you begin

Create the adapter project in WebSphere Integration Developer. Also, obtain the software dependency files.

About this task

The third party libraries that you add to the connector project are used when you deploy the project.

How to perform this task

The Siebel application requires that you add external software dependencies to the adapter project. These software dependencies enable the Adapter for Siebel Business Applications to communicate with the Siebel environment.

1. Import the Siebel Business Application software dependency files to the connectorModule folder inside the adapter project.

a. In the J2EE perspective of WebSphere Integration Developer, expand the Connector Projects folder, then expand the adapter project.
b. Right-click the **connectorModule** folder, then select **Import**.
c. In the Import window, select **File system** from the list of import sources, then click **Next**.
d. In the File system window, click **Browse**, then navigate to the directory that contains the Siebel Business Applications software dependencies and configuration files, then click **OK**.
e. To select all of the files that are displayed in the right pan of the File system window, click **Select All**, then click **Finish**.

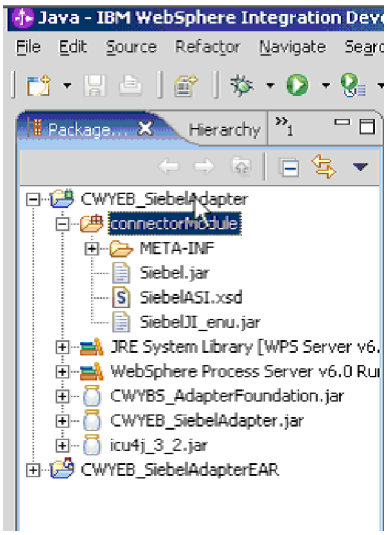


Figure 15. Viewing the dependency JAR files added to the connectorModule folder

2. Import the .jar files into the connectorModule folder.
   a. In the J2EE perspective of WebSphere Integration Developer, expand the Connector Projects folder, then expand the adapter project.
   b. Right-click the **connectorModule** folder, then select **Import**.
   c. In the Import window, select **File system** from the list of import sources, then click **Next**.
   d. In the File system window, click **Browse** to navigate to the following directory:*WebSphere_Integration_Developer_Installation_Directory*\runtimes\bi_v6\lib.

      Note: If you accepted the default installation directory when you installed WebSphere Integration Developer, the .jar files are located in the following directory: C:\Program Files\IBM\WebSphere\ID\6.0\runtimes\bi_v6\lib.
3. Add the software dependency files from the connectorModule folder to the adapter project classpath.
   a. Right-click the adapter project, then select **Properties**.
   b. In the Properties for CWYEB_Siebel Adapter window, select **Java Build Path** from the left pane.
   c. In the Libraries page of the right pane, click **Add JARs**.
   d. In the JAR Selection window, expand the adapter project folder (CWYEB_Siebel Adapter), then expand the connectorModule folder.

e. Highlight all of the JAR files listed under the connectorModule folder, then click **OK**.

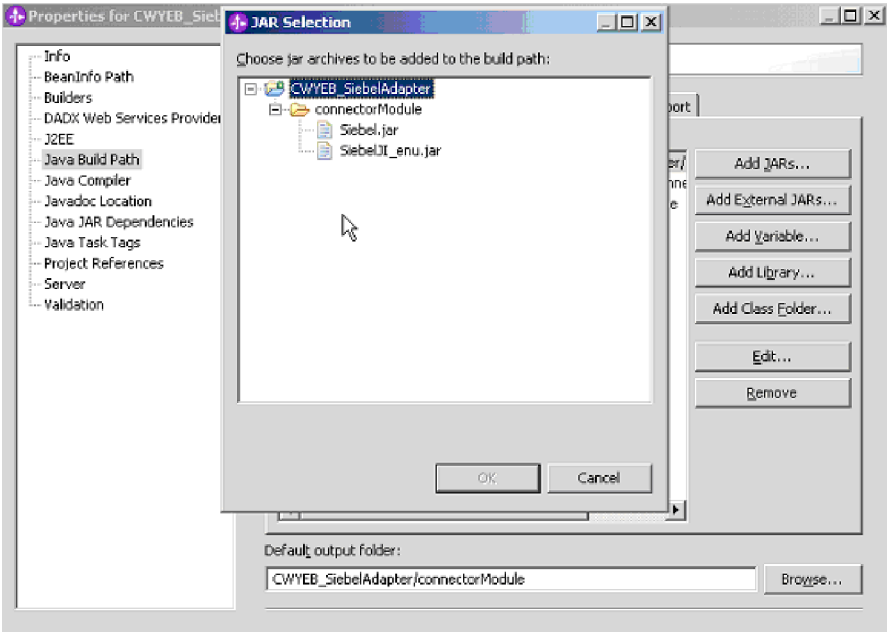


Figure 16. JAR files added to the connectorModule folder

f. In the Properties window, click **OK**.

Result

The external dependencies are displayed in the adapter project folder.

What to do next

Configure the adapter.

Tutorial 1: Outbound processing for Siebel business services

In this tutorial you configure the adapter for outbound processing; deploy; and test the module for processing the Siebel business service, Siebel Account. After working through the tutorial, use the Siebel client to clear the tutorial content of the Siebel business service, Siebel Account.

Before you begin

If you have not done so already, create an authentication alias on the server to process outbound requests; create an adapter project; and add external software dependencies.

Configuring the adapter

Use the enterprise service discovery wizard to set connection properties, select business objects, and generate artifacts.

Setting connection properties for enterprise service discovery

Use the enterprise service discovery wizard tool to introspect the Siebel application and locate the connection properties that are needed for the application and the adapter to communicate.

Before you begin

You must have created an adapter project with the external dependencies added to the adapter project.

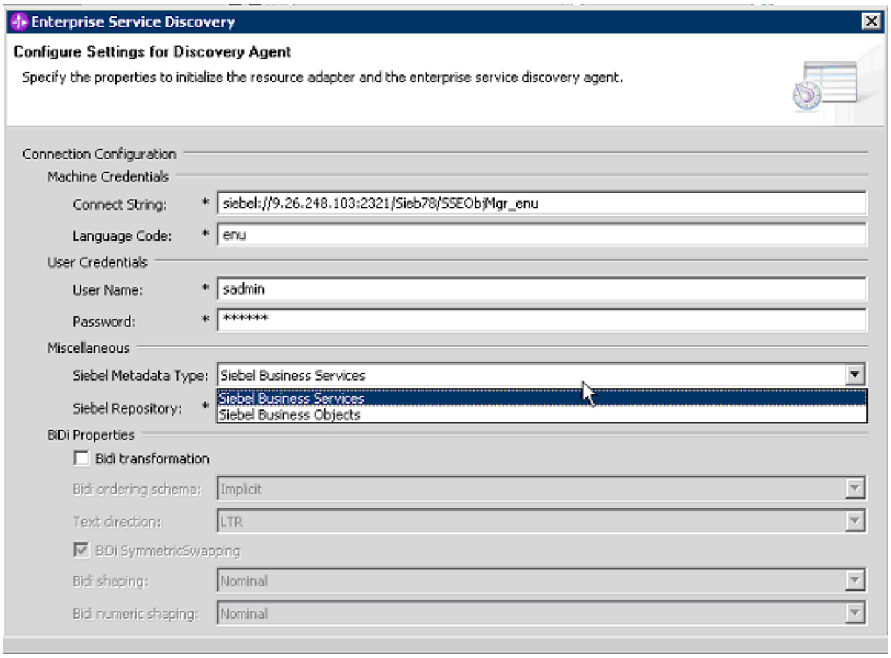
About this task

To set the outbound connection properties so that the application and the adapter can communicate once the adapter is deployed to the server, use the following procedure.

How to perform this task

1. From the WebSphere Integration Developer window, switch to the business integration perspective by selecting **Window** → **Open Perspective** → **Other** → **Business Integration** from the menu bar.
2. From the **File** menu, select **New** → **Enterprise Service Discovery**.
3. From the Select an Enterprise Service Resource Adapter window, select the **IBM WebSphere Adapter for Siebel Business Applications**, and click **Import Resource Adapter**.
4. From the Configure settings for Discovery Agent window, provide values for the following connection properties:
   - Connect String
   - Language Code
   - User Name
   - Password
5. In the **Siebel Metadata Type** field, select **Siebel Business Services**.
6. In the **Siebel Repository** field, type a repository name. The default name is Siebel Repository.
7. Select the BiDi transformation check box if bidirectional language text support is needed.

Figure 17. Discovery agent configuration examples



8. **Optional:** To set the logging level and the prefix, perform the following steps:
   a. At the bottom of the window, click Show Advanced.
   b. Set the Logging Level. In a test environment, select FINEST, which provides the highest level of logging. In a production environment, choose a level lower than FINEST to optimize the logging process.
   c. Set the Prefix.
9. Click **Next**.

What to do next

Select the Siebel business services to use with the adapter.

Selecting the Siebel business service: Siebel Account

Browse the metadata information of the EIS and select the Siebel business service, Siebel Account, to use for processing outbound requests.

Before you begin

The connection properties must be set for the enterprise service discovery wizard.

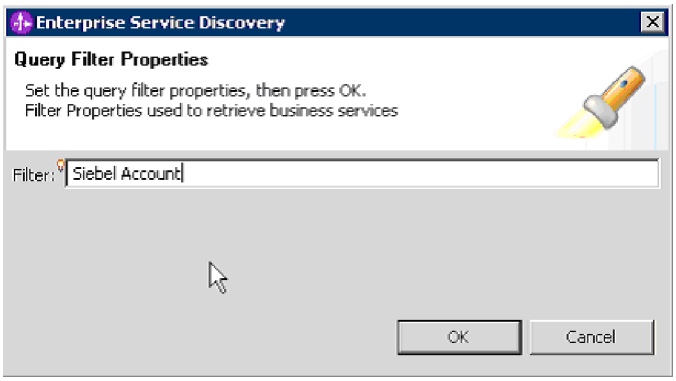
About this task

To select the Siebel business service, Siebel Account, for processing outbound requests, use the following procedure.

How to perform this task

1. From the Find and Discover Enterprise Services window, click **Edit Query**.
2. In the **Filter** field, enter **Siebel Account**, and click **OK**.

Figure 18. Using the query filter



3. From the Find and Discover Enterprise Services window, click **Run Query**.
4. From the query results displayed in the tree structure, expand the *Siebel Account* business service node.
5. For the business service methods, select the **Insert** and **QueryByExample** methods, and click **Add to import list**.

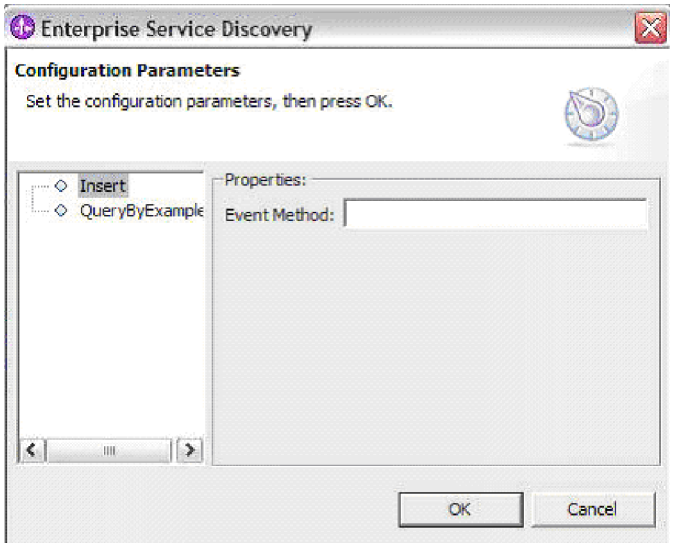


Figure 19. Selecting the business service methods

6. To close the Configuration Parameters window, click **OK**, and then click **Next**.

Result

You have selected the business services that are used to configure outbound event processing.

What to do next

Generate business object definitions and related artifacts.

Generating artifacts

Use the enterprise service discovery wizard to generate artifacts for use with your adapter project. When you generate artifacts, you are adding instructions to the metadata that you extracted from the enterprise information file system. This process also bundles everything together to create an assembled adapter application, also known as an SCA module.

Before you begin

You must have already selected WebSphere business objects to add to the adapter project. You must also have created an authentication alias in your server.

About this task

To configure the artifacts that are deployed to the server, use the following procedure.

How to perform this task

1. From the Configure Objects window, specify the properties for the objects that will be imported by the discovery agent:
   a. In the **Service Type** field, select **Outbound**.
   b. Accept the default in the **Namespace** field provided. The value for Namespace is initially set to the default for all business objects. Do not change the Namespace value.
   c. Enter a **Business object location** and click **Next**. The business object location is where the generated WebSphere business objects are stored. This location is created as a folder under the top-level module folder.
2. Create a new business integration module:
   a. From the Generate Artifacts window, click **New**.
   b. From the Integration Project window, select **Create a module project**, and click **Next**.
   c. In the **Module Name** field, type the name, **Siebel_BS_Outbound**, and click **Finish**.
3. In the **Folder** field, type a folder name. The corresponding folder is created under the module. This is the folder where the generated *.import* and *.wsdl* files are stored.
4. Select the **Use Discovered Connections** property radio button.
5. Provide the authentication alias name that you created in the WebSphere Process Server for the Password property.
6. Provide the values for the properties listed. Example values are shown in the following table.

Table 19. Property examples

Property	Possible value
Adapter ID	ResourceAdapter
Log file size	500000
Log file name	C:\BSOutboundlog.log
Log files	1
Trace file size	500000
Trace file name	C:\BSOutboundTrace.trc
Trace files	1

7. Select the **Resonate Support** check box if your Siebel server supports resonate support. Generally, the Siebel server administrator can answer this question.
8. Type a value for **Siebel View Mode**, and click **Finish**. Using the default (3) allows you to see all the views.

Result

The Siebel_BS_Outbound module is displayed in the J2EE perspective of WebSphere Integration Developer with ″App″ appended to its name, indicating that the module is a deployable application.

What to do next

Deploy the module for testing.

Deploying the module for testing

To deploy the module to the application server, add the module to the server view in the WebSphere Integration Developer integration test client. By completing this step, you automatically install and start the module on the server.

Before you begin

You must have a configured project module that is free of build errors.

About this task

To deploy the module to the server, use the following procedure.

How to perform this task

1. In the Servers View in WebSphere Integration Developer, right-click on the server, and select **Add and remove projects**
2. From the Available Projects in the left pane, select the project, **Siebel_BO_InboundApp**, and click **Add**. The Siebel_BS_outboundApp project is added to the Configured Projects pane on the right.
3. To successfully install and start the SCA module on the server, click **Finish**.

Result

You have successfully installed and started the service component architecture (SCA) module on the server.

