WebSphere® Adapters

IBM

Version 6.0.2

Adapter for SAP Software User Guide

Note
Before using this information, be sure to read the general information in "Notices" on page 269.

22December2006

This edition applies to version 6, release 0, modification 2 of WebSphere Adapter for SAP Software (product number 5724-L79) 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 SAP Software. 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 SAP Software in a solution that requires data exchange between an enterprise information system (EIS) and Java™ Platform, Enterprise Edition (J2EE) applications. To use it, you should understand and have experience with the following concepts, standards, and tools:

- The SAP interfaces–BAPI, ALE, and SQI–that are used with the adapter.
- The business solution and environment.
- Databases, data access issues, transactional models, and connections across heterogeneous relational databases, queues, and Web services.
- Business integration mechanisms, including the Service Component Architecture (SCA) programming model and the Service Data Object (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 testing and deployment
- Resolve interdependencies between entities such as enterprise beans, workflows, and Web pages
- Write procedures to use database access logic efficiently
- Build data models for external data access tools
- Implement security measures

Chapter 2. What's new

WebSphere Adapter for SAP Software, version 6.0.2 provides enhancements to version 6.0 of the adapter.

New in this release

WebSphere Adapter for SAP Software, version 6.0.2 includes enhancements to the BAPI and ALE interfaces and provides a new SAP query interface (SQI), with which you can retrieve data from SAP database tables through the SAP application layer.

New in version 6.0.2:

- The BAPI interface supports nested structures.
- The ALE interface contains the following enhancements:
  - Support for multiple endpoints for inbound processing
  - Message handling without IDoc conversion
  - Detection and correction of RFC server connection failures
  - IDoc packet processing outbound from SAP
  - Transaction ID processing inbound to SAP
- The new SQI interface is available.
- Event recovery has been modified. You now set up the data source in the administrative console.
- Error codes can be defined to validate data returned from RFC.
- High availability support is available for inbound processing. For more information, see ″WebSphere Adapters in clustered environments″.
- Six tutorials, which guide you through the process of creating an adapter project, generating business objects, deploying a module, and testing the module, have been added. These tutorials are self-contained, and each one can be completed in under an hour. These tutorials take the place of the samples that were documented in previous versions of the user guide.

Release notes

The release notes for WebSphere Adapter for SAP Software, version 6.0.2 summarize new features and functions in this release and document any known issues and workarounds.

The release notes are located at the following Web site: Adapter for SAP Software Release Notes.

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 are illustrated in the following figure.

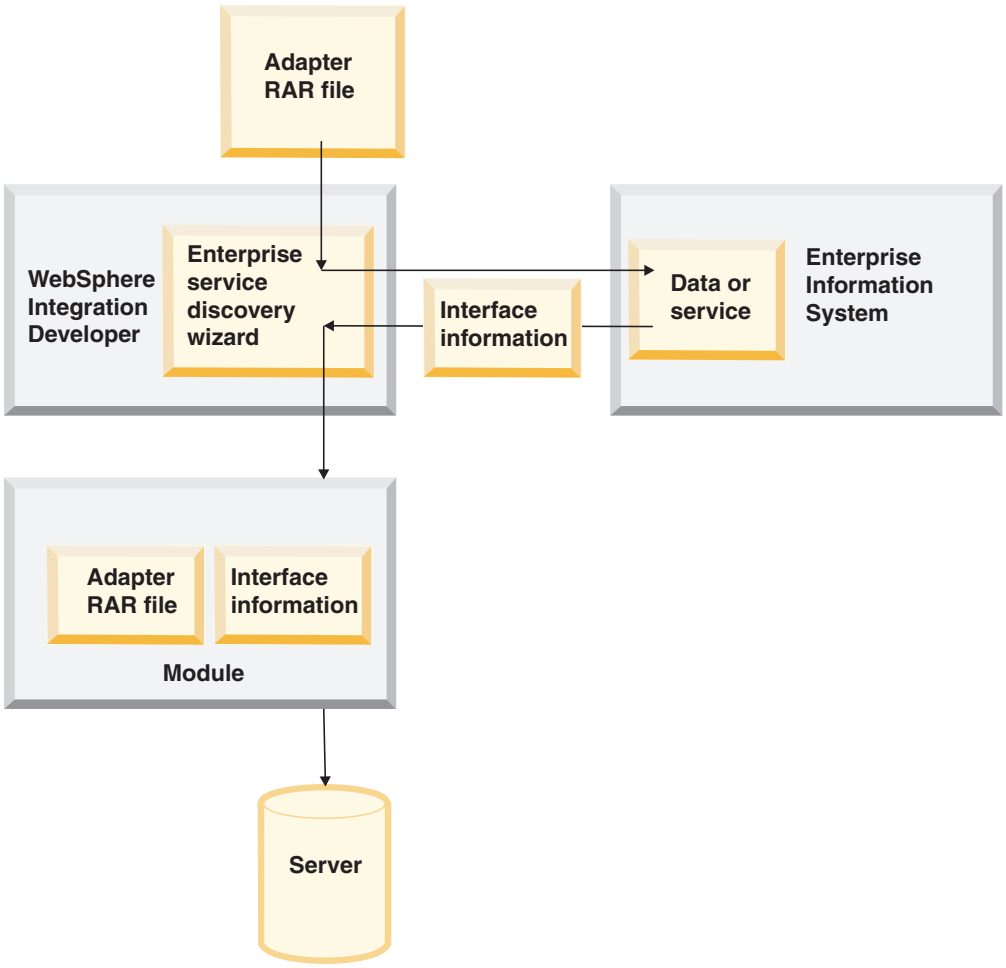


Figure 1. How a module is generated and deployed

Chapter 4. Introduction to WebSphere Adapter for SAP Software

WebSphere Adapter for SAP Software provides the means for clients (J2EE components) to send requests to the SAP server (for example, to update a customer record in an SAP table or to query data from that table) or to receive events from the server (for example, to be notified that a customer record has been updated).

Adapter for SAP Software provides three interfaces to interact with an SAP server:

- Business Applications Programming Interface (BAPI)

  A BAPI is an SAP-standardized application programming interface that enables third-party systems to interact with an SAP server. The adapter models SAP BAPI function calls as business objects. These function calls create, update, or retrieve data on an SAP server. The BAPI interface of the adapter is used for outbound processing only. The client can call the SAP server through the BAPI interface.
- Application Link Enabling (ALE)

  ALE integrates business processes between SAP systems and external applications as well as between SAP systems. ALE uses IDocs (SAP Intermediate Document structures) for data exchange. The data is exchanged asynchronously. The ALE interface is used for both outbound and inbound processing. The client can call the SAP server through the ALE interface and receive events from it.
- SAP query interface (SQI)

  The SAP query interface retrieves data from SAP application tables. The tables are modeled as hierarchical business objects, which are generated during the enterprise service discovery process. The SAP query interface is used for outbound processing only. The client can call the SAP server through the SAP query interface.

The following figure shows how clients use outbound interfaces to send requests to SAP applications. The adapter provides request and response interfaces for BAPI and SQI. The request originates at the client, and the request is returned by the SAP application. For the ALE interface, the outbound request is one-way. The figure also shows how the inbound interface for ALE originates at the SAP application and is sent to one or more endpoints or to an event-recovery disk.