What to do next

Test the module. However, if you experience problems installing the SCA module on the server, as an alternative approach, you can use the WebSphere Process Server administrative console to install the SCA module as an EAR file.

Testing the module

Test the module by using the WebSphere Integration Developer integration test client. The Insert and QueryByExample operations are tested against the business service, Siebel Account.

Testing the Insert operation

Test the execution of the Insert operation against the business service, Siebel Account, by using the WebSphere Integration Developer integration test client.

Before you begin

The module must be deployed to the project test environment.

About this task

You must test the module in the project test environment to ensure that it works properly.

How to perform this task

1. From the *Business Integration* profile, select the **Siebel_BS_OutboundApp** module.
2. Right-click on the **Siebel_BS_OutboundApp** module and select **Test → Test Module**.
3. From the Events window, select the **insertSiebelAccountInsertAccountInterface** operation against the Operation property.
4. Right-click the **SiebelMessage** attribute and click **Add Element**.

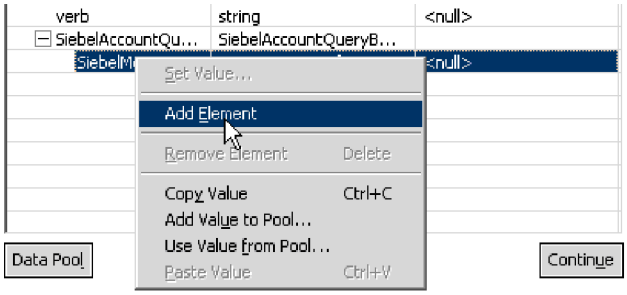


Figure 20. Adding an element

5. Set the values as needed against the attributes of the **SiebelMessage** element.

   Note: Under the SiebelMessage instance (of IOAccountInterfaceICAccount type), the attributes, Name, PrimaryOrganization, RelatedOrganization, RelatedSalesRep, Location, and CurrencyCode, are mandatory attributes that must have values set. For the remaining ones, there is no need to set the values unless there is a specific need in your environment to set the attributes.

   The values provided below are samples that may or not be valid in your environment. You can get the valid values for the attributes by checking existing enterprise information system records for the *Account* integration object.

Table 20. Sample attribute values

Attribute	Attribute value
CurrencyCode	USD
Location	Burlingame
Name	TestSample
PrimaryOrganization	(This can be blank or contain the name of the organization)
RelatedOrganization	The attribute, Organization, should be set to the value, Default Organization.
RelatedSalesRep	The attribute, Login, should be set the value, sadmin. The attribute, Position, should be set the value, Siebel Administrator.

6. For the container attributes under the SiebelMessage instance, select the null value, **<null>** for the container attributes (mandatory) under the SiebelMessage instance, with the exception of the attributes, RelatedOrganization and RelatedSalesRep.
7. Set the rest of the simple non-required attributes to **<unset>**.
   a. To set the attribute, select one or more attributes.
   b. Right-click and select, **Set-value**.
   c. Enter the value, **<null>**.
8. In the *SiebelAccountInsertAccountInterfacewindow* status-object attribute, set the value to **<unset>**, click **OK** and click **Continue**.
9. To execute the operation, select the appropriate WebSphere Process Server instance, and click **Finish**.

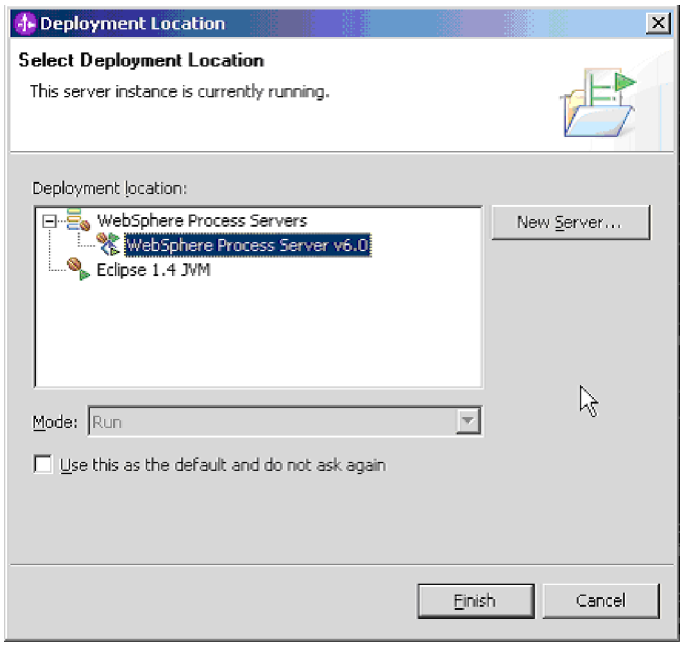


Figure 21. Selecting the deployment location

When the execution of the insert operation is successfully completed, the resultant business object with the key values that were filled in are returned, shown in the following figure. Make note of the AccountId key value because it is used later to clear the sample content that you created.

For Insert, the result object might only contain key field values, which means common non-key field values might be null. This will depend on the Siebel method design.

SiebelMessage	IOAccountInterfaceICAccount [ ]	
SiebelMessage[0]	IOAccountInterfaceICAccount	
AccountId	String	1-CT8
AccountStatus	String	
Alias	String	
AssignmentAreaC...	String	
AssignmentCountr...	String	
CompetitorFlag	String	
CreditAutoApprov...	String	
CreditStatusDate	String	
CurrencyCode	String	
DUNSNumber	String	

Figure 22. Business objects showing key value examples

What to do next

Test the QueryByExample operation.

Testing the QueryByExample operation

Test the execution of the QueryByExample operation against the business service, Siebel Account, by using the WebSphere Integration Developer integration test client.

Before you begin

The module must be deployed to the project test environment.

About this task

You must test the module in the project test environment to ensure that it works properly.

How to perform this task

1. From the *Business Integration* profile, select the **Siebel_BS_OutboundApp** module.
2. Right-click on the **Siebel_BS_OutboundApp** module and select **Test → Test Module**.
3. From the Events window, select the **querybyexampleSiebelAccountQueryByExampleAccountInterface** operation against the Operation property.
4. Right-click the **SiebelMessage** attribute and click **Add Element**.
5. Set the values as needed against the attributes of the **SiebelMessage** element.

   Get the AccountId key value from any existing account record in the Siebel enterprise information system and set the value against the AccountId attribute of the SiebelMessage added element.

   You can use the values of the AccountId that were generated earlier against the Insert operation. You can also get the AccountId values from any existing record in the enterprise information system.

   Note: Do not use the example value set below. The AccountId that was generated earlier against the Insert operation is set below. If you used the Insert operation prior to the execution of this operation, you would be able to use the values. If not, get the AccountId from any existing record in the enterprise information system.

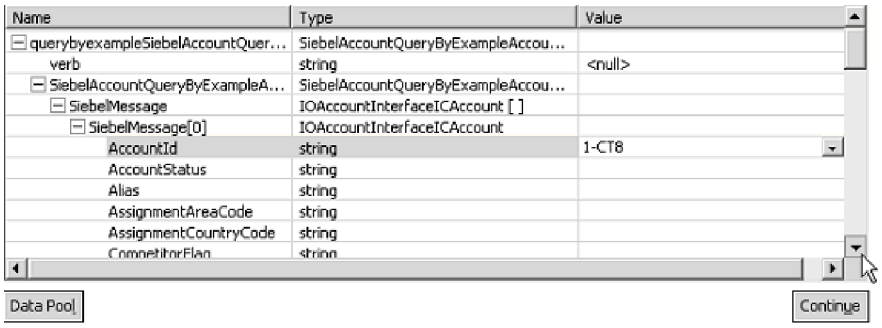


Figure 23. Business objects showing key value examples

6. Set the simple attributes with values:
   a. Select the simple attributes, other than AccountId.
   b. Right-click the selected attributes list (use either the *Shift* or *Ctrl* keys), and select **Set Value**.
   c. Type the value, **<unset>**, click **OK**, and click **Continue**.
7. To execute the operation, select the appropriate WebSphere Process Server instance, and click **Finish**.

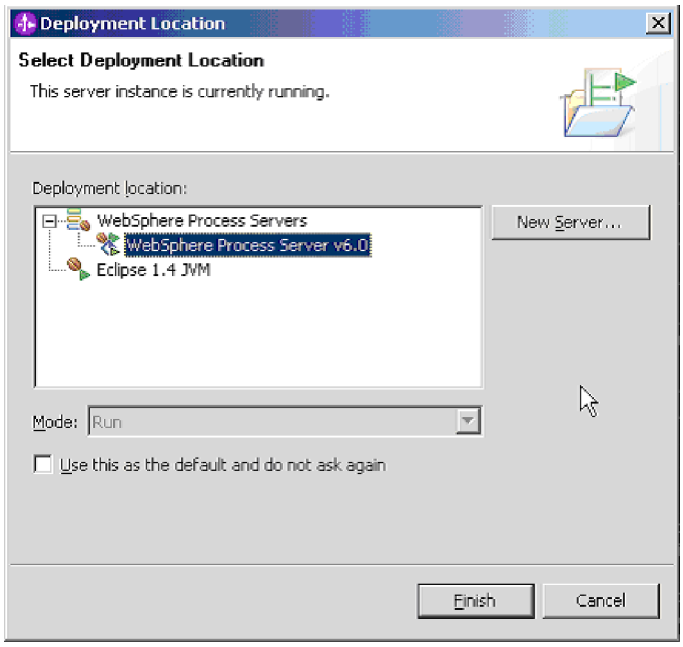


Figure 24. Selecting the deployment location

When the execution of the QueryByExample operation is successfully completed, the resultant business object with the key values that were filled in are returned as shown below.

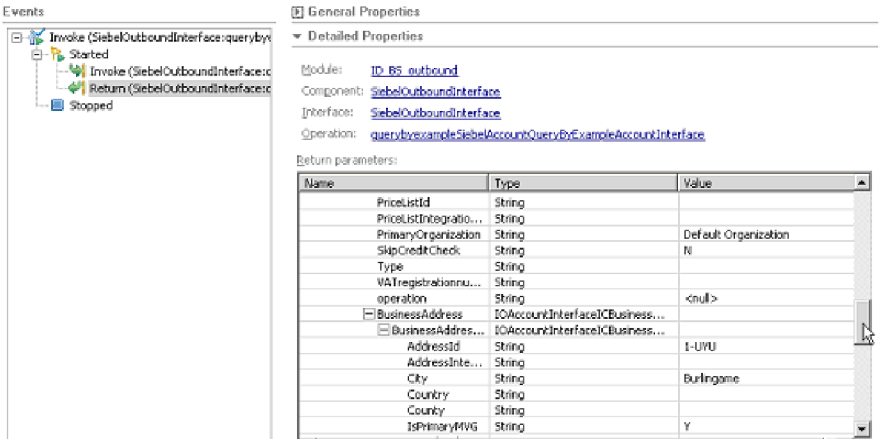


Figure 25. Business objects showing key value examples

Troubleshooting the tutorial

If you encountered the "EMD, child picklist business component" error, follow the instructions for the EMD defect workaround. If you were unable to deploy the adapter, as an alternative method, use WebSphere Process Server to deploy the enterprise archive file to the application server.

EMD defect workaround

If you encountered the "EMD defect, child picklist business component" error, follow these instructions for the EMD defect workaround. In addition, if you were unable to deploy the adapter, as an alternative method, use WebSphere Process Server to deploy the enterprise archive file to the application server.

Before you begin

Use the following instructions if you were unsuccessful in executing the Create operation.

About this task

To edit the *createBOAccountU45EXPBC Account* operation, use the following procedure.

How to perform this task

1. Scroll down to the attribute, **PrimaryBillToLastName**.
2. Go to **Properties view** → **Application info**.
3. Under **ASIelement** properties, go to **sasi:SiebelContainerAttributeMetadataForBC>sasi:PickList**.
4. Change to sasi: From to Id:AccountId
5. Change to sasi: To to AccountId

Exporting the project module as an EAR file

If you are unable to deploy the adapter using WebSphere Integration Developer, use the WebSphere Process Server administrative console to deploy the enterprise

archive (EAR) file to the application server. By creating an EAR file, you capture all of the contents of your adapter project in a format can be easily deployed to the application server.

Before you begin

You must have created a configured project that is free of build errors.

About this task

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

How to perform this task

1. Confirm that there are no errors by building the project.
2. Right-click on the project and select **Export → EAR file**.
3. In the EAR Export window, select the EAR project.
4. Provide the absolute path, including the EAR file name, for the Destination. Examples of absolute paths, including the EAR file name, are C:\SiebelBuild\Siebel_BS_OutboundApp.ear and C:\SiebelBuild\Siebel_BO_InboundApp.ear.
5. Select the following options:
   - Export source files
   - Overwrite existing file
   - Include project build paths and metadata files
6. Click **Finish**.

Result

The EAR file now contains an enterprise information system import.

Deploying the EAR using WebSphere Process Server:

Use the WebSphere Process Server administrative console to deploy the enterprise archive (EAR) file, as an alternative to using WebSphere Integration Developer.

Before you begin

You must have an adapter project that has been exported to an enterprise archive (EAR) file.

About this task

To deploy the EAR file using the WebSphere Process Server administrative console, use the following procedure.

How to perform this task

1. In the bottom right side window of WebSphere Integration Developer, click **Servers** view.
2. If not already started, right-click on the WebSphere Process Server instance and start the WebSphere Process Server instance.
3. Confirm that the server is up by checking the status, it should be listed as started.

4. Start the administrative console by right-clicking on the server and selecting **Run administrative console**.
5. In the administrative console window, click **login**.
6. Select **Applications → Enterprise Applications**.

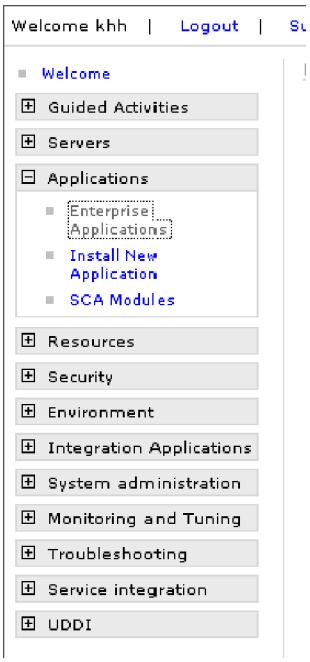


Figure 26. Selecting Enterprise Applications

7. From the Enterprise Applications window, select **Install**.

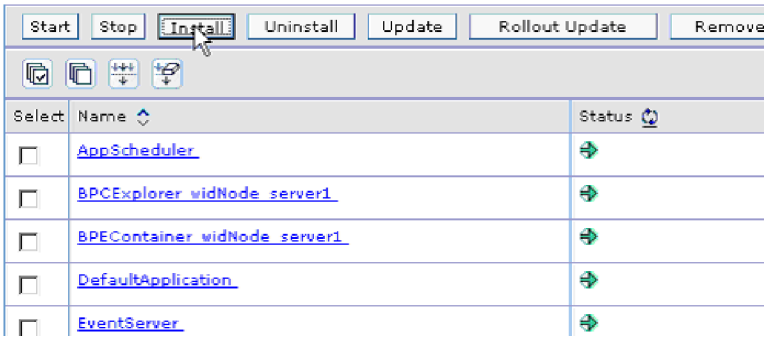


Figure 27. Selecting the Install option

8. Open the EAR file that you want to deploy:
   a. In the Path to the new application window, choose the system where the EAR file was saved.
   b. Browse to the corresponding folder on the system.
   c. Select the EAR file from the folder where you it saved earlier, click **Open**, and click **Next**.

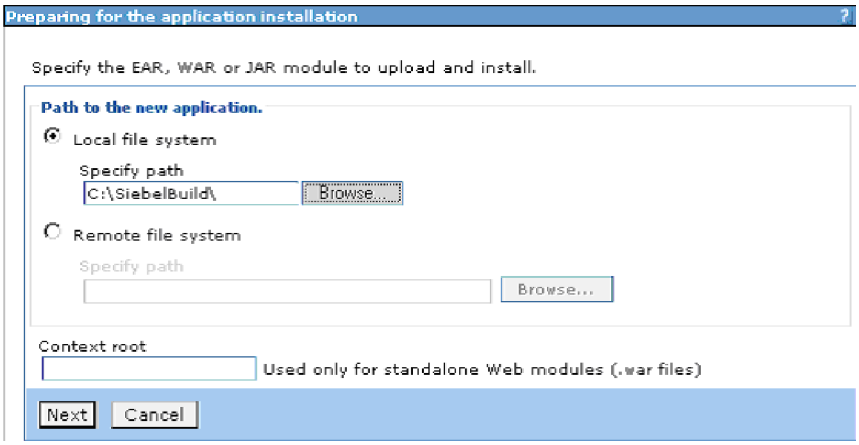


Figure 28. Specifying the module to install

9. From the Preparing the application for installation window, click **Next**.

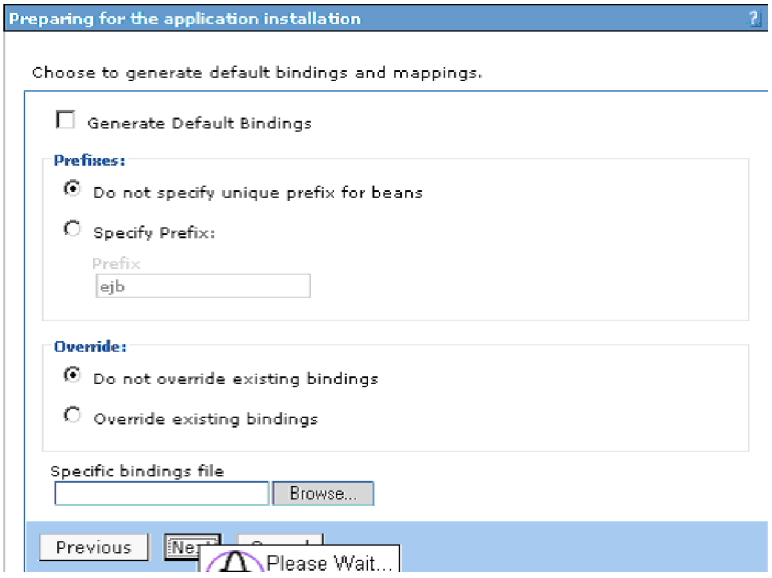


Figure 29. Generating default bindings and mappings

10. From the Install New Application window, click **Step 9 Summary**, then click **Finish**. The project is installed. When installation is complete, a message confirming the successful installation of the application appears.
11. From the same window where the message appears, click **Save to Master Configuration**.
12. From the Enterprise Applications window, click **Save**.

Result

The application is now installed on the server.

What to do next

Start the application.

Starting the application using WebSphere Process Server:

Use the WebSphere Process Server administrative console to start the application.

Before you begin

You must have deployed the adapter project through the WebSphere Process Server administrative console.

About this task

Use the WebSphere Process Server administrative console to start the application to make sure that it works properly.

How to perform this task

1. From the Enterprise Applications window, select the check-box next to the installed application, and click **Start**.
2. Confirm that the application has been started successfully. A message is provided at the top of the window and the Status against the application changes from an X to an arrow.

Clearing the tutorial content

Clear the Insert operation tutorial content for the Siebel Account business service.

Before you begin

You should have noted down the AccountID generated earlier against the business service Insert operation.

How to perform this task

Using the Siebel client, delete the corresponding *Account* record. More information on deleting records, see the documentation for the Siebel client.

Note: Clearing is not required for the QueryByExample operation because this operation retrieves an existing record and does not create or update any sample content in the Siebel enterprise information system.

Tutorial 2: Outbound processing for Siebel business objects

In this tutorial you configure the adapter for outbound processing; deploy; and test the module for processing Siebel business objects. After working through the tutorial, use the Siebel client to clear the tutorial content of the Siebel business object, Account – ESP.

Before you begin

If you have not done so already, create an authentication alias on the server to process outbound requests; create an adapter project; and add external software dependencies.

Configuring the adapter

Use the enterprise service discovery wizard to set connection properties, select business objects, and generate artifacts.

Setting connection properties for enterprise service discovery

Use the enterprise service discovery wizard tool to introspect the Siebel application and locate the connection properties that are needed for the application and the adapter to communicate.

Before you begin

You must have created an adapter project with the external dependencies added to the adapter project.

About this task

To set the outbound connection properties so that the application and the adapter can communicate once the adapter is deployed to the server, use the following procedure.

How to perform this task

1. From the WebSphere Integration Developer window, switch to the business integration perspective by selecting **Window** → **Open Perspective** → **Other** → **Business Integration** from the menu bar.
2. From the **File** menu, select **New** → **Enterprise Service Discovery**.
3. From the Select an Enterprise Service Resource Adapter window, select the **IBM WebSphere Adapter for Siebel Business Applications**, and click **Next**.
4. From the Configure settings for Discovery Agent window, provide values for the following connection properties:
   - Connect string
   - User name
   - Password
   - Language code
5. In the **Siebel Metadata Type** field, select **Siebel Business Objects** or **Siebel Business Object**.
6. In the **Siebel Repository** field, type a repository name. The default name is Siebel Repository.
7. Select the BiDi transformation check box if bidirectional language text support is needed and click **Next**.

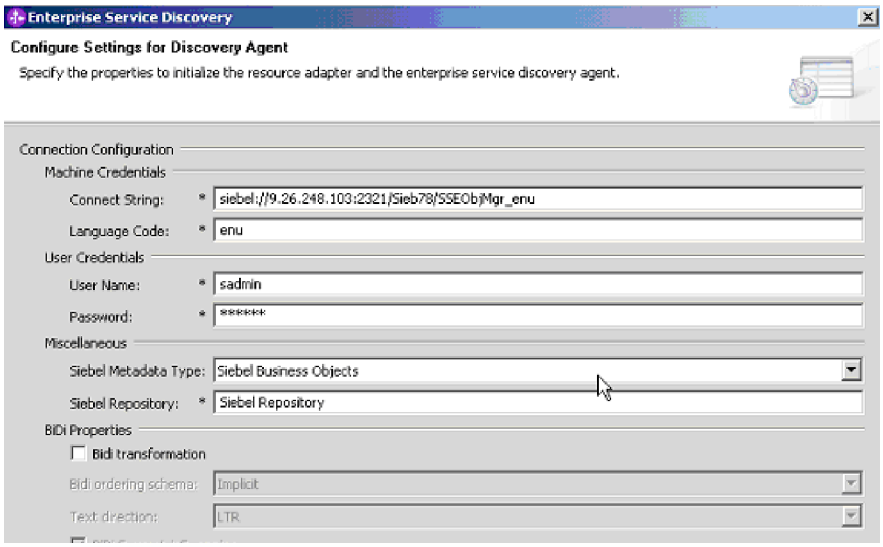


Figure 30. Discovery agent configuration examples

8. **Optional:** To set the logging level and the prefix, perform the following steps:
   a. At the bottom of the window, click Show Advanced.
   b. Set the Logging Level. In a test environment, select FINEST, which provides the highest level of logging. In a production environment, choose a level lower than FINEST to optimize the logging process.
   c. Set the Prefix.

What to do next

Select the Siebel business objects to use with the adapter.

Selecting the Siebel business object: Account – ESP

Browse the metadata information of the enterprise information system and select the Siebel business object, Account – ESP, to use for processing outbound requests.

Before you begin

The connection properties must be set for the enterprise service discovery wizard.

About this task

To select the Siebel business object, Account – ESP, to use for processing outbound requests, use the following procedure.

How to perform this task

1. From the Find and Discover Enterprise Services window, click **Edit Query**.

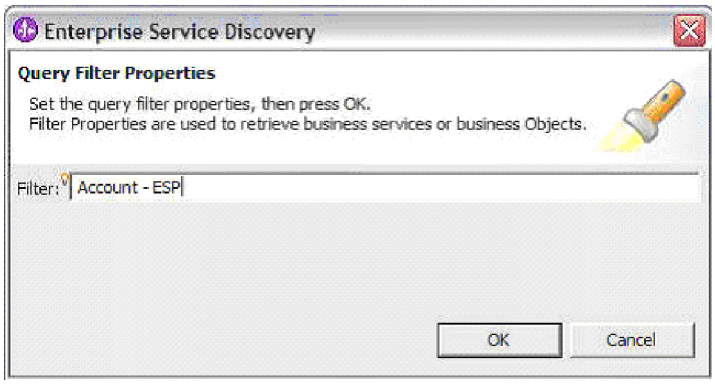


Figure 31. Using the query filter

2. In the **Filter** field, enter **Account – ESP**, and click **OK**.
3. From the Find and Discover Enterprise Services window, click **Run Query**.
4. From the query results displayed in the tree structure, expand the *Account – ESP* business object node, select **Account – ESP**, click **Add to import list**, and click **Next**.
5. To close the windows, click **OK**, then click **Next**.

Result

You have selected the business objects that will be used to configure outbound event processing.

What to do next

Generate business object definitions and related artifacts for outbound processing.

Generating artifacts

Use the enterprise service discovery wizard to generate artifacts for use with your adapter project. When you generate artifacts, you are adding instructions to the metadata that you extracted from the enterprise information file system. This process also bundles everything together to create an assembled adapter application, also known as an SCA module.

Before you begin

You must have already selected WebSphere business objects to add to the adapter project. You must also have created an authentication alias in your server.

About this task

To configure the artifacts that are deployed to the server, use the following procedure.

How to perform this task

1. From the Configure Objects window, specify the properties for the objects that will be imported by the discovery agent:
   a. In the **Service Type** field, select **Outbound**.
   b. Accept the default in the **Namespace** field provided. The value for Namespace is initially set to the default for all business objects. Do not change the Namespace value.

c. Enter a **Business object location** and click **Next**. The business object location is where the generated WebSphere business objects are stored. This location is created as a folder under the top-level module folder.

2. Create a new business integration module:
   a. From the Generate Artifacts window, click **New**.
   b. From the Integration Project window, select **Create a module project**, and click **Next**.
   c. In the **Module Name** field, type the name, **Siebel_BO_Outbound**, and click **Finish**.
3. In the **Folder** field, type a folder name. The corresponding folder is created under the module. This is the folder where the generated *.import* and *.wsdl* files are stored.
4. Select the **Use Discovered Connections** property radio button.
5. Provide the authentication alias name that you created in the WebSphere Process Server for the Password property.
6. Provide the values for the properties listed. Example values are shown in the following table.

Table 21. Property examples

Property	Possible value
Adapter ID	ResourceAdapter
Log file size	500000
Log file name	C:\BOOutboundlog.log
Log files	1
Trace file size	500000
Trace file name	C:\BOOutboundTrace.trc
Trace files	1

7. Select the **Resonate Support** check box if you have resonate support. Generally, the Siebel server administrator can answer this question.
8. Type a value for **Siebel View Mode**, and click **Finish**. Using the default (3) allows you to see all the views.

Result

The Siebel_BO_Outbound module is displayed in the J2EE perspective of WebSphere Integration Developer with ″App″ appended to its name, indicating that the module is a deployable application.

What to do next

Deploy the module for testing.

Deploying the module for testing

To deploy the module to the application server, add the module to the server view in the WebSphere Integration Developer integration test client. By completing this step, you automatically install and start the module on the server.

Before you begin

You must have a configured project module that is free of build errors.

About this task

To deploy the module to the server, use the following procedure.

How to perform this task

1. In the Servers View in WebSphere Integration Developer, right-click on the server, and select **Add and remove projects**
2. From the Available Projects in the left pane, select the project, **Siebel_BO_outboundApp**, and click **Add**. The Siebel_BO_outboundApp project is added to the Configured Projects pane on the right.
3. To successfully install and start the SCA module on the server, click **Finish**.

Result

You have successfully installed and started the SCA module on the server.

What to do next

If you experience problems installing the SCA module on the server, as an alternative approach, export an EAR file using the WebSphere Process Server administrative console and install the module.

Testing the module

Test the adapter application by using the WebSphere Integration Developer integration test client. The two operations used in the tutorial, Create and Exists, are tested against the business object, Account – ESP.

Testing the Create operation

Test the eCreate operation against the business object, Account – ESP, using the WebSphere Integration Developer integration test client.

Before you begin

The adapter project, also known as the SCA module, must be deployed to the project test environment.

About this task

You test the module in the project test environment to ensure that it works properly.

How to perform this task

1. From the *Business Integration* view, select the **Siebel_BO_OutboundApp** module.
2. Right-click on the **Siebel_BO_OutboundApp** module and select **Test → Test Module**.
3. From the Events window, select the **createBOAccountU45ESPBCAccount** operation against the Operation property. U45 in the operation name is the Unicode space character.
4. From the list of verbs, select **Create**.
5. Provide sample values for the business object attributes.

   The values provided below are samples that may or not be valid in your environment.

Table 22. Sample attribute values

Attribute	Possible sample value
CurrencyCode	USD
InternalOrgFlag	N
Name	AccountSample4ID
PartyTypeCode	P
PartyUID	Q

Note: If you run this test again, make sure that you use a unique Name and PartryUID for each run, otherwise an error occurs.