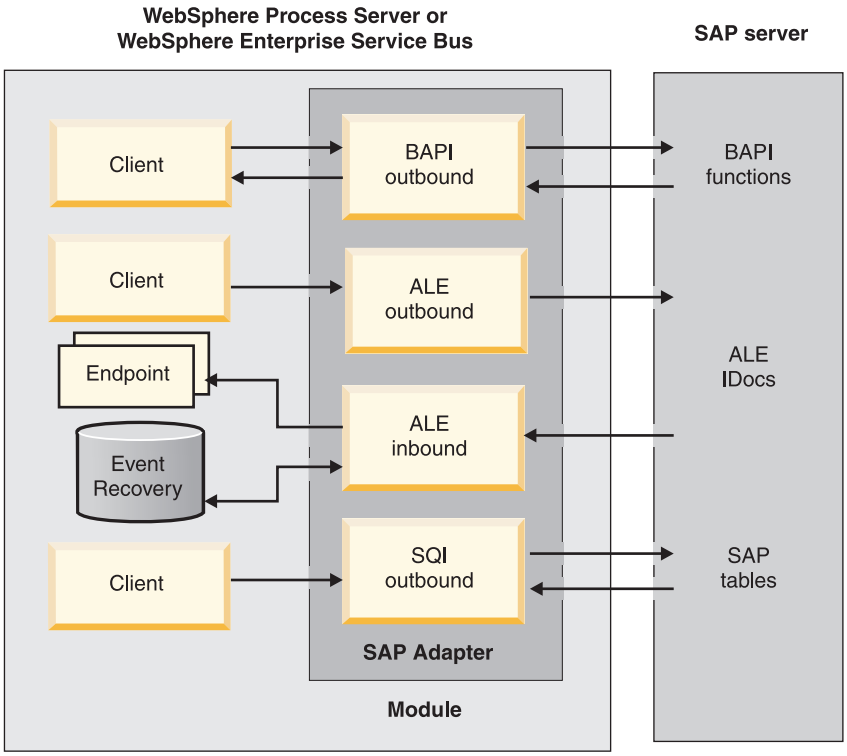


Figure 2. SAP inbound and outbound interfaces

Hardware and software requirements

Before installing Adapter for SAP Software, 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 SAP Software, 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 Adapter for SAP Software

IBM WebSphere Adapter for SAP Software connects to SAP systems running on SAP Web application servers. The adapter supports the Business Application Programming Interface (BAPI) for outbound processing, Application Link Enabling (ALE) for both outbound and inbound processing, and the SAP Query Interface (SQI) for outbound processing. You set up the adapter to perform outbound and inbound processing by using enterprise service discovery to generate business objects based on the services it discovers on the SAP server.

The adapter uses the SAP Java Connector (SAP JCo) API to communicate with SAP applications, as shown in the following illustration.

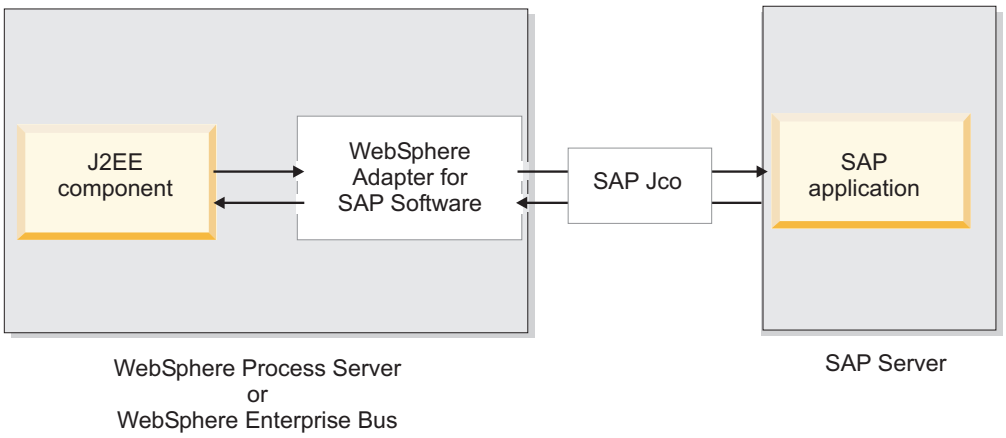


Figure 3. How the adapter connects J2EE components with SAP applications

Adapter for SAP Software supports WebSphere Application Server-managed deployment only. The adapter does not support a non-managed environment.

WebSphere Adapter for SAP Software is packaged and delivered as two RAR files:

- CWYAP_SAPAdapter.rar, which does not support J2C local transactions
- CWYAP_SAPAdapter_Tx.rar, which supports J2C local transactions

  J2C local transactions are managed internally and involve no external transaction managers. The adapter supports J2C local transactions for the BAPI interface and for the ALE outbound interface.

Enterprise service discovery

The enterprise service discovery wizard is a tool you use to configure your adapter before it is deployed to WebSphere Process Server or WebSphere Enterprise Service Bus. Enterprise service discovery connects to the SAP server, discover services (based on search criteria you provide), and generates business objects and interfaces.

The enterprise service discovery for Adapter for SAP Software imports SAP metadata definitions and other artifacts related to the BAPI, ALE, and SAP query interfaces.

Using WebSphere Integration Developer, you establish a connection to the SAP system for purposes of browsing the metadata repository. You specify connection information (such as the user name and password needed to access the server, as shown in the following figure), and you select the interface you want to use (for example, BAPI).

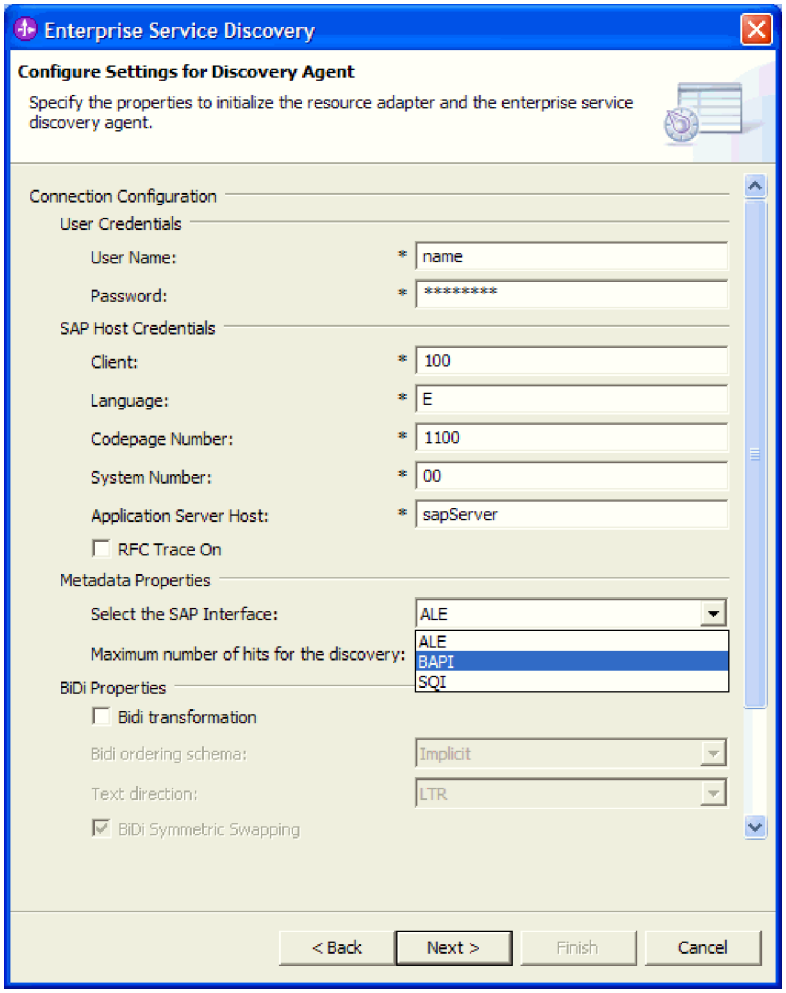


Figure 4. Configure Settings for Discovery Agent window

The service metadata associated with that interface is displayed. You can then browse the information and select the artifacts of interest (for example, you can list all BAPIs that begin with "CUSTOMER" and then select one or more BAPIs).