6. Set the attribute values under the **BOAccountU45ESPBCAccount** instance.
   a. Select the value, **<null>**.
   b. Select the simple attributes to be set (use the *Shift* or *Ctrl* keys), right-click on the selected attributes list, and select **Set Value**.
   c. From the Set Value window, type the value, **<unset>**.
7. Click **OK**, then click **Continue**.
8. To execute the operation, select the appropriate WebSphere Process Server instance, and click **Finish**.

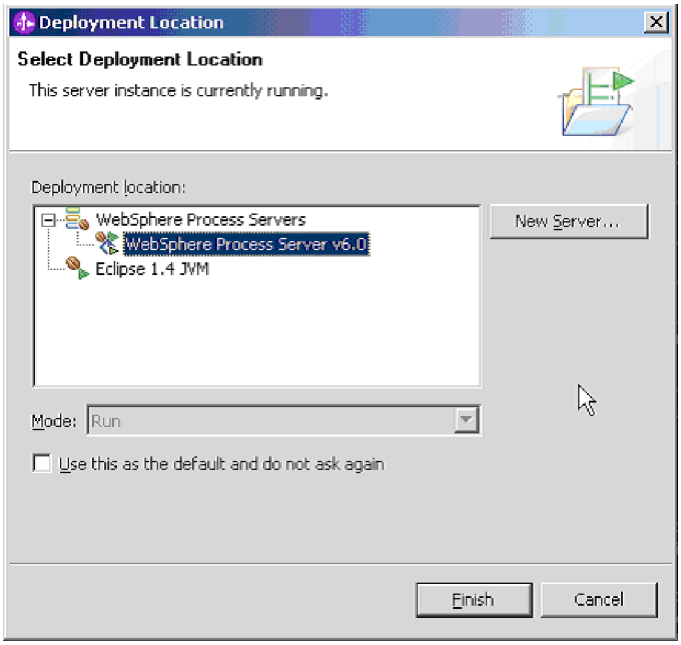


Figure 32. Selecting the deployment location

Result

When the execution of the Create operation is successfully completed, the resultant business object with the key values that were filled in are returned. If you are unsuccessful and encounter the error, "The child PickLlist business component with keys is not found: Contact...," check the troubleshooting section for information on the EMD defect workaround.

What to do next

Make note of the Id key value because it is used later to clear the sample content that you created.

Testing the Exists operation

Test the execution of the Exists operation against the business object, Account - ESP, by using the WebSphere Integration Developer integration test client.

Before you begin

The adapter project, also known as the SCA module, must be deployed to the project test environment.

About this task

You test the module in the project test environment to ensure that it works properly.

How to perform this task

1. From the *Business Integration* profile, select the **Siebel_BS_OutboundApp** module.
2. Right-click on the **Siebel_BS_OutboundApp** module and select **Test → Test Module**.
3. From the Events window, select the **existsBOAccountU45ESPBCAccount** operation against the Operation property.
4. Get the Id key value from any existing account record in the Siebel enterprise information system and set the value against the Id attribute of the SiebelMessage added element.

   You can use the value of the Id that was generated earlier against the Create operation. You can also get the Id value from any existing record in the enterprise information system.
5. Set the attribute values, other than ID.
   a. Select the value, **Set Value**.
   b. Select the simple attributes to be set (use the *Shift* or *Ctrl* keys), right-click on the selected attributes list, and select **Set Value**.
   c. From the Set Value window, type the value, **<unset>**.
6. Click **OK** and click **Continue**.
7. To execute the operation, select the appropriate WebSphere Process Server instance, and click **Finish**.

Result

When the execution of the Exists operation is successfully completed, the resultant business object of type, SiebelExistsResult, is returned, as shown below. This business object has a single Boolean type variable. The value returned is true if the object was present in the Siebel enterprise information system and false if it was not present.

Troubleshooting the tutorial

If you encountered the ″EMD, child picklist business component″ error, follow the instructions for the EMD defect workaround. If you were unable to deploy the adapter, as an alternative method, use WebSphere Process Server to deploy the enterprise archive file to the application server.

Exporting the project module as an EAR file

If you are unable to deploy the adapter using WebSphere Integration Developer, use the WebSphere Process Server administrative console to deploy the enterprise archive (EAR) file to the application server. By creating an EAR file, you capture all of the contents of your adapter project in a format can be easily deployed to the application server.

Before you begin

You must have created a configured project that is free of build errors.

About this task

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

How to perform this task

1. Confirm that there are no errors by building the project.
2. Right-click on the project and select **Export → EAR file**.
3. In the EAR Export window, select the EAR project.
4. Provide the absolute path, including the EAR file name, for the Destination. Examples of absolute paths, including the EAR file name, are C:\SiebelBuild\Siebel_BS_OutboundApp.ear and C:\SiebelBuild\Siebel_BO_InboundApp.ear.
5. Select the following options:
   - Export source files
   - Overwrite existing file
   - Include project build paths and metadata files
6. Click **Finish**.

Result

The EAR file now contains an enterprise information system import.

Deploying the EAR using WebSphere Process Server

Use the WebSphere Process Server administrative console to deploy the enterprise archive (EAR) file, as an alternative to using WebSphere Integration Developer.

Before you begin

You must have an adapter project that has been exported to an enterprise archive (EAR) file.

About this task

To deploy the EAR file using the WebSphere Process Server administrative console, use the following procedure.

How to perform this task

1. In the bottom right side window of WebSphere Integration Developer, click **Servers** view.
2. If not already started, right-click on the WebSphere Process Server instance and start the WebSphere Process Server instance.

3. Confirm that the server is up by checking the status, it should be listed as started.
4. Start the administrative console by right-clicking on the server and selecting **Run administrative console**.
5. In the administrative console window, click **login**.
6. Select **Applications → Enterprise Applications**.

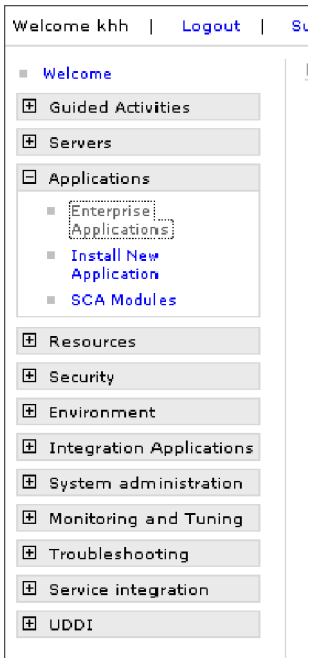


Figure 33. Selecting Enterprise Applications

7. From the Enterprise Applications window, select **Install**.

Figure 34. Selecting the Install option

8. Open the EAR file that you want to deploy:
   a. In the Path to the new application window, choose the system where the EAR file was saved.
   b. Browse to the corresponding folder on the system.
   c. Select the EAR file from the folder where you it saved earlier, click **Open**, and click **Next**.

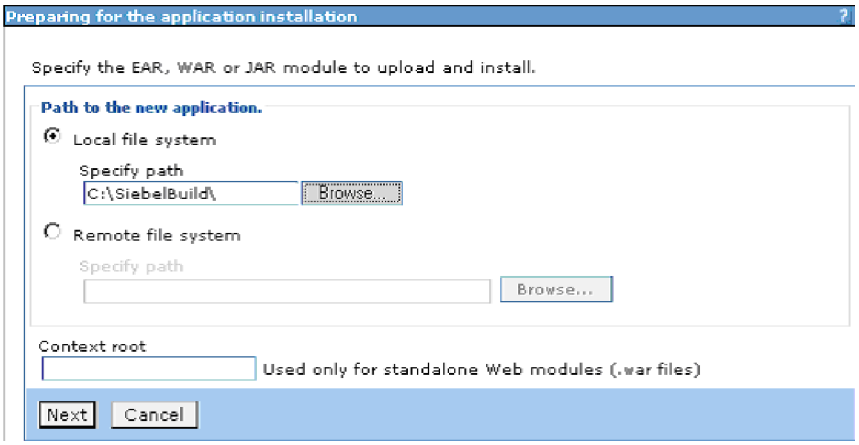


Figure 35. Specifying the module to install

9. From the Preparing the application for installation window, click **Next**.

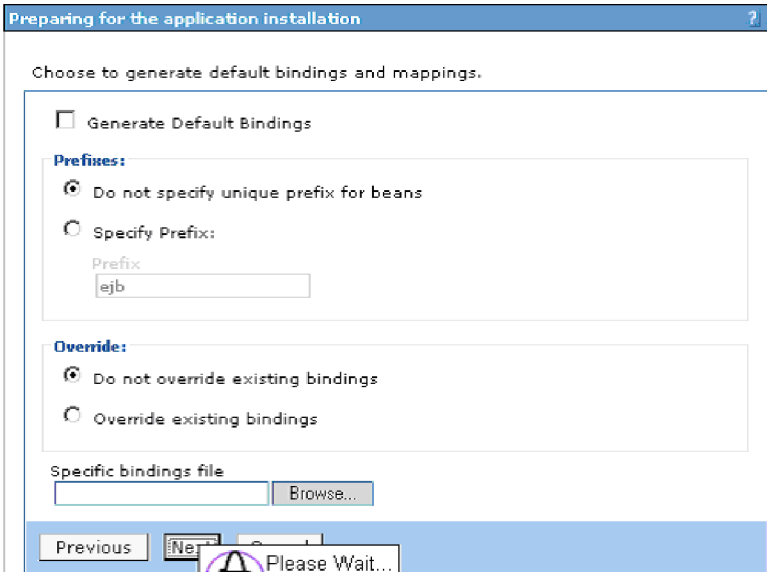


Figure 36. Generating default bindings and mappings

10. From the Install New Application window, click **Step 9 Summary**, then click **Finish**. The project is installed. When installation is complete, a message confirming the successful installation of the application appears.
11. From the same window where the message appears, click **Save to Master Configuration**.
12. From the Enterprise Applications window, click **Save**.

Result

The application is now installed on the server.

What to do next

Start the application.

Starting the application using WebSphere Process Server

Use the WebSphere Process Server administrative console to start the application.

Before you begin

You must have deployed the adapter project through the WebSphere Process Server administrative console.

About this task

Use the WebSphere Process Server administrative console to start the application to make sure that it works properly.

How to perform this task

1. From the Enterprise Applications window, select the check-box next to the installed application, and click **Start**.
2. Confirm that the application has been started successfully. A message is provided at the top of the window and the Status against the application changes from an X to an arrow.

Clearing the tutorial content

Clear the Create operation tutorial content for the Siebel business object, Account – ESP.

Before you begin

You should have noted down the AccountID generated earlier against the business object Create operation.

How to perform this task

Using the Siebel client, delete the corresponding *Account* record. More information on deleting records, see the documentation for the Siebel client.

Note: Clearing is not required for the Exists operation because this operation retrieves an existing record and does not create or update any sample content in the Siebel enterprise information system.

Tutorial 3: Inbound processing for Siebel business services

In this tutorial, you configure the adapter for inbound processing; deploy; and test the module for processing the Siebel business service, Account Interface.

Before you begin

If you have not already done so, create an event table in the Siebel application to track inbound events that occur in the Siebel enterprise information system; create the adapter project; and add external software dependencies.

Configuring the adapter

Use the enterprise service discovery wizard to set connection properties, select business objects, and generate artifacts.

Setting connection properties for enterprise service discovery

Use the Configure settings for Discovery Agent window in the enterprise service discovery wizard to set the inbound connection properties that are needed for the application and the adapter to communicate.

Before you begin

You must have created an adapter project with the external dependencies added to the adapter project. In addition, you must have completed Tutorial 1: Outbound processing for Siebel business services.

About this task

To set the connection properties so that the application and the adapter can communicate once the adapter is deployed to the server, use the following procedure.

How to perform this task

1. From the WebSphere Integration Developer window, switch to the business integration perspective by selecting **Window** → **Open Perspective** → **Other** → **Business Integration** from the menu bar.
2. From the **File** menu, select **New** → **Enterprise Service Discovery**.
3. From the Select an Enterprise Service Resource Adapter window, select the **IBM WebSphere Adapter for Siebel Business Applications**, and click **Next**.
4. From the Configure settings for Discovery Agent window, provide values for the following connection properties:
   - Connect string
   - User name
   - Password
   - Language code
5. In the **Siebel Metadata Type** field, select **Siebel Business Services**.
6. In the **Siebel Repository** field, type the repository name. The default name is Siebel Repository.
7. Select the BiDi transformation check box if bidirectional language text support is needed.

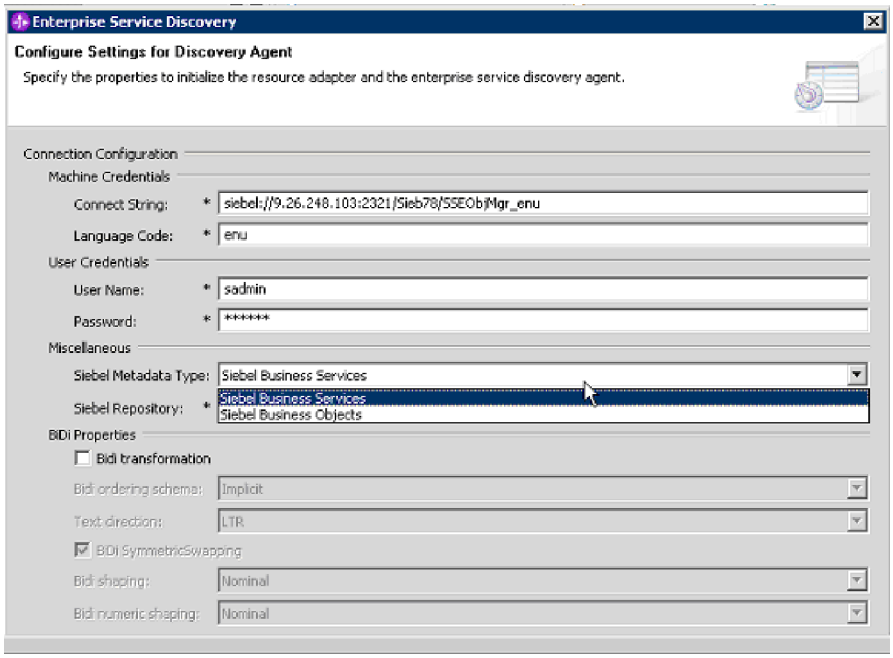


Figure 37. Discovery agent configuration examples

8. **Optional:** To set the logging level and the prefix, perform the following steps:
   a. At the bottom of the window, click Show Advanced.
   b. Set the Logging Level. In a test environment, select FINEST, which provides the highest level of logging. In a production environment, choose a level lower than FINEST to optimize the logging process.
   c. Set the Prefix.
9. Click **Next**.

What to do next

Select the Siebel business services to use with the adapter.

Selecting the Siebel business service: Siebel Account

Browse the metadata information of the EIS, select artifacts of interest, and generate deployable Siebel integration objects to configure inbound event processing.

Before you begin

After you set the connection properties for the enterprise service discovery wizard, you select Siebel business services from the enterprise information system to configure the adapter.

How to perform this task

1. From the Find and Discover Enterprise Services window, click **Edit Query**.

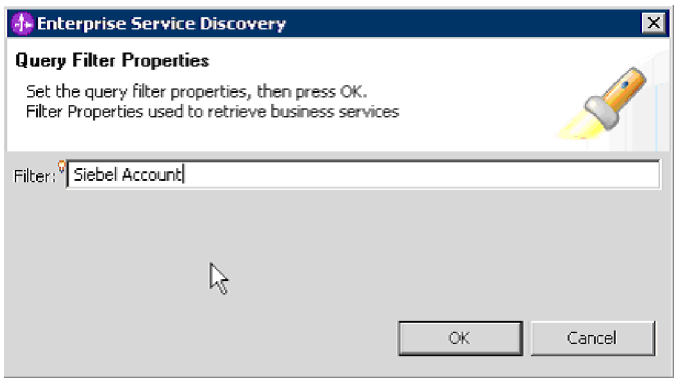


Figure 38. The Query Filter Properties window

2. In the **Query Filter Properties** field, type **Siebel Account**, click **OK**, and click **Run Query**.

   The query results are displayed in a tree structure.
3. From the search results, under the objects discovered by the query, expand the Siebel Account business service node, select the business service, **QueryByExample**, and click **Add to import list**.

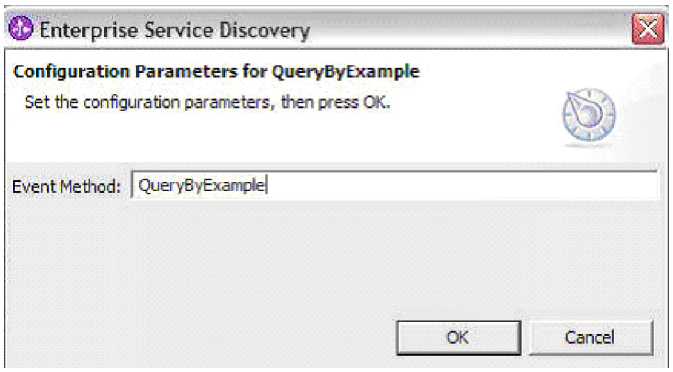


Figure 39. The Configuration Parameters for QueryByExample window

4. From the Configuration parameters window, set the event method property to **QueryByExample**, and click **OK**.
5. For the inbound service type, enter an Event Method. For the inbound Siebel integration object, you can use *QueryByExample*. You can remove an object by selecting the object from the lower pane of the Find and Discover Enterprise Services window and clicking **Remove**.
6. When you finish making your selections, click **Next**.

Result

You have selected the business integration objects that will be used to configure inbound event processing.

What to do next

Generate business object definitions and related artifacts for inbound processing.

Generating artifacts

Use the enterprise service discovery wizard to generate artifacts for use with your adapter project. When you generate artifacts, you are adding instructions to the metadata that you extracted from the enterprise information file system. This process also bundles everything together to create an assembled adapter application, also known as an SCA module.

Before you begin

You must have already selected WebSphere business objects to add to the adapter project. You must also have created an authentication alias in your server.

About this task

To configure the artifacts that are deployed to the server, use the following procedure.

How to perform this task

1. From the Configure Objects window, specify the properties for the objects that will be imported by the discovery agent:
   a. In the **Service Type** field, select **Inbound**.
   b. Accept the default in the **Namespace** field provided. The value for Namespace is initially set to the default for all business objects. Do not change the Namespace value.
   c. Enter a **Business object location** and click **Next**. The business object location is where the generated WebSphere business objects are stored. This location is created as a folder under the top-level module folder.
2. Create a new business integration module:
   a. From the Generate Artifacts window, click **New**.
   b. From the Integration Project window, select **Create a module project**, and click **Next**.
   c. In the **Module Name** field, type the module name, **Siebel_ BS_Inbound**, and click **Finish**.
3. In the **Folder** field, type a folder name. The corresponding folder is created under the module. This is the folder where the generated *.import* and *.wsdl* files are stored.
4. Select the **Use Discovered Connections** property radio button.
5. Provide the authentication alias name that you created in the WebSphere Process Server for the Password property.
6. Provide the values for the properties listed. Example values are shown in the following table.

Table 23. Property examples

Property	Possible value
Event component name	IBM2
Adapter ID	ResourceAdapter
Log file size	500000
Log file name	C:\BSInboundlog.log
Log files	1
Trace file size	500000

Table 23. Property examples (continued)

Property	Possible value
Trace file name	C:\BSInboundTrace.trc
Trace files	1

7. Select the **Resonate Support** check box if your Siebel server supports resonate support. Generally, the Siebel server administrator can answer this question.
8. Type a value for **Siebel View Mode**, and click **Finish**. Using the default (3) allows you to see all the views.

Result

The Siebel_BS_Inbound module is displayed in the J2EE perspective of WebSphere Integration Developer with ″App″ appended to its name, indicating that the module is a deployable application.

What to do next

Generate reference bindings for the adapter.

Generating reference bindings (test environment only)

Generate reference bindings to create a reference in the assembly editor from the adapter project to a stand-alone reference. The stand-alone reference represents an a generic J2EE component, such as the application server. By wiring the adapter project to the stand-alone reference, you link the adapter to other server processes.

Before you begin

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

About this task

To generate reference bindings to bind to the service component, use the following procedure.

How to perform this task

1. From the Business Integration perspective in the WebSphere Integration Developer main window, beneath **All Resources**, open the assembly diagram under your project, and select the **SiebelInboundInterface** export.
2. From the left pane, select the icon, **Component with no Implementation type**.
3. From the icon list that appears, select the icon **Component with no Implementation type** again, and click on the assembly diagram window.

   The new element, Component1, is added, as shown below.

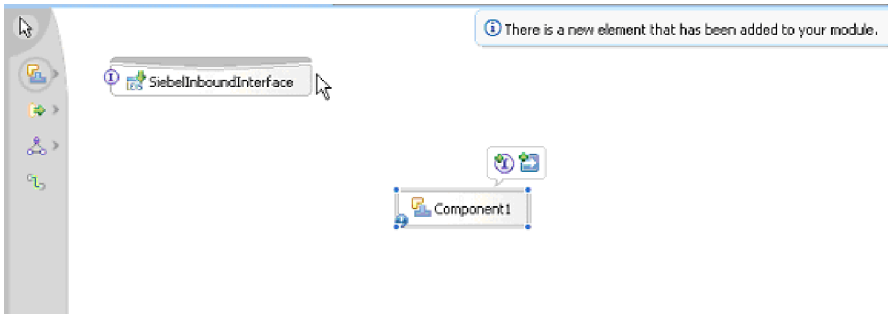


Figure 40. Adding the Component1 element

4. Use the Link icon to link the new element, **Component1**, with the export, **SiebelInboundInterface**.

   The Component1 element, is linked with the SiebelInboundInterface export, as shown below.

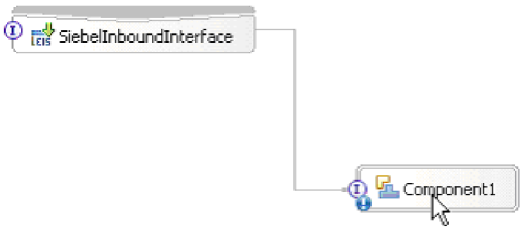


Figure 41. The new Component1 element

5. When the Add Wire window appears, click **OK**.
6. Right-click on the new element, Component1, and select **Generate Implementation** → **Java**. The class file, Component1Impl.java, is generated.
7. From the **Generate Implementation** window, ensure that **(default package)** is selected, and click **OK**.

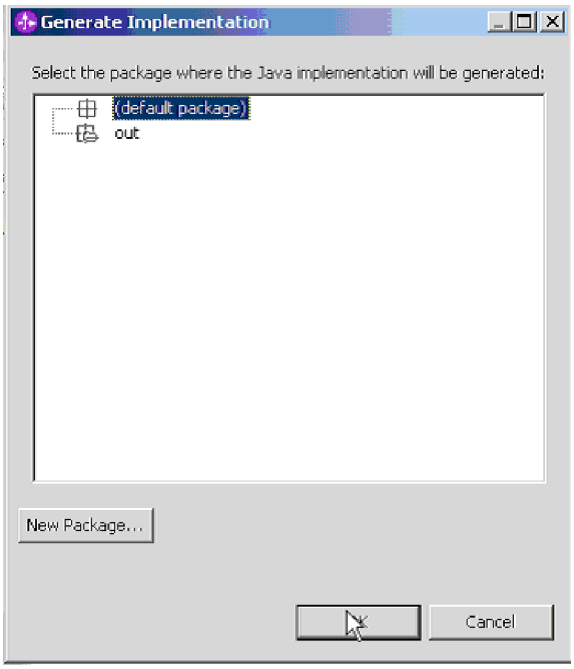


Figure 42. Generating the Java implementation

8. Add **System.out.println** statements in the emit methods of the class file, Component1Impl.java, as shown.

```
public void emitCreateAfterImageIOAccountInterfaceICAccount(
        DataObject emitCreateAfterImageIOAccountInterfaceICAccountInput) {
    System.out.println("-------Event generated for 'Create' was polled successfully by adapter and delivered.");
}

/**
 * Method generated to support implemention of operation "emitUpdateAfterImageIOAccountInterfaceICAccount" defined
 * named "interface.SiebelInboundInterface".
 *
 * The presence of commonj.sdo.DataObject as the return type and/or as a parameter
 * type conveys that its a complex type. Please refer to the WSDL Definition for more information
 * on the type of input, output and fault(s).
 */
public void emitUpdateAfterImageIOAccountInterfaceICAccount(
        DataObject emitUpdateAfterImageIOAccountInterfaceICAccountInput) {
    System.out.println("-------Event generated for 'Update' was polled successfully by adapter and delivered.");
}

/**
 * Method generated to support implemention of operation "emitDeleteAfterImageIOAccountInterfaceICAccount" defined
 * named "interface.SiebelInboundInterface".
 *
 * The presence of commonj.sdo.DataObject as the return type and/or as a parameter
 * type conveys that its a complex type. Please refer to the WSDL Definition for more information
 * on the type of input, output and fault(s).
 */
public void emitDeleteAfterImageIOAccountInterfaceICAccount(
        DataObject emitDeleteAfterImageIOAccountInterfaceICAccountInput) {
    System.out.println("-------Event generated for 'Delete' was polled successfully by adapter and delivered.");
}
```

Figure 43. Class file example

9. When you finish with your changes, save the class file and then the project.

   Note: Select the assembly diagram window before you save the project with the SiebelInboundInterface export. If you do not the select the assembly diagram window, the changes cannot be saved.

Result

The result of running the enterprise service discovery wizard is an service component architecture (SCA) module that contains an enterprise information system (EIS) import.

What to do next

Install the module on the WebSphere Integration Developer integration test client.

Deploying the module for testing

To deploy the module to the application server, add the module to the server view in the WebSphere Integration Developer integration test client. By completing this step, you automatically install and start the module on the server.

Before you begin

You must have a configured project module that is free of build errors.

About this task

Install the SCA module that you created on the WebSphere Integration Developer integration test client to the server using the Servers View in WebSphere Integration Developer.

How to perform this task

1. Right-click on the server, and select **Add and remove projects**.
2. From Available Projects in the left pane, select the project, **Siebel_BS_InboundApp**, and click **Add**. The project is added to the Configured Projects pane on the right.
3. Click **Finish**.

Result

The SCA module has been successfully added to the server.

Note: If you have any problems installing the application using the above approach, you can export the project into an EAR, install and start the project through the WebSphere Process Server administrative console.

Testing the assembled adapter application

Use the Siebel application event component to test whether the event was successfully polled and delivered by the adapter.

Before you begin

You should have an service component (SCA) module installed on the server.

About this task

Create a new record in the event component in the Siebel application to test the polling and delivery of an event by the adapter.

How to perform this task

1. Create a new record in the event component in the Siebel application, as shown:

Siebel application event component table

Event component attribute	Attribute value
Object name	IOAccountInterfaceICAccountBG
Event type	Update **Note:** This value is dependent on having completed Tutorial 1: Outbound processing for Siebel business services.
Status	0
Object key	Name=SampleTest

Note: If you provided any prefix in the connection properties, you must add the prefix to the object name. The provided object key value is a sample only. You must set an appropriate name by checking the existing Account record in the enterprise information system.

2. In the WebSphere Process Server Admin Console window, select **Applications** → **Enterprise Applications**.
3. Confirm that the application, SiebelBS_InboundApp, has been started successfully. A message is provided at the top of the window and also the status against the application will change from an "x" to an arrow.
4. To check the WebSphere Process Server log information, select the **Console** tab at the bottom of the window.

   Assuming the event was successfully polled and delivered by the adapter, you should see the System.out.println() statement, added earlier in the class, on the console window.

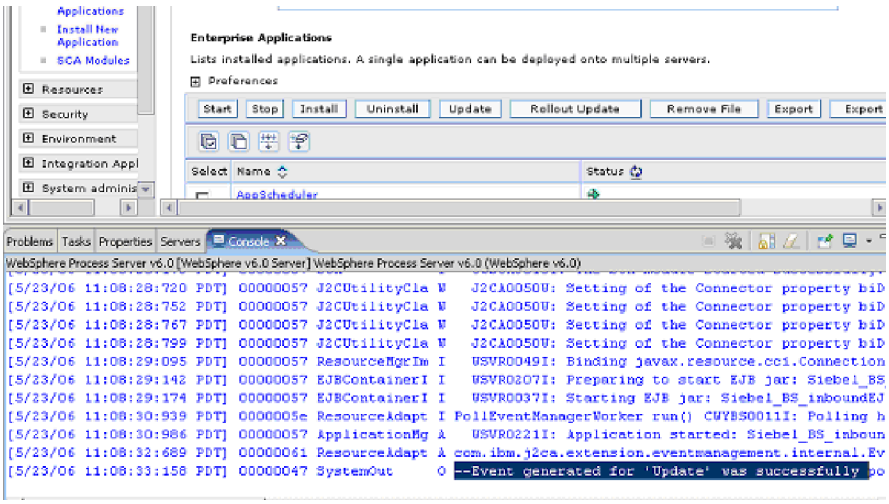


Figure 44. System.out.println() statement example

Result

This concludes the successful polling and delivery of an event by the adapter using the business service, Siebel Account, and its integration object, Account Interface. You can also confirm the success by checking the event component using the Siebel client. The corresponding event record will have been deleted from the event component.