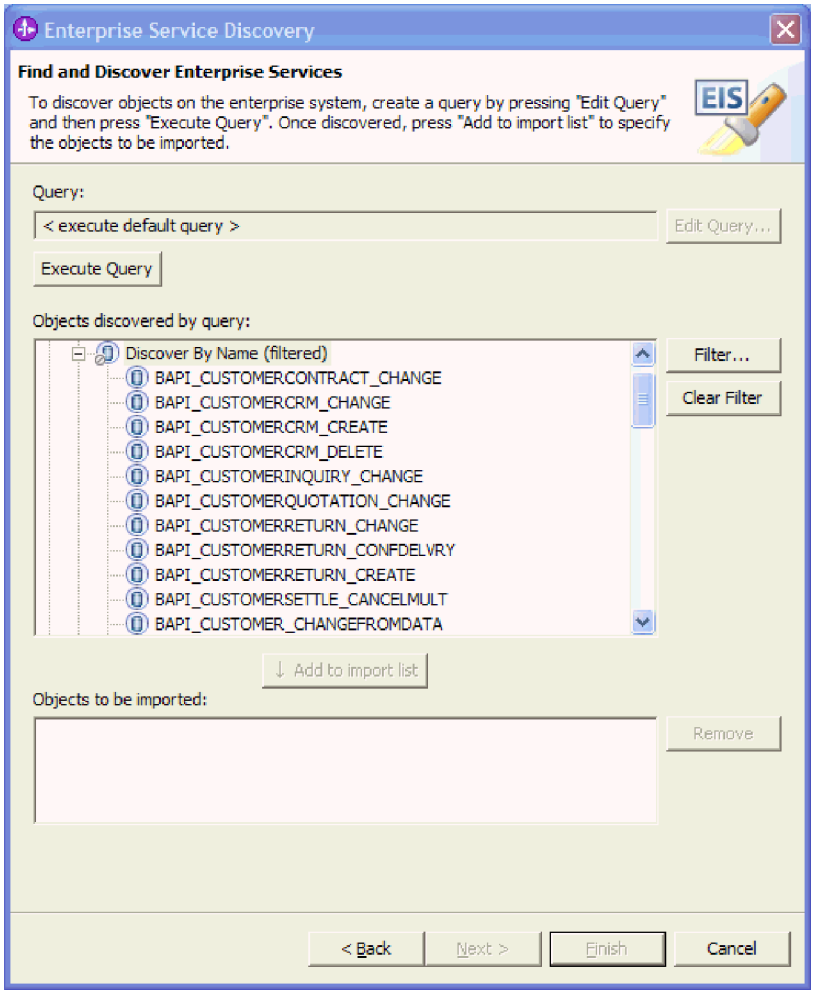


Figure 5. Find and Discover Enterprise Services window

The result of running enterprise service discovery is a module that contains the interfaces and business objects along with the adapter. You deploy this module on WebSphere Process Server or WebSphere Enterprise Service Bus.

For example, if you ran enterprise service discovery and selected BAPI_CUSTOMERGETLIST, you would see the following artifacts listed under the module name (in this example, CustomerList).

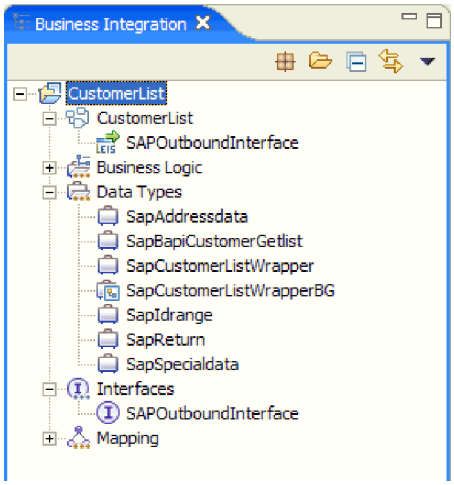


Figure 6. An example of the module generated by the enterprise service discovery wizard

Client applications use the interface information generated by enterprise service discovery to make calls to the SAP server.

The BAPI interface

BAPIs are SAP-standardized business application programming interfaces that enable third-party systems to interact with SAP systems. The adapter models SAP BAPI function calls as business objects. These function calls create, update, or retrieve data on an SAP system. The BAPI interface is used for outbound processing only.

Features of the BAPI interface

The adapter supports simple BAPIs as well as BAPI transactions managed by the application server. The adapter also supports the use of local (JCA) transactions for calling BAPIs.

Simple BAPIs

A simple BAPI call is a synchronous blocking call. A client can invoke a BAPI to perform a single operation, such as retrieving a list of customers. The adapter supports simple BAPI calls by representing each with a single business object schema.

Note: The BAPI interface supports after-image updates only. An after-image update is a complete snapshot of an object with information about whether the object was created, updated, or deleted but without the specifics of exactly what changed.

Make sure you install the CWYAP_SAPAdapter.rar file to work with simple BAPIs.

BAPI transactions

A BAPI transaction, also referred to as a BAPI logical unit of work, consists of a set of BAPIs that are executed in sequence to complete the entire transaction.

For example, to update an employee record in the SAP system, the record needs to be locked before being updated. This is accomplished by calling three BAPIs, in sequence, in the same transaction. The following three BAPIs illustrate the kind of sequence that forms such a transaction:

- BAPI_ADDRESSEMP_REQUEST
- BAPI_ADDRESSEMP_CHANGE
- BAPI_ADDRESSEMP_APPROVE

The first BAPI in the transaction locks the employee record, the second updates the record, and the third approves the update. The advantage of using a transaction is that the client can invoke the employee record change with a single call, rather than with three separate calls. In addition, if SAP requires that the BAPIs execute in a specific sequence for the business flow to complete correctly, the transaction supports this sequence.

Make sure you install the CWYAP_SAPAdapter.rar file to work with BAPI transactions in which the transaction is managed within the application server.

J2C local transactions for BAPIs

The adapter supports calling BAPIs from within a J2C local transaction. Local transactions are managed internally and involve no external transaction managers.

You can invoke multiple BAPIs as part of a J2C local transaction before issuing a COMMIT operation.

Use CWYAP_SAPAdapter_Tx.rar file if you are calling BAPIs from J2C local transactions.

Outbound processing

The adapter uses the BAPI interface to request BAPI functions or transactions. The request is sent to the SAP system, and the response is returned.

The following steps describe how the adapter supports outbound processing for the BAPI interface.

Note: The client application that makes the BAPI call uses the interface information that was generated by the enterprise service discovery wizard.

1. The adapter receives a request, which includes a BAPI business object, from a client application.
2. The adapter converts the BAPI business object to an SAP JCo function call.
3. The adapter uses the Remote Function Call (RFC) interface to execute the BAPI/RFC function call in the SAP application.
4. After passing the data to SAP, the adapter handles the response from SAP and converts it back into the business object.
5. The business object is then sent to the calling component (the client application).

BAPI transactions

The adapter does not provide an automated rollback mechanism for BAPI transactions. Rollback of a BAPI transaction can be achieved in one of the following ways:

- Do not put explicit COMMITs in the application-specific information sequence. When an error occurs in one of the BAPIs, the sequence of BAPI calls is terminated and BAPI_TRANSACTION_ROLLBACK is called. If there is no intrinsic COMMIT in any of the BAPIs already called, no further steps are required. Most BAPIs do not have an intrinsic COMMIT.
- Call another BAPI that can compensate for the work that is already committed, as in the case of the BAPIs that have an intrinsic COMMIT.

Local transactions

The adapter supports J2C local transactions for the BAPI interface.