Troubleshooting the tutorial

If you encountered the ″EMD, child picklist business component″ error, follow the instructions for the EMD defect workaround. If you were unable to deploy the adapter, as an alternative method, use WebSphere Process Server to deploy the enterprise archive file to the application server.

Exporting the project module as an EAR file

If you are unable to deploy the adapter using WebSphere Integration Developer, use the WebSphere Process Server administrative console to deploy the enterprise archive (EAR) file to the application server. By creating an EAR file, you capture all of the contents of your adapter project in a format can be easily deployed to the application server.

Before you begin

You must have created a configured project that is free of build errors.

About this task

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

How to perform this task

1. Confirm that there are no errors by building the project.
2. Right-click on the project and select **Export → EAR file**.
3. In the EAR Export window, select the EAR project.
4. Provide the absolute path, including the EAR file name, for the Destination. Examples of absolute paths, including the EAR file name, are C:\SiebelBuild\Siebel_BS_OutboundApp.ear and C:\SiebelBuild\Siebel_BO_InboundApp.ear.
5. Select the following options:
   - Export source files
   - Overwrite existing file
   - Include project build paths and metadata files
6. Click **Finish**.

Result

The EAR file now contains an enterprise information system import.

Deploying the EAR using WebSphere Process Server

Use the WebSphere Process Server administrative console to deploy the enterprise archive (EAR) file, as an alternative to using WebSphere Integration Developer.

Before you begin

You must have an adapter project that has been exported to an enterprise archive (EAR) file.

About this task

To deploy the EAR file using the WebSphere Process Server administrative console, use the following procedure.

How to perform this task

1. In the bottom right side window of WebSphere Integration Developer, click **Servers** view.
2. If not already started, right-click on the WebSphere Process Server instance and start the WebSphere Process Server instance.
3. Confirm that the server is up by checking the status, it should be listed as started.
4. Start the administrative console by right-clicking on the server and selecting **Run administrative console**.
5. In the administrative console window, click **login**.
6. Select **Applications** → **Enterprise Applications**.

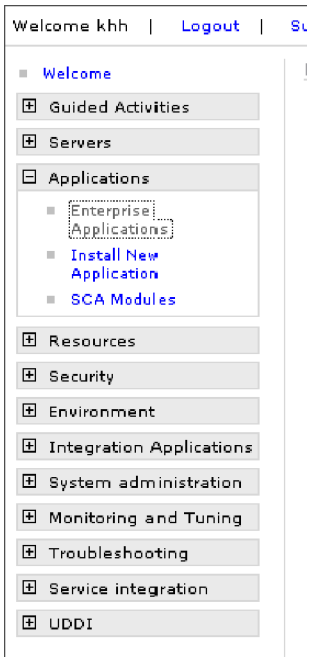


Figure 45. Selecting Enterprise Applications

7. From the Enterprise Applications window, select **Install**.

Figure 46. Selecting the Install option

8. Open the EAR file that you want to deploy:

a. In the Path to the new application window, choose the system where the EAR file was saved.
b. Browse to the corresponding folder on the system.
c. Select the EAR file from the folder where you it saved earlier, click **Open**, and click **Next**.

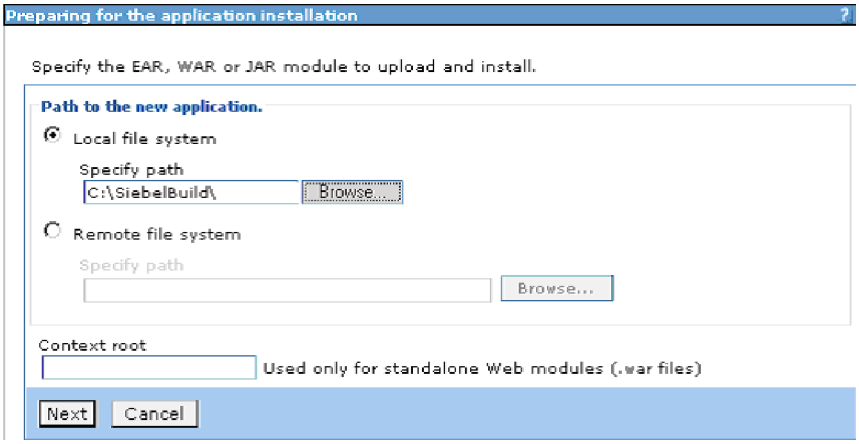


Figure 47. Specifying the module to install

9. From the Preparing the application for installation window, click **Next**.

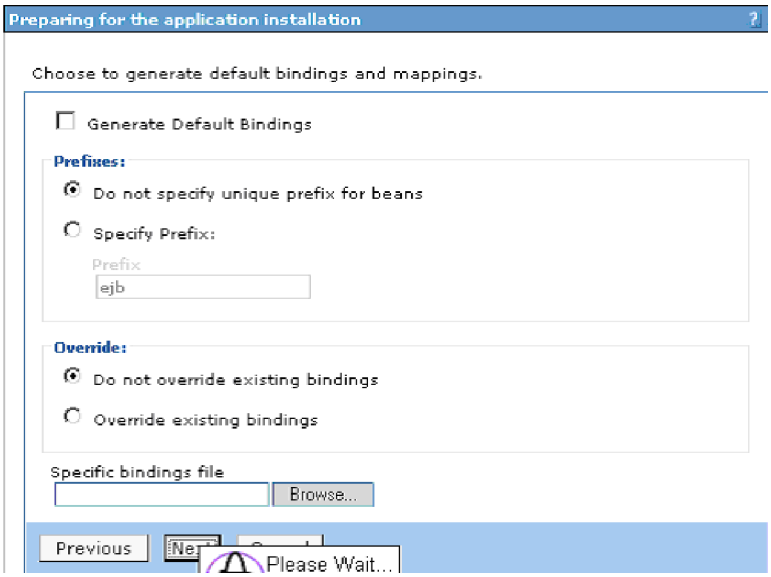


Figure 48. Generating default bindings and mappings

10. From the Install New Application window, click **Step 9 Summary**, then click **Finish**. The project is installed. When installation is complete, a message confirming the successful installation of the application appears.
11. From the same window where the message appears, click **Save to Master Configuration**.
12. From the Enterprise Applications window, click **Save**.

Result

The application is now installed on the server.

What to do next

Start the application.

Starting the application using WebSphere Process Server

Use the WebSphere Process Server administrative console to start the application.

Before you begin

You must have deployed the adapter project through the WebSphere Process Server administrative console.

About this task

Use the WebSphere Process Server administrative console to start the application to make sure that it works properly.

How to perform this task

1. From the Enterprise Applications window, select the check-box next to the installed application, and click **Start**.
2. Confirm that the application has been started successfully. A message is provided at the top of the window and the Status against the application changes from an X to an arrow.

Tutorial 4: Inbound processing for Siebel business objects

In this tutorial, configure the adapter for inbound processing; deploy; and test the module for processing the Siebel integration object, Account – ESP.

Before you begin

If you have not already done so, create an event table in the Siebel application to track inbound events that occur in the Siebel enterprise information system; create an adapter project; and add external software dependencies.

Configuring the adapter

Use the enterprise service discovery wizard to set connection properties, select business objects, and generate artifacts.

Setting connection properties for enterprise service discovery

Use the Configure settings for Discovery Agent window in the enterprise service discovery wizard to set the inbound connection properties that are needed for the application and the adapter to communicate.

Before you begin

You must have created an adapter project with the external dependencies added to the adapter project.

About this task

To set the connection properties so that the application and the adapter can communicate once the adapter is deployed to the server, use the following procedure.

How to perform this task

1. From the WebSphere Integration Developer window, switch to the business integration perspective by selecting **Window** → **Open Perspective** → **Other** → **Business Integration** from the menu bar.
2. From the **File** menu, select **New** → **Enterprise Service Discovery**.
3. From the Select an Enterprise Service Resource Adapter window, select the **IBM WebSphere Adapter for Siebel Business Applications**, and click **Next**.
4. From the Configure settings for Discovery Agent window, provide values for the following connection properties:
   - Connect string
   - User name
   - Password
   - Language code
5. In the **Siebel Metadata Type** field, select **Siebel Business Objects**.
6. In the **Siebel Repository** field, type the repository name. The default name is Siebel Repository.
7. If bidirectional text language is supported, select the **BiDi transformation** check box, and click **Next**.

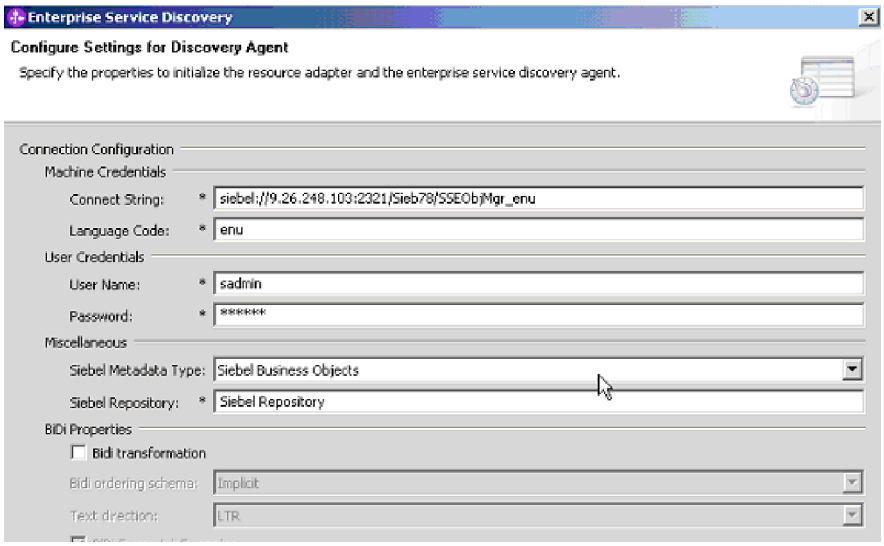


Figure 49. Configuring the settings for the discovery agent

8. **Optional:** To set the logging level and the prefix, perform the following steps:
   a. At the bottom of the window, click Show Advanced.
   b. Set the Logging Level. In a test environment, select FINEST, which provides the highest level of logging. In a production environment, choose a level lower than FINEST to optimize the logging process.
   c. Set the Prefix.

What to do next

Select the Siebel business objects to use with the adapter.

Selecting the Siebel business object: Account – ESP

Browse the metadata information of the EIS, select artifacts of interest, and generate deployable Siebel business objects used to configure inbound event processing.

Before you begin

After you set the connection properties for the enterprise service discovery wizard, you select business objects from the enterprise information system to configure the adapter.

About this task

Select the business services that you will use to configure inbound event processing.

How to perform this task

1. From the Find and Discover Enterprise Services window, click **Edit Query**.

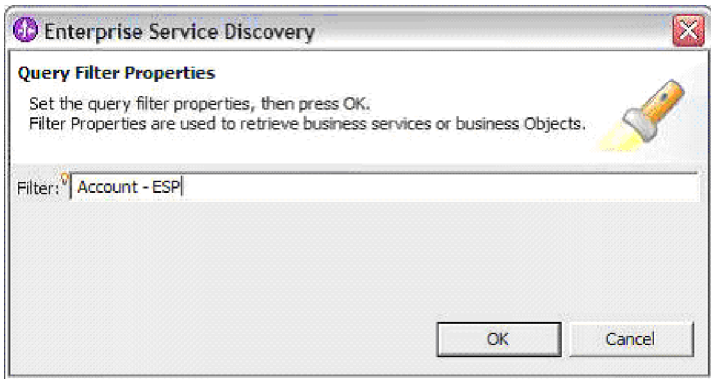


Figure 50. The Query Filter Properties window

2. In the **Query Filter Properties** field, type **Account - ESP**, click **OK**, and click **Run Query**.

   The query results are displayed in a tree structure.

Result

You have selected the business objects or services that will be used to configure inbound event processing.

What to do next

Generate business object definitions and related artifacts for inbound processing.

Generating artifacts

Use the enterprise service discovery wizard to generate artifacts for use with your adapter project. When you generate artifacts, you are adding instructions to the metadata that you extracted from the enterprise information file system. This process also bundles everything together to create an assembled adapter application, also known as an SCA module.

Before you begin

You must have already selected WebSphere business objects to add to the adapter project. You must also have created an authentication alias in your server.

About this task

To configure the artifacts that are deployed to the server, use the following procedure.

How to perform this task

1. From the Configure Objects window, specify the properties for the objects that will be imported by the discovery agent:
   a. In the **Service Type** field, select **Inbound**.
   b. Accept the default in the **Namespace** field provided. The value for Namespace is initially set to the default for all business objects. Do not change the Namespace value.
   c. Enter a **Business object location** and click **Next**. The business object location is where the generated WebSphere business objects are stored. This location is created as a folder under the top-level module folder.
2. Create a new business integration module:
   a. From the Generate Artifacts window, click **New**.
   b. From the Integration Project window, select **Create a module project**, and click **Next**.
   c. In the **Module Name** field, type the module name, **Siebel_BO_Inbound**, and click **Finish**.
3. In the **Folder** field, type a folder name. The corresponding folder is created under the module. This is the folder where the generated *.import* and *.wsdl* files are stored.
4. Select the **Use Discovered Connections** property radio button.
5. Provide the authentication alias name that you created in the WebSphere Process Server for the Password property.
6. Provide the values for the properties listed. Example values are shown in the following table.

Table 24. Property examples

Property	Possible value
Adapter ID	ResourceAdapter
Log file size	500000
Log file name	C:\BOInboundlog.log
Log files	1
Trace file size	500000
Trace file name	C:\BOInboundTrace.trc
Trace files	1

7. Select the **Resonate Support** check box if your Siebel server supports resonate support. Generally, the Siebel server administrator can answer this question.
8. Type a value for **Siebel View Mode**, and click **Finish**. Using the default (3) allows you to see all the views.

Result

The Siebel_BO_Inbound module is displayed in the J2EE perspective of WebSphere Integration Developer with "App" appended to its name, indicating that the module is a deployable application.

What to do next

Generate reference bindings for the adapter.

Generating reference bindings (test environment only)

Generate reference bindings to create a reference in the assembly editor from the adapter project to a stand-alone reference. The stand-alone reference represents an a generic J2EE component, such as the application server. By wiring the adapter project to the stand-alone reference, you link the adapter to other server processes.

Before you begin

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

About this task

To generate reference bindings to bind to the service component, use the following procedure.

How to perform this task

1. From the Business Integration perspective in the WebSphere Integration Developer main window, open the assembly diagram under your project, and select the **SiebelInboundInterface** export.
2. From the left pane, select the icon, **Component with no Implementation type**.
3. From the icon list that appears, select the icon, **Component with no Implementation type** again, and click on the assembly diagram window.

   The new element, Component1, is added, as shown below.

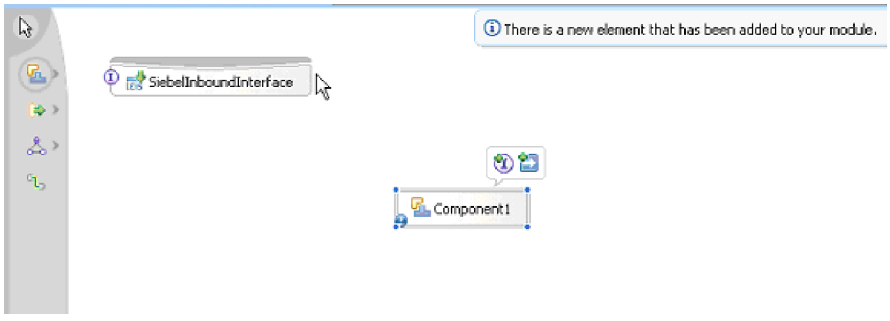


Figure 51. Adding the Component1 element

4. Use the Link icon to link the new element, **Component1**, with the export, **SiebelInboundInterface**.

   The Component1 element, is linked with the SiebelInboundInterface export, as shown below.

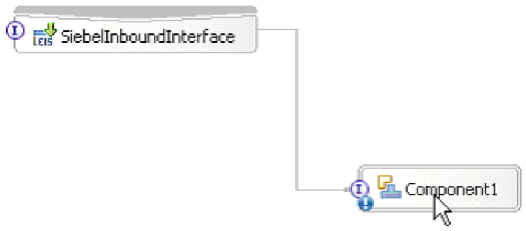


Figure 52. Linking the elements

5. When the Add Wire window appears, click **OK**.
6. Right-click on the new element, Component1, and select **Generate Implementation** → **Java**. The class file, Component1Impl.java, is generated.
7. From the **Generate Implementation** window, ensure that **(default package)** is selected, and click **OK**.

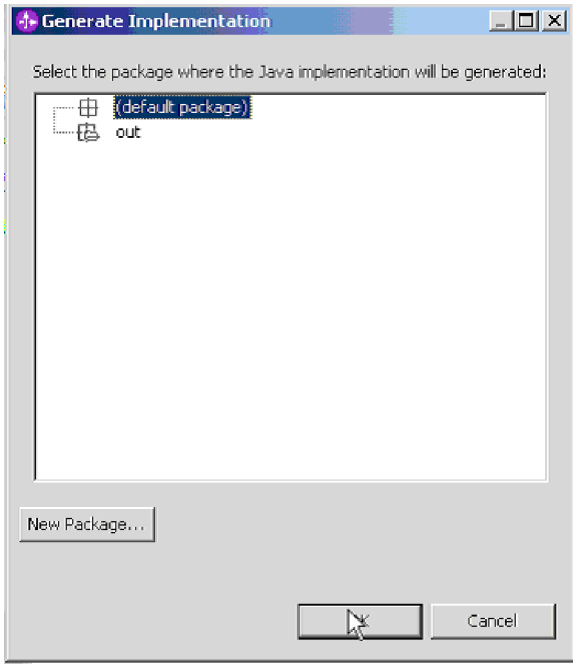


Figure 53. Generating the Java implementation

8. Add **System.out.println** statements in the emit methods of the class file, Component1Impl.java, as shown.

```
public void emitCreateAfterImageBOAccountESPBCAccount(
        DataObject emitCreateAfterImageBOAccountESPBCAccountInput) {
    System.out.println("-------Event generated for 'Create' was polled successfully by adapter and delivered.");
}

/**
 * Method generated to support implemention of operation "emitDeleteAfterImageBOAccountESPBCAccount" defined for
 * named "interface.SiebelInboundInterface".
 *
 * The presence of commonj.sdo.DataObject as the return type and/or as a parameter
 * type conveys that its a complex type. Please refer to the WSDL Definition for more information
 * on the type of input, output and fault(s).
 */
public void emitDeleteAfterImageBOAccountESPBCAccount(
        DataObject emitDeleteAfterImageBOAccountESPBCAccountInput) {
    System.out.println("-------Event generated for 'Delete' was polled successfully by adapter and delivered.");
}

/**
 * Method generated to support implemention of operation "emitUpdateAfterImageBOAccountESPBCAccount" defined for
 * named "interface.SiebelInboundInterface".
 *
 * The presence of commonj.sdo.DataObject as the return type and/or as a parameter
 * type conveys that its a complex type. Please refer to the WSDL Definition for more information
 * on the type of input, output and fault(s).
 */
public void emitUpdateAfterImageBOAccountESPBCAccount(
        DataObject emitUpdateAfterImageBOAccountESPBCAccountInput) {
    System.out.println("-------Event generated for 'Update' was polled successfully by adapter and delivered.");
}
```

Figure 54. System.out.println statement examples

9. When you finish with your changes, save the class file and then the project.

   Note: Select the assembly diagram window before you save the project with the SiebelInboundInterface export. If you do not the select the assembly diagram window, the changes cannot be saved.

Result

The result of running the enterprise service discovery wizard is an service component architecture (SCA) module that contains an enterprise information system (EIS) import.

What to do next

Install the module in the WebSphere Integration Developer integration test client.

Deploying the module for testing

To deploy the module to the application server, add the module to the server view in the WebSphere Integration Developer integration test client. By completing this step, you automatically install and start the module on the server.

Before you begin

You must have a configured project module that is free of build errors.

About this task

Install the SCA module that you created on the WebSphere Integration Developer integration test client to the server using the Servers View in WebSphere Integration Developer.

How to perform this task

1. Right-click on the server, and select **Add and remove projects**.
2. From Available Projects in the left pane, select the project, **Siebel_BO_inboundApp**, and click **Add**. The project is added to the Configured Projects pane on the right.

3. Click **Finish**.

Result

The SCA module has been successfully added to the server.

Note: If you have any problems installing the application using the above approach, you can export the project into an EAR, install and start the project through the WebSphere Process Server administrative console.

What to do next

Test the assembled application.

Testing the assembled adapter application

Use the Siebel application event component to test whether the event was successfully polled and delivered by the adapter.

Before you begin

You should have an service component (SCA) module installed on the server.

About this task

Create a new record in the event component in the Siebel application to test the polling and delivery of an event by the adapter.

How to perform this task

1. Create a new record in the event component in the Siebel application, as shown:

Table 25. Siebel application event components

Event component attribute	Attribute value
Object name	BOAccountU45ESPBCAccountBG
Event type	Create
Status	0
Object key	Id=1-XDF

Note: If you provided any prefix in the connection properties, you must add the prefix to the object name. The provided object key value is a sample only. You must set an appropriate Id in the Object Key by checking the existing Account record in the enterprise information system.

2. In the WebSphere Process Server Admin Console window, select **Applications** → **Enterprise Applications**.
3. Confirm that the application, SiebelBO_inboundApp, has been started successfully. A message is provided at the top of the window and also the status against the application will change from an ″x″ to an arrow.
4. To check the WebSphere Process Server log information, select the **Console** tab at the bottom of the window. Assuming the event was successfully polled and delivered by the adapter, you should see the System.out.println() statement, added earlier in the class, on the console window.

Result

This concludes the successful polling and delivery of an event by the adapter using the business object, Account - ESP. You can also confirm the success by checking the event component using the Siebel client. The corresponding event record will have been deleted from the event component.

Troubleshooting the tutorial

If you encountered the ″EMD, child picklist business component″ error, follow the instructions for the EMD defect workaround. If you were unable to deploy the adapter, as an alternative method, use WebSphere Process Server to deploy the enterprise archive file to the application server.

Exporting the project module as an EAR file

If you are unable to deploy the adapter using WebSphere Integration Developer, use the WebSphere Process Server administrative console to deploy the enterprise archive (EAR) file to the application server. By creating an EAR file, you capture all of the contents of your adapter project in a format can be easily deployed to the application server.

Before you begin

You must have created a configured project that is free of build errors.

About this task

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

How to perform this task

1. Confirm that there are no errors by building the project.
2. Right-click on the project and select **Export → EAR file**.
3. In the EAR Export window, select the EAR project.
4. Provide the absolute path, including the EAR file name, for the Destination. Examples of absolute paths, including the EAR file name, are C:\SiebelBuild\Siebel_BS_OutboundApp.ear and C:\SiebelBuild\Siebel_BO_InboundApp.ear.
5. Select the following options:
   - Export source files
   - Overwrite existing file
   - Include project build paths and metadata files
6. Click **Finish**.

Result

The EAR file now contains an enterprise information system import.

Deploying the EAR using WebSphere Process Server

Use the WebSphere Process Server administrative console to deploy the enterprise archive (EAR) file, as an alternative to using WebSphere Integration Developer.

Before you begin

You must have an adapter project that has been exported to an enterprise archive (EAR) file.

About this task

To deploy the EAR file using the WebSphere Process Server administrative console, use the following procedure.

How to perform this task

1. In the bottom right side window of WebSphere Integration Developer, click **Servers** view.
2. If not already started, right-click on the WebSphere Process Server instance and start the WebSphere Process Server instance.
3. Confirm that the server is up by checking the status, it should be listed as started.
4. Start the administrative console by right-clicking on the server and selecting **Run administrative console**.
5. In the administrative console window, click **login**.
6. Select **Applications** → **Enterprise Applications**.

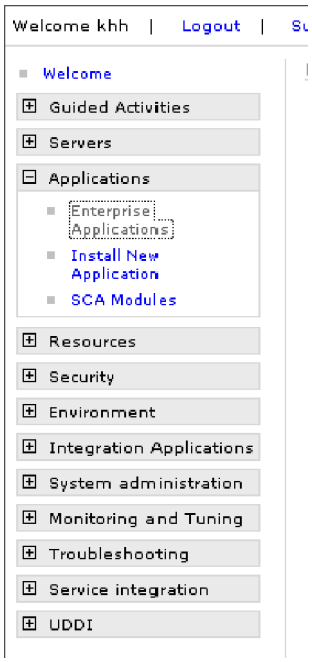


Figure 55. Selecting Enterprise Applications

7. From the Enterprise Applications window, select **Install**.

Figure 56. Selecting the Install option

8. Open the EAR file that you want to deploy:
   a. In the Path to the new application window, choose the system where the EAR file was saved.
   b. Browse to the corresponding folder on the system.
   c. Select the EAR file from the folder where you it saved earlier, click **Open**, and click **Next**.

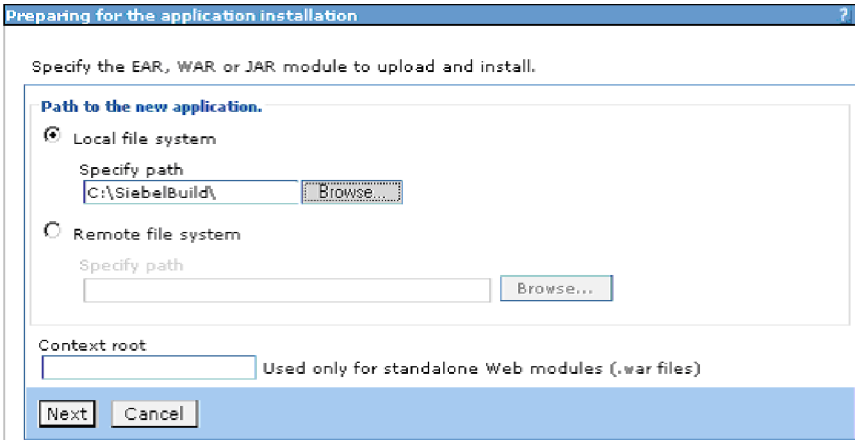


Figure 57. Specifying the module to install

9. From the Preparing the application for installation window, click **Next**.

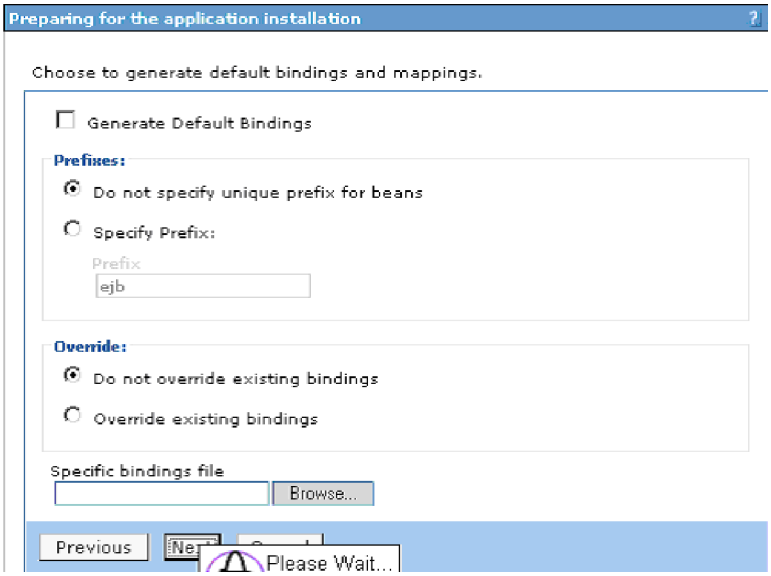


Figure 58. Generating default bindings and mappings

10. From the Install New Application window, click **Step 9 Summary**, then click **Finish**. The project is installed. When installation is complete, a message confirming the successful installation of the application appears.
11. From the same window where the message appears, click **Save to Master Configuration**.
12. From the Enterprise Applications window, click **Save**.

Result

The application is now installed on the server.

What to do next

Start the application.

Starting the application using WebSphere Process Server

Use the WebSphere Process Server administrative console to start the application.

Before you begin

You must have deployed the adapter project through the WebSphere Process Server administrative console.

About this task

Use the WebSphere Process Server administrative console to start the application to make sure that it works properly.

How to perform this task

1. From the Enterprise Applications window, select the check-box next to the installed application, and click **Start**.
2. Confirm that the application has been started successfully. A message is provided at the top of the window and the Status against the application changes from an X to an arrow.

Chapter 12. Reference information

Reference information supports the tasks you want to complete. The information includes all configurable properties for the adapter, messages, and related product information.

Enterprise service discovery connection properties

The enterprise service discovery connection properties include outbound and inbound connection properties required for performing metadata discovery and bidirectional configuration. You configure these properties using the enterprise service discovery wizard when you initially deploy the adapter. When you run the enterprise service discovery wizard in WebSphere Integration Developer, specify the connection properties listed below.

When you run the enterprise service discovery wizard in WebSphere Integration Developer, specify the connection properties listed below.

Enterprise service discovery connection properties for the Adapter for Siebel Business Applications.