Business objects

The adapter uses a business object to represent each BAPI it calls from the SAP system. The structure of the business object varies depending on whether the BAPI is a simple BAPI or a BAPI transaction.

The adapter depends on the BAPI metadata that is generated by the enterprise service discovery wizard to construct the business objects. This metadata contains BAPI-related information such as the operation of the business object, import parameters, export parameters, table parameters, transaction information, and dependent or grouped BAPIs.

Business object structure for a simple BAPI:

A business object for a simple BAPI call reflects a BAPI method or function call in SAP. Each business object property maps to a BAPI parameter. The metadata of each business-object property indicates the corresponding BAPI parameter. The operation metadata determines the correct BAPI to call.

For a simple BAPI that performs Create, UpdateWithDelete, Retrieve, and Delete operations, each operation is represented by a business object, with the business objects being grouped together within a wrapper that is contained in the business graph.

Note: The object definition can be associated with multiple operations, but only one operation is executed at run time. Each business object is a child of the wrapper and represents a complex property of the wrapper object.

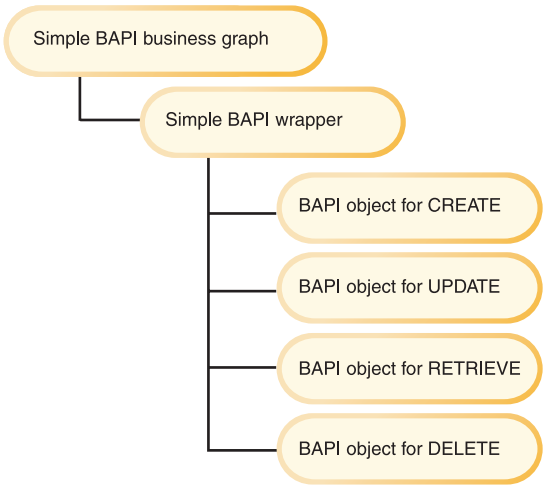


Figure 7. Simple BAPI business structure

The BAPI business objects are children of the business object wrapper, and, depending on the operation to be performed, only one child object in this wrapper needs to be populated at run time in order to execute the simple BAPI call. Only one BAPI, the one that is associated with the operation to be performed, is called at a time.

The following figure shows an example of a BAPI business object graph, which contains a verb and the BAPI business object wrapper.

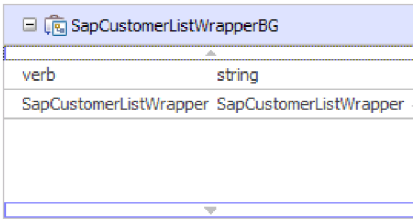


Figure 8. Example of a BAPI business object graph

An example of a BAPI business object wrapper is shown in the following figure. The wrapper contains a BAPI business object.

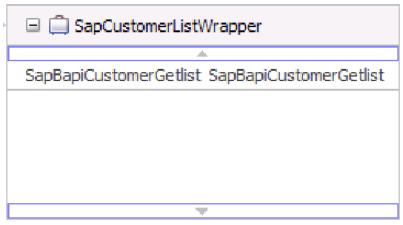


Figure 9. Example of a BAPI wrapper business object

The following figure shows an example of the BAPI business object. This object represents the CustomerGetList BAPI.

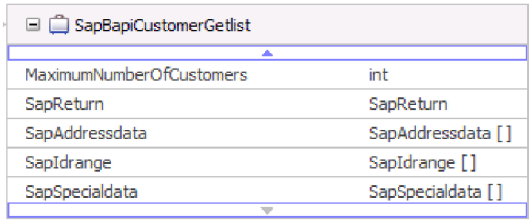


Figure 10. Example of a simple BAPI business object

Business object structure for a nested BAPI:

A nested BAPI business object contains structure parameters that can contain one or more other structures as components.

The following illustration shows the structure of a nested BAPI.

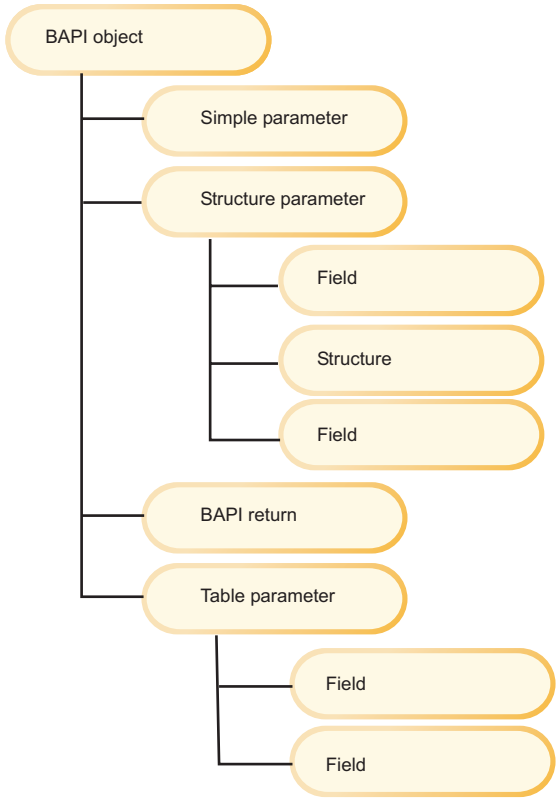


Figure 11. Structure of a nested BAPI

The following figure shows an example of the BAPI business object that contains both simple parameters (for example, LanguageOfTheTexts) and structure parameters (for example, SapLinesDescr).

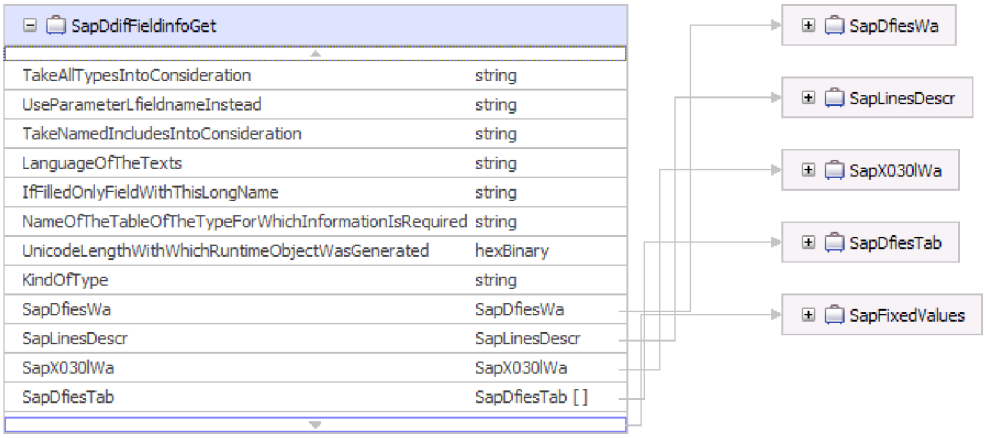


Figure 12. The SapDdifFieldingoGet business object

The SapLinesDescr business object contains simple parameters and a business object.

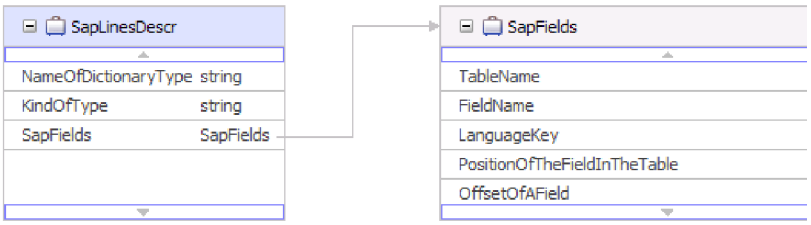


Figure 13. The SapLinesDescr business object

Business object structure for a BAPI transaction:

A business object representing a BAPI transaction is a wrapper object that contains multiple BAPI objects as children. Each individual child BAPI object within the wrapper BAPI transaction object represents a BAPI.

A BAPI transaction business graph consists of a verb and the top-level business object that represents the transaction. An example of a BAPI transaction business graph is shown in the following figure.

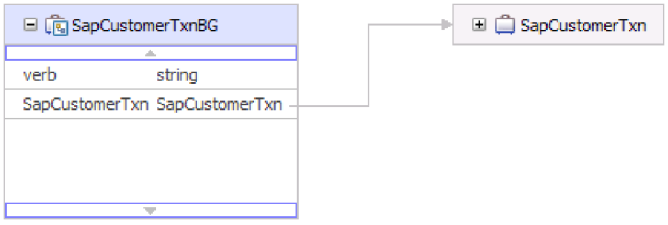


Figure 14. Example of a BAPI transaction business graph

The adapter supports a BAPI transaction using a top-level wrapper business object that consists of multiple child BAPIs, each one representing a simple BAPI in the sequence. The BAPI transaction wrapper object represents the complete transaction.

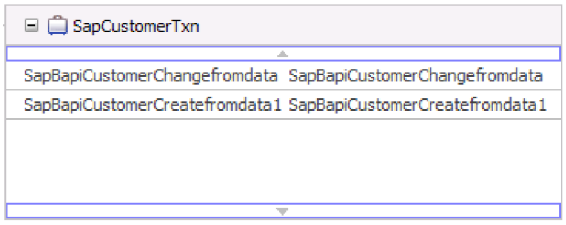


Figure 15. Example of a top-level wrapper object for a BAPI transaction

The adapter uses the sequence of operations in the operation metadata to execute the BAPIs in the transaction. In other words, it does not use the order of the child business objects within the business graph.

Each second-level child business object represents a structure parameter or table parameter of the method. Simple attributes correspond to simple parameters of the method. The following figure shows the structure.

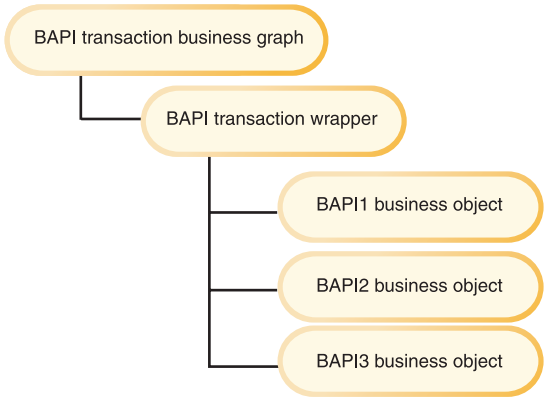


Figure 16. Structure of BAPI transaction objects

Child business object structure:

Each BAPI business object (child business object of the wrapper) represents a BAPI call. The business object properties correspond to the parameters of the BAPI call in SAP. These properties are defined by metadata in the business object.

The adapter supports both single- and multiple-cardinality relationships between business objects. A business object based on a BAPI can contain no more than two levels of hierarchy, unless it is a nested BAPI. Therefore, all BAPI simple parameters correspond to attributes of the top-level business object, and BAPI structure and table parameters correspond to child business objects that contain attributes only.

The following table describes the mapping of the business object structure to the actual BAPI call in SAP.

Table 1. Mapping of BAPIs to business objects

BAPI interface parameter	SAP adapter business object property
Simple parameter	A property of the business object. This is a simple property.

Table 1. Mapping of BAPIs to business objects (continued)

BAPI interface parameter	SAP adapter business object property
Structure parameter	Single-cardinality child business object. This is a complex property because it is two-dimensional.
Property	Maps to a field in the business object.
BAPI return	Contains a structure or a table parameter.
Table parameter	Multiple-cardinality child business object. This property is three-dimensional.

Import and export parameters can be simple field or structure parameters.

The ALE interface

The SAP ALE (Application Link Enabling) interface is part of the integration layer within the SAP Business Framework Architecture (BFA). BFA is a component-based architecture that enables business process integration and asynchronous data communication between two or more SAP systems or between SAP and external systems. Application systems are loosely coupled in an ALE integrated system and the data is exchanged asynchronously.

The adapter interacts with the ALE interface to support outbound and inbound processing by enabling the exchange of data in the form of business objects. The data exchange includes the following activities:

- SAP Intermediate Document (IDoc) exchange for inbound and outbound events.
  - The IDocs can be exchanged either as individual documents or in packets.
  - From the SAP application, IDocs can be sent as parsed or unparsed documents.
- Transaction ID (TID) management for inbound and outbound events. The adapter uses tRFC (transactional RFC) for inbound and outbound events to guarantee delivery and to ensure that each IDoc is exchanged only once with SAP.

For inbound processing, the adapter is able to listen to and deliver events from multiple SAP systems.

Adapter for SAP Software is also able to deliver events to multiple SCA endpoints. You enable delivery to multiple endpoints by configuring multiple activation specifications.

- If the endpoints subscribe to the same events from the same SAP system, all properties in the individual activation specifications must be identical.
- Endpoints that subscribe to different activation specifications receive events that match the criteria for the activation specification.

Define a separate activation specification for each endpoint to which events need to be delivered, except when the adapter delivers events only to those endpoints that are active.

ALE interface prerequisites

The SAP server must be properly configured to process business objects.

Before running the adapter with the ALE interface, verify that the SAP system is properly configured to process business objects. The following conditions apply to both inbound and outbound processing:

- Check that the logical systems are defined and assigned for the SAP system and external system (SAP transaction code SALE).
- Check that the distribution model has been maintained and that the required message types have been added to the model (SAP transaction code BD64).
- Check that there are partner profiles for the logical system or distribution model (SAP transaction code WE20).
- Check that the port definition (SAP transaction code WE21) is defined for the version of IDoc record types that you want.

Refer to the SAP documentation for information about the SAP transaction codes.

Outbound processing

The adapter includes an ALE interface so that clients can send outbound requests. The adapter supports ALE outbound event processing by working with business objects that represent outbound IDocs.

The following steps describe how the adapter supports outbound processing for the ALE interface.

Note: The client application that makes the request uses the interface information that was generated by the enterprise service discovery wizard.

1. The adapter receives a request, which includes an IDoc business object, from a client application.
2. The adapter uses the IDoc business object to populate the appropriate RFC-enabled function call used by the ALE interface.
3. The adapter establishes an RFC connection to the ALE interface and passes the IDoc data to the SAP system.
4. After passing the data to SAP, the adapter performs one of the following steps:
   - If the call is not managed by a J2C local transaction, the adapter releases the connection to SAP and does not return any data to the caller. When no exceptions are raised, the outbound transaction is considered successful. You can verify whether the data is incorporated into the SAP application by inspecting the IDocs that have been generated in SAP.
   - If the call is managed by a J2C local transaction, the adapter returns the transaction ID.

     The adapter uses the tRFC protocol to support J2C local transactions.

Inbound processing

The adapter supports inbound processing (from the SAP system to the adapter) for the ALE interface only. The adapter can process events as individual IDocs or as an IDoc packet. Additionally, the IDoc can be sent in a parsed format or it can be sent directly (without conversion).

Inbound event processing consists of the following steps:

1. The adapter generates event listeners to the SAP system.
2. Whenever an event occurs in SAP, the event is sent to the adapter by way of the event listeners.
3. The adapter converts the event into a business object before sending it to the endpoint.

The adapter uses the event recovery mechanism to track and recover events in case of abrupt termination. The event recovery mechanism uses a data source for persisting the event state.

Event error handling:

WebSphere Adapter for SAP Software provides error handling for inbound ALE events by logging the errors and attempting to restart the event listener.