Property	Description	Required	Globalized	Default value	Example
Connect string	The connection string needed to connect to the Siebel enterprise information system.	Yes.	No.	None.	Protocol://machinename/enterprise name/object manager /server name Siebel 7.7.x: protocol: //machine name: portnumber /enterprise name/object manager
User name	The user account for the Siebel enterprise information system.	Yes.	No.	None.	user1
Password	The password for the user name.	Yes.	No.	None.	pass1
Language code	The Siebel enterprise information system instance language.	Yes.	No.	None.	ENU

Property	Description	Required	Globalized	Default value	Example
Prefix	The prefix for all business object names.	No.	No.	None.	If the business services option is selected against the Siebel object type, it is used only when the WebSphere Business Integration business object is generated for the outbound service type for the business service method. If the business objects option is selected against the Siebel object type, it is used for all the WebSphere Business Integration business objects generated against business components.
Siebel repository	The name of the Siebel repository.	Yes.	No.	Siebel repository	The list of repository names are provided as choices.
Siebel object type	The type of Siebel object that needs to be listed.	Yes.	No.	Business services	The list has two values from which to choose: business objects and business services.

Note: If the enterprise service discovery wizard is unable to connect to the enterprise information system, due to any problems with the input values or the application server status, an appropriate error is logged in the enterprise service discovery log file.

Adapter configuration properties

The properties in this section should be configured using the enterprise service discovery wizard before deployment or with the WebSphere Application Server administrative console after deployment.

Resource adapter properties

Resource adapter properties consist of logging and tracing, bidirectional language support, and activities specific to the adapter, such as the default configuration properties of the adapter. Configure these properties using WebSphere Process Server administrative console.

When you configure the adapter, specify the resource adapter properties listed below.

Table 26. Resource adapter properties

Property	Type	Description	Default value	Example
BONameSpace	String	Namespace for the business object definitions to be used by this adapter. This value should be taken from the value you provided during the enterprise service discovery process. This property is required.	http://www.ibm.com/xmlns/prod/wbi/j2ca/siebel	None
EIS BiDi format	String	Specifies the BiDI format used by the Siebel EIS to store content business data.	None	Five letters, long string. For more information, see the section, WebSphere Process Server bidirectional language format.
EIS BiDi special format	String	Specifies the name of the category for values subject to special treatment during BiDi transformation invocation.	NORMAL _STRING	WIN_DIR, WIN_NET_DIR, UNIX_DIR, MVS_DIR, URL_WIN_DIR, URL_UNIX_DIR, URL_FTP_HTTP, EMAIL_ FOLDER, JDBC_URL_SQL, SIEBEL _CONNSTR _77, NORMAL _STRING
enableHA Support	None	None	None	None

Table 26. Resource adapter properties (continued)

Property	Type	Description	Default value	Example
eventComponent Name	String	This is the event store name where events are stored for inbound processing.	None	None
eventDelimiter	String	This is the delimiter used between two name value pairs containing the object key name and value resonateSupport (Boolean). This is used to decide if the Siebel Server uses resonate.	None	None
LogFileName	String	The full path of the log file. This property is required. Specify a path like the following: c:\logs\log.txt.	None	None
LogNumber OfFiles	Integer	The number of log files to use. When a log file reaches its maximum size, another log file is started. If no value is specified, the value is set to 1.	None	None
LogMaxFileSize	Integer	Size of the log files in kilobytes. If no value is specified, the files have no maximum size.	None	None
Metadata BiDi format	String	Specifies the BiDi format used by the Siebel EIS to store metadata (for example, business service names).	None	Five letters, long string. For more information, see the section, WebSphere Process Server bidirectional language format.

Table 26. Resource adapter properties (continued)

Property	Type	Description	Default value	Example
ResonateSupport	Boolean	Specifies whether the adapter should use Attach and Detach calls on the Siebel data bean. By default, the ResonateSupport property is set to true.	None	None
Skip BiDi transformation	String	Used to control invocation of BiDi transformation on content data. If true, the transformation is invoked. If false, the transformation is not invoked. An empty string is used to invoke the lookup mechanism.	None	True, false, <empty string>
supportName ValuePair	Boolean	This is used if the event store supports name value pairs for object keys.	None	None
Turn BiDi off	Boolean	If true, BiDi transformation support is turned off. If false, BiDi transformation support is turned on.	True	True, false

Managed (J2C) connection factory properties

Managed connection factory configuration properties are used at run time to create an outbound connection instance with an enterprise information system.

When you configure the adapter, specify the properties listed below.

Note: The ESD wizard refers to these properties as managed connection properties and WebSphere Process Server refers to these as (J2C) connection factory properties.

Table 27. Managed (J2C) connection factory properties

Property	Type	Description	Globalized	Default value	Example
Connect string	N/A	This property determines the Siebel instance information. Siebel: //Gateway Server/ enterprise name/ Object Manager/ Siebel Server	N/A	N/A	N/A
EIS BiDi format	String	Specifies the BiDI format used by the Siebel enterprise information system to store content business data.	N/A	`ILYNN`	Five letters, long string. For more information, see the section, "WebSphere Process Server bi-directional language format."
Language code	N/A	The Siebel instance language. For example, enu for English, or jpn for Japanese.	N/A	N/A	N/A
Metadata BiDi format	String	Specifies the BiDi format used by the Siebel EIS to store metadata (for example, business service names).	N/A	`ILYNN`	Five letters, long string. For more information, see the section, "WebSphere Process Server bi-directional language format."
Password	N/A	The password for the user.	N/A	N/A	N/A

Table 27. Managed (J2C) connection factory properties (continued)

Property	Type	Description	Globalized	Default value	Example
Password BiDi format	String	Specified bi-directional format for the password.	N/A	ILYNN	Five letters, long string. For more information, see the section, "WebSphere Process Server bi-directional language format."
Skip BiDi transforma tion	String	Used to control the invocation of bi-directional trans-formation on the user name. If True, the trans formation is invoked. If False, the transform is not invoked. An empty string is used to invoke the lookup mechanism.	N/A	<empty string>	True, False, <empty string>
User name	N/A	The user name for logging into the Siebel enterprise information system.	N/A	N/A	N/A

Activation specification properties

Activation specification properties hold the inbound event processing configuration information for a message endpoint. They can be set through the enterprise service discovery wizard or the WebSphere Process Server administrative console.

When you configure the adapter, specify the activation specification properties listed below.

Table 28. Activation specification properties

Property	Type	Description	Default value	Example
AutoCreateEDT	Boolean	Flag that indicates whether the adapter should create the EDT table automatically if it does not already exist. The default is *True*.	None	None
Connect string	String	Determines the Siebel instance information. For Siebel 7.5. Siebel: //Gateway Server/ enterprise name/ Object Manager/ Siebel Server. For Siebel 7.7. Siebel: //Gateway Server: portnumber/ enterprise name/ Object Manager	None	None
DeliveryType	String	Either ordered or unordered. This property determines the order in which the events are published. Ordered means one at a time, while unordered means all at once. The default is *ordered*.	None	None
EDT BiDi format	String	Specified BiDi format for EDT properties.	`ILYNN`	Five letters, long string. For more information, see the section "WebSphere Process Server bidirectional language format."
EDT DatabaseName	String	The name of event recovery database.	None	None

Table 28. Activation specification properties (continued)

Property	Type	Description	Default value	Example
EDTDriverName	String	The name of the XA database driver to use to connect to the event distribution table for inbound events. For example: com.ibm.db2j .DB2jXAData Source. If no value is present, the event manager cannot perform recovery.	None	None
EDTSchema Name	String	The schema used to automatically create the event recovery database.	None	None
EDTTableName	String	The name of event recovery table.	None	None
EDTURL	String	The URL to the EDT database.	None	None
EDT URL BiDi special format	String	Specifies the name of the category for values subject to special treatment during bi-directional transformation invocation.	NORMAL _STRING	WIN_DIR, WIN_NET_DIR, UNIX_DIR, MVS_DIR, URL_WIN_DIR, URL_UNIX_DIR, URL_FTP_HTTP, EMAIL_ FOLDER, JDBC_URL_SQL, SIEBEL_ CONNSTR _77, NORMAL _STRING
EDTUserName	String	The user name for connecting to the database.	None	None
EDTUser Password	String	The user password for connecting to the database.	None	None

Table 28. Activation specification properties (continued)

Property	Type	Description	Default value	Example
ESI BiDi format	String	Specifies the bi-directional format used by the Siebel enterprise information system to store content business data.	ILYNN	Five letters, long string. For more information, see the section "WebSphere Process Server bidirectional language format."
EIS BiDi special format	String	Specifies the name of the category for values subject to special treatment during BiDi transformation invocation.	NORMAL _STRING	WIN_DIR, WIN_NET_DIR, UNIX_DIR, MVS_DIR, URL_WIN_DIR, URL_UNIX_DIR, URL_FTP_HTTP, EMAIL_ FOLDER, JDBC_URL_SQL, SIEBEL _CONNSTR_77, NORMAL _STRING
Event component name	String	Specifies the name of the Siebel component for the event table.	None	None
Event component name BiDi format	String	None	ILYNN	Five letters, long string. For more information, see the section "WebSphere Process Server bidirectional language format."
Language code	String	The Siebel instance language.	None	None
Metadata BiDi format	String	None	ILYNN	Five letters, long string. For more information, see the section, "WebSphere Process Server bidirectional language format."
Password	String	The password for the corresponding user name.	None	None

Table 28. Activation specification properties (continued)

Property	Type	Description	Default value	Example
Password BiDi format	String	None	`ILYNN`	Five letters, long string. For more information, see the section "WebSphere Process Server bidirectional language format."
PollPeriod	Integer that is equal to or greater than 0.	The rate in milliseconds at which to poll the EIS event store for new inbound events. If *0*, the adapter will not wait between cycles. The poll cycle is established at a fixed rate, meaning that is an execution of the poll cycle is delayed (for example, the prior poll cycle takes longer than expected to complete the next cycle will occur immediately to catch up.) This is a required property. The default is *500*.	None	None
PollQuantity	Integer that is greater than 0.	This property is used to determine the number of events to deliver to each endpoint per poll cycle. It is a required property.	None	None
RetryInterval	Integer	Used to reestablish the connection for inbound delivery.	None	None
Skip BiDi transformation	String	None	`<empty string>`	`True`, `False`, `<empty string>`

Table 28. Activation specification properties (continued)

Property	Type	Description	Default value	Example
Skip BiDi transformation for connection string	String	None	`<empty string>`	`True`, `False`, `<empty string>`
Skip BiDi transformation for EDT	String`True`, `False`, `<empty string>`	None	`<empty string>`	True, False, `<empty string>`
Skip BiDi transformation for EDT URL	String	None	`<empty string>`	`True`, `False`, `<empty string>`
Skip BiDi transformation for event component name	String	None	`<empty string>`	True, False, `<empty string>`
Skip BiDi transformation for password	String	None	`<empty string>`	True, False, `<empty string>`
Skip BiDi transformation for user name	String	Used to control invocation of BiDi transformation on the user name. If *True*, the transformation is invoked. If *False* the transformation is not invoked. An empty string is used for the invocation of the lookup mechanism	`<empty string>`	True, False, `<empty string>`
User name	String	The user name to log into the Siebel application.	None	None
User name BiDi format	String	Specified BiDi format for the user name.	`ILYNN`	Five letters, long string. For more information, see the section ″WebSphere Process Server bidirectional language format.″

Settings for controlling bidirectional transformation

Within each category of adapter properties, certain properties can be set to control bidirectional transformation of content or metadata. Properties controlling bidirectional transformation can be set for the resource adapter, the managed connection factory, and the activation specification; data transformation properties can also be set to control bidirectional transformation.

Resource adapter properties

The following resource adapter properties can be set to control .

- EIS BiDi format
- EIS BiDi special format
- Metadata BiDi Format
- Skip BiDi transformation
- Turn BiDi off

Managed (J2C) connection factory properties

The following managed (J2C) connection properties can be set to control bidirectional transformation.

- EIS BiDi Format
- Metadata BiDi Format
- Password BiDi Format
- Skip BiDi Transformation

Activation specification properties

The following activation specification properties can be set to control bidirectional transformation.

- Edit BiDi Format
- Edit URL BiDi special format
- ESD BiDi format
- EIS BiDi Special Format
- Event component name BiDi Format
- Metadata BiDi format
- Password BiDi format
- Skip BiDi transformation
- Skip BiDi transformation for connection string
- Skip BiDi transformation for EDT
- Skip BiDi transformation for EDT URL
- Skip BiDi transformation for event component name
- Skip BiDi transformation for password
- Skip BiDi transformation for user name
- User name BiDi format

Adding jar files to WebSphere Integration Developer versions 6.0.1.1 and earlier

If you are using WebSphere Integration Developer version 6.0.1.1 or earlier, you must manually add three jar files to the classpath of the connector project.

You must have installed the adapter and all of the adapter prerequisites before the jar files can be added to the connector project in WebSphere Integration Developer.

1. Open WebSphere Integration Developer.
2. In J2EE perspective, right-click the connector project and select **Properties**.
3. Select **Java Build Path** and click **Add External Jars**.
4. Select your WebSphere Process Server or Enterprise Server Bus Install/lib folder and select ffdcSupport.jar, aspectjrt.jar and icu4j_3_2.jar.
5. Click **Open** and then **OK**.

Messages

The messages issued by IBM WebSphere Adapters are documented in the WebSphere Adapters, version 6.0.2 information center.

You can view the adapter messages at the following link: WebSphere Adapters messages..

Related product information

The following links, information centers, Redbooks, and Web pages contain related information for the IBM WebSphere Adapter for Siebel Business Applications.

Additional information you might need

Table 29. WebSphere Adapters information you might need

Information	How to find it
How to edit business objects using the Business Object Editor	In the IBM WebSphere Business Process Management information center, which includes documentation for WebSphere Integration Developer, search for the topic, "Editing Business Objects."
How to uninstall a deployed adapter	On the WebSphere Application Server library page, open the information center for your version of WebSphere Application Server and search for the topic, "Uninstalling applications."

Information for related products

- WebSphere Adapters, Version 6.0
- WebSphere Business Integration Adapters
- WebSphere Integration Developer
- WebSphere Process Server
- WebSphere Enterprise Service Bus
- WebSphere Application Server

Redbooks

- WebSphere Adapter Development Redbook
- WebSphere Redbooks domain

developerWorks® resources

- WebSphere Adapter Toolkit
- WebSphere business integration zone

Support and assistance

- WebSphere Adapters product support
- WebSphere Adapters technotes - in the **Additional search terms** field, specify the name of the adapter and click **Go**.

Notices

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

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

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

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

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

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

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

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

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

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

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

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

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

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

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

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, 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: © (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © 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 and related trademarks: http://www.ibm.com/legal/copytrade.shtml

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

K

L

M

O

P

Q

R

S

T

U

V

W

Printed in USA