When the adapter detects an error condition, it performs the following actions:

1. The adapter logs the error information in the event log or trace file.
2. The adapter attempts to restart the existing event listeners.

   The adapter uses the activation specification values for RetryLimit and RetryInterval.

   - If the SAP application is not active, the adapter attempts to restart the listeners for the number of times configured in the RetryLimit property.
   - The adapter waits for the time specified in the RetryInterval parameter before attempting to restart the event listeners.
3. If the attempt to restart the event listeners fails, the adapter performs the following actions:
   - The adapter logs the error condition in the event log or trace file.
   - The adapter cleans up the existing ALE event listeners.
   - The adapter starts new event listeners.

   Note: The adapter uses the values of the RetryLimit and RetryInterval properties when starting the new event listeners.
4. If all the retry attempts fail, the adapter logs the relevant message and CEI events and stops trying to recover the ALE event listener. The adapter or SCA application would need to be restarted manually.

Event recovery:

The adapter supports event recovery for ALE inbound processing in case of abrupt termination. During event processing, the adapter persists the event state in an event recovery table that resides on the data source. You must set up this data source before you can create the event recovery table.

Data source

ALE inbound scenarios require a JDBC data source to be configured. The data source is used for event tracking and recovery.

You set up the data source in the administrative console of WebSphere Process Server or WebSphere Enterprise Service Bus. You select a JDBC provider (for example, Cloudscape™ 5.1) and then create a new data source.

Event recovery table

The EP_CreateTable configuration property in the J2C activation specification properties determines whether the event recovery table is created automatically. The default value of this property is `True` (create the table automatically). To create the table manually, use the information provided in the following table.

Table 2. Event recovery table fields

Table field name	Type	Description
EVNTID	VARCHAR(255)	Transaction ID for the tRFC (Transactional Remote Function Call) protocol. The tRFC protocol significantly improves the reliability of the data transfer, but it does not ensure that the order of ALE transactions specified in the application is observed. Event ordering is also affected by the number of event listeners. However, at some point all ALE transactions are transferred.
EVNTSTAT	INTEGER	Event processing status. Possible values are: • 0 (Created) • 1 (Executed) • 3 (In Progress) • -1 (Rollback)
XID	VARCHAR(255)	An XA resource keeps track of transaction IDs (XIDs) in the event recovery table. The adapter queries and updates that XID field. During recovery, WebSphere Application Server calls the resource adapter, querying it for XA resources, and then does transaction recovery on them. **Note:** The XA resource is used to enable assured once delivery. Make sure the activation specification property Assured Once Delivery is set to `true`.
BQTOTAL	INTEGER	Total number of IDocs in the packet.
BQPROC	INTEGER	Sequence number of the IDoc in the packet that the adapter is currently processing.
EVNTDATA	VARCHAR(255)	Not used.

Configure a separate event recovery table for each endpoint. The same data source can be used to hold all of the event recovery tables.

Event processing for a single, parsed IDoc:

An IDoc corresponds to a single business object. The adapter can process an ALE event that contains just one IDoc.

The adapter processes an inbound event for a single IDoc according to the following procedure:

1. When the SAP system sends a transaction ID to the adapter, the adapter checks the status of the event and takes the following actions:
   - If this is a new event, the adapter stores an EVNTID (which corresponds to the transaction ID) along with a status of `0` (Created) in the event recovery table.
   - If the event status is `-1` (Rollback), the adapter updates the status to `0` (Created).
   - If the event status is `1` (Executed), the adapter returns an indication of success to the SAP system.

2. The SAP system sends the single IDoc to the adapter, which parses and converts the IDoc to a business object and stores it in memory.
3. The SAP system sends a COMMIT call to the adapter.
4. The adapter sends the business object to the appropriate message endpoint. The adapter can deliver objects to endpoints that support transactions as well as to endpoints that do not support transactions.
   - For endpoints that support transactions, the adapter delivers the business object as part of a unique XA transaction (a two-phase commit transaction) controlled by the application server. When the endpoint receives the event and the transaction is committed, the status of the event is updated to 1 (Executed). The message endpoint must be configured to support XA transactions.
   - For endpoints that do not support transactions, the adapter delivers the business object to the endpoint and updates the status of the event to 1 (Executed). The adapter delivers the business object without the quality of service (QOS) that guarantees once only delivery.
5. If an exception occurs while the adapter is processing the event or if the endpoint raises an exception, the event status is updated to -1 (Rollback).
6. If no exception occurs, the SAP system performs the following tasks:
   - It sends a COMMIT call to the adapter, and the event status is updated to 1 (Executed).
   - It sends a CONFIRM call to the adapter.

   The adapter then deletes the records with a status of 1 (Executed) and logs a common event infrastructure (CEI) event that can be used for tracking and auditing purposes.

Event processing for parsed IDoc packets:

An inbound event can contain multiple IDocs, with each IDoc corresponding to a single business object. The multiple IDocs are sent by the SAP system to the adapter in the form of an IDoc packet. The packet can be split or non-split.

Event processing for split, parsed IDoc packets:

A split IDoc packet contains multiple IDocs, with each IDoc corresponding to a single business object.

The adapter processes an inbound event for an IDoc packet that contains multiple individual IDocs according to the following procedures:

1. When the SAP system sends the transaction ID to the adapter, the adapter checks the status of the event and takes one of the following actions:
   - If this is a new event, the adapter stores an EVNTID (which corresponds to the transaction ID) along with a status of 0 (Created) in the event recovery table.
   - If the event status is -1 (Rollback), the adapter updates the status to 0 (Created).
   - If the event status is 1 (Executed), the adapter returns an indication of success to the SAP system.
2. The SAP system sends the IDoc packet to the adapter, which parses it and converts the IDoc into multiple business objects and stores them in memory.

3. The adapter updates the BQTOTAL column (or table field) in the event recovery table to the number of IDocs in the packet. This number is used for audit and recovery purposes.
4. If the adapter encounters an error while processing the IDoc packet, it can behave in one of two ways, depending on the IgnoreIDocPacketErrors configuration property:
   - If the IgnoreIDocPacketErrors property is set to `false`, the adapter stops processing any additional IDocs in the packet and reports errors to the SAP system.
   - If the IgnoreIDocPacketErrors property is set to `true`, the adapter logs an error and continues processing the rest of the IDocs in the packet. The status of the transaction is marked 3 (InProgress). In this case, the adapter log shows the IDoc numbers that failed, and you must resubmit those individual IDocs separately. You must also manually maintain these records in the event recovery table.
5. The SAP system sends a COMMIT call to the adapter.
6. The adapter sends the business objects to the message endpoint, one after the other, and updates the BQPROC property to the sequence number of the IDoc it is working on. The adapter delivers the objects to the appropriate endpoint as part of a unique XA transaction (a two-phase commit transaction) controlled by the application server.
7. When the endpoint receives the event and the transaction is committed, the adapter increments the number in the BQPROC property.

   Note: The message endpoint must be configured to support XA transactions.
8. After the adapter delivers all the business objects in the IDoc packet to the message endpoint, it updates the event status to 1 (Executed).
9. In case of abrupt interruptions during IDoc packet processing, the adapter resumes processing the IDocs from the current sequence number. The adapter continues updating the BQPROC property, even if IgnoreIDocPacketErrors is set to `true`. The adapter continues the processing in case you terminate the adapter manually while the adapter is processing an IDoc packet.
10. If an exception occurs either while the adapter is processesing the event or if the endpoint generates an exception, the event status is updated to `-1` (Rollback).
11. If no exception occurs, the adapter performs the following tasks:
    - It sends a COMMIT call to the adapter, and the event is updated to 1 (Executed).
    - It sends a CONFIRM call to the adapter.

    The adapter then deletes the records with a 1 (Executed) status and logs a common event infrastructure (CEI) event that can be used for tracking and auditing purposes.

Event processing for non-split, parsed IDoc packets:

The adapter processes an IDoc packet as one unit of work if all the IDocs in the packet are of the same IDoc type. All the IDocs in the IDoc packet are part of one XA transaction to an endpoint. The event processing is similar to a single IDoc with respect to the event recovery mechanism; the IDoc packet is considered a single unit of work, so a failure registered by the endpoint for any one IDoc causes the whole packet to fail. A failure requires that all the IDocs in the IDoc packet be sent to the endpoint again.

The adapter processes an inbound event for a non-split packet according to the following procedure:

1. When the SAP system sends a transaction ID to the adapter, the adapter checks the status of the event and takes one of the following actions:
   - If this is a new event, the adapter stores an EVNTID (which corresponds to the transaction ID) along with a status of `0` (Created) in the event recovery table.
   - If the event status is `-1` (Rollback), the adapter updates the status to `CREATED`.
   - If the event status is `1` (Executed), the adapter returns an indication of success to the SAP system.
2. The SAP system sends the IDoc packet to the adapter, which parses the IDoc packet, converts it to an array of IDoc objects, and stores it in memory.
3. The adapter sends the business objects to the endpoint. The adapter can deliver objects to endpoints that support transactions as well as to endpoints that do not support transactions.
   - For endpoints that support transactions, the adapter delivers the object as part of a unique XA transaction controlled by WebSphere Application Server. When the endpoint processes the event and the transaction is committed, the status of the event is updated to `1` (Executed).

     Note: The endpoint must be configured to support XA transactions.
   - For endpoints that do not support transactions, the adapter delivers the object to the endpoint and updates the status of the event to `1` (Executed). The adapter delivers the business object without the quality of service (QOS) that guarantees once only delivery.
4. If an exception occurs while the adapter is processing the event or if the endpoint generates an exception, the event status is updated to `-1` (Rollback).
5. If no exception occurs, the SAP system performs the following tasks:
   - It sends a COMMIT call to the adapter, and the event is updated to `1` (Executed).
   - It sends a CONFIRM call to the adapter.

   The adapter then deletes the records with a `1` (Executed) status and logs a common event infrastructure (CEI) event that can be used for tracking and auditing purposes.

Event processing for unparsed IDocs:

The adapter can process unparsed ALE events for applications or components that have parsing capability. The adapter processes the unparsed IDocs as it does for parsed IDocs, except that it does not parse the data part of the IDoc. The direct exchange of IDocs in the adapter enables high-performance, asynchronous interaction with SAP, because the parsing and serializing of the IDoc occurs outside the adapter.

The adapter processes the data based on whether the packet IDoc is split or non-split and whether the data needs to be parsed.

- The adapter can process packet IDocs as a packet or as individual IDocs. When an IDoc is received by the adapter from SAP as a packet IDoc, it is either split and processed as individual IDocs, or it is processed as a packet. The value of the SplitIDocPacket metadata at the business-object level determines how the IDoc is processed.

In the case of split IDocs, the wrapper contains only a single, unparsed IDoc object.

- The Type metadata specifies whether the data should be parsed. For unparsed IDocs, this value is `UNPARSEDIDOC`; for parsed IDocs, the value is `IDOC`. This value is set by enterprise service discovery.

Unparsed data format

In the fixed-width format of an unparsed IDoc, the segment data of the IDoc is set in the IDocData field of the business object. It is a byte array of fixed-length data.

The entire segment length might not be used. The adapter pads spaces to the fields that have data; the rest of the fields are ignored, and an end of segment is set. The end of segment is denoted by null.

The following figure shows a segment with fields demarcated by the '|' symbol for reference.

Figure 17. Example of a segment before processing

When the adapter processes this segment into unparsed data, it takes into account only those fields that have data in them. It maintains the field width for each segment field. When it finds the last field with data, it appends a null to mark the end of segment.

Figure 18. Example of a segment after processing

The next segment data processed as unparsed data would be appended after the null.

Limitations

The unparsed event feature introduces certain limitations on the enterprise application for a particular IDoc type.

- The enterprise application supports either parsed or unparsed business-object format for a given IDoc type or message type.
- For a given IDoc type, if you select unparsed business-object format for inbound, you cannot have inbound and outbound interfaces in the same EAR file, because outbound is based on parsed business objects.

IDoc status updates:

To monitor your IDoc processing, you can configure the adapter to update the IDoc status. When the adapter configuration property ALEUpdateStatus is set to `true` (indicating that an audit trail is required for all message types), the adapter updates the IDoc status of ALE business objects that are retrieved from the SAP

system. After the event is sent to the message endpoint, the adapter updates the status of the IDoc in SAP to indicate whether the processing succeeded or failed.

The adapter updates a status IDoc called ALEAUD that the adapter sends to the SAP system as an inbound IDoc event.

The IDoc status codes are defined in the following table:

Table 3. IDoc status codes

IDoc status code value	Description
12	The dispatch processed without errors.
11	An error occurred during dispatch.

An IDoc that is not successfully sent to the endpoint is considered a failure, and the IDoc status is updated by the adapter to 11. Similarly, an IDoc that reaches the endpoint is considered successfully processed, and the status of the IDoc is updated to 12.

These codes and their associated text are configurable properties of the adapter, as specified in the J2C activation specification properties. The properties and their values are listed in the following table:

Table 4. Configuration properties for IDoc status codes

Adapter property	Value
ALESuccessCode	12
ALEFailureCode	11
ALESuccessText	Dispatch OK
ALEFailureText	Error during dispatch

For inbound processing, perform the following tasks to ensure that the adapter updates the standard SAP status code after it retrieves an IDoc:

- Set the AleUpdateStatus configuration property to `true` and set values for the AleSuccessCode and AleFailureCode configuration properties.
- Configure the inbound parameters of the partner profile of the logical system in SAP to receive the ALEAUD message type. Set the following properties to the specified values:

Table 5. Inbound properties of the logical system partner profile

SAP property	Value
Basic Type	ALEAUD01
Logical Message Type	ALEAUD
Function module	IDOC_INPUT_ALEAUD
Process Code	AUD1

ALE business objects

WebSphere Adapter for SAP Software depends on the IDoc metadata that is generated by the enterprise service discovery wizard to construct business objects. This metadata contains ALE-related information such as segment information, field names, and an indication of whether the business object handles a single IDoc or an IDoc packet.

ALE business object structure:

During ALE processing, the adapter exchanges business objects with the SAP application. The business object represents an individual IDoc or an IDoc packet. This business object is a top-level wrapper object that contains one or more IDoc child objects, each one corresponding to a single IDoc. The same business object is used for inbound and outbound processing.

The ALE business structure is shown in the following figure.

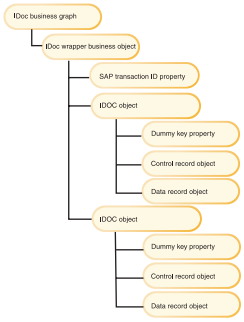


Figure 19. ALE business structure

An ALE business graph contains a verb and a business object. The business graph can refer to a single cardinality business object or to a wrapper representing a group of business objects, each with single cardinality.

ALE IDoc business objects support the following verbs:

- Outbound business objects: No verb support
- Inbound business objects: Create, UpdateWithDelete, Delete

The wrapper business object contains a transaction ID and one or more IDoc business objects. For individual IDocs, the wrapper business object contains only one instance of an IDoc business object. For IDoc packets, the wrapper business object contains multiple instances of an IDoc business object.

The following figure illustrates a business graph and its associated wrapper business object, which, in this example, contains one IDoc business object.

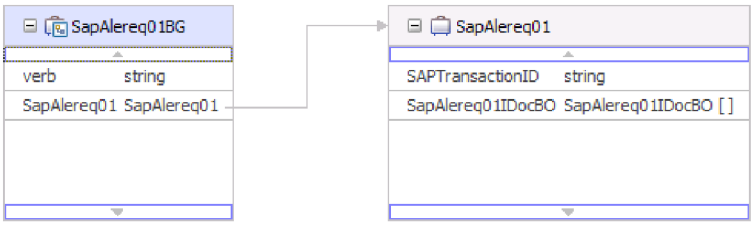


Figure 20. Example of an ALE business graph and wrapper business object

The IDoc business object (SapAlereq01IDocBO, in the example) has the structure shown in the following figure.

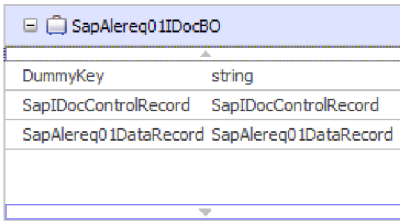


Figure 21. Example IDoc business object structure

The IDoc business object contains the following objects:

- The control record business object contains the metadata required by the adapter to process the business object.
- The data record business object contains the actual business object data to be processed by the SAP application and the metadata required for the adapter to convert it to an IDoc structure for the RFC call.

Transaction ID support:

An SAP transaction ID is contained within the ALE wrapper business object. You can use transaction ID support to ensure once-only delivery of ALE objects.

You can make use of transaction ID support for ALE processing. The most common reason for using transaction ID support is to ensure once and only once delivery of data. Install the transaction RAR file (CWYAP_SAPAdapter_Tx.rar) to use this feature, and import that RAR file into your adapter project before enterprise service discovery is performed.

Note: The SAP transaction ID property is always generated by enterprise service discovery; however, it is supported only for outbound operations when the CWYAP_SAPAdapter_Tx.rar version of the adapter is used.

The client application must determine how to store the SAP transaction ID and how to relate the SAP transaction ID to the data being sent to the adapter. When the events are successful, the client application should not resubmit the event associated with this TID again to prevent the processing of duplicate events.

- If the client application does not send an SAP transaction ID with the business object, the adapter returns one after executing the transaction.
- If the client application has an SAP transaction ID, it needs to populate the SAP transaction ID property with that value before executing the transaction.

The SAP transaction ID can be used for cross referencing with a global unique ID that is created for an outbound event. The global unique ID is something you can create for managing integration scenarios.

Dummy keys:

You use a dummy key to map a key field from an IDoc control or data record business object to the dummyKey property of the top-level business object. The dummyKey property is used for flow control and business process logic. You can use the dummyKey when you need the top-level business object to participate in a relationship.

The adapter supports dummy key mapping in the following manner:

- You must configure the property-level application-specific information of the dummyKey property as the XPATH of the property from which the value should be set. In other words, the property-level application-specific information is set to the XPATH within the business object hierarchy of the attribute that is being mapped to the top-level object.
- If multiple cardinality objects are found in this path, the adapter uses the cardinality defined in the XPATH. This is true for all multiple cardinality objects wherever they occur in the hierarchy. The following line of code is an example of the XPATH:

  ```
  <sapasi:ForeignBOKeyRef>Orders05/Orders05DataRecord/ Orders05E2edk14[1]/
  OrgID />
  ```
- If the application-specific information is incorrect or if the mapped property value is empty, the adapter causes the event to fail. This is also the case when the application-specific information is configured to set an object type value as the dummyKey.

  Note: The dummyKey property can contain only a simple type.

The SQI interface

The SAP query interface (SQI) provides you with the means to retrieve data from application tables on an SAP system or to query SAP application tables for the existence of data. The adapter can perform hierarchical data retrieval from the SAP application tables.

SQI supports outbound interactions for read operations (RetrieveAll and Exists) only. You can use this interface in local transactions to look up records before write operations (Create, UpdateWithDelete, or Delete). For example, you could use SQI as part of a local transaction to do an existence check on a customer before creating a sales order. You can also use the interface in non-transaction scenarios.

The SQI interface supports data retrieval form SAP application tables, including hierarchical data retrieval from multiple tables. The interface supports static as well as dynamic specification of where clauses for the queries.

Enterprise service discovery finds and builds the hierarchical business object structures for multiple tables using the relationships between them. With enterprise service discovery, you can also build a default where clause for the query.

Outbound processing

You use the SAP query interface for outbound processing only.

The following steps describe how the adapter supports outbound processing for the SQI interface.

Note: The client application that makes the request uses the interface information that was generated by the enterprise service discovery wizard.

1. The adapter receives a request, which includes a table object, from a client application.
2. The adapter determines, from the table object sent with the query, the name of the table to examine.
3. The adapter determines the columns to retrieve or examine.
4. The adapter determines the rows to retrieve or examine.
5. The adapter responds.

- In the case of a RetreiveAll operation, the adapter returns the data.
- In the case of the Exists operation, the adapter returns an indication of whether the data exists in the SAP table.
- If no data exists, the adapter generates an exception.

Business objects

An SQI business graph, which acts as the input to the SQI interface, consists of a table business object and verb. The table business object represents the columns in a table on the SAP server. The verbs supported for an SQI business graph are RetrieveAll and Exists.

An example of a business graph associated with a table business object is shown in the following figure.

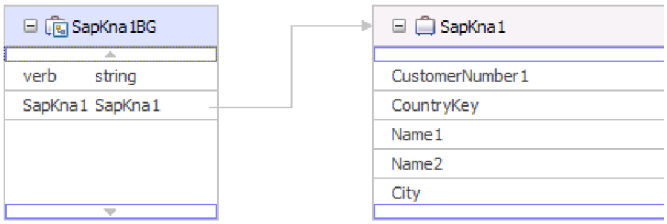


Figure 22. Example of an SQI business graph

The table business object contains columns selected from the specified SAP table. An example of a table business object (representing the KNA1 table) is shown in the following figure.

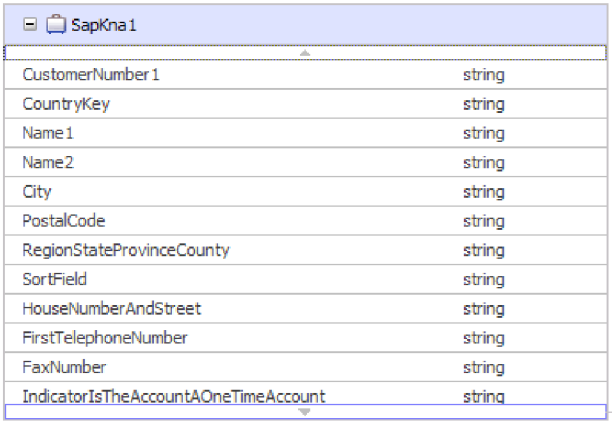


Figure 23. Example of an SQI table business object

In addition to column information, the table business object also contains a query business object as the last parameter.

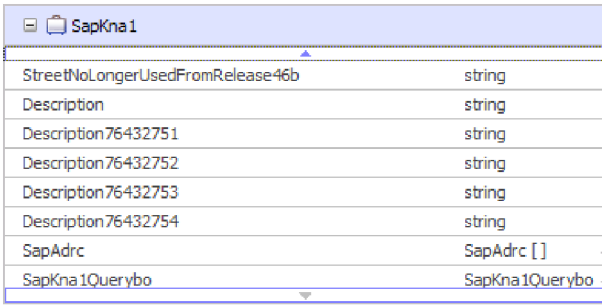


Figure 24. The query business object as a parameter of the table business object (represented by the SapKna1Querybo parameter)

The query business object looks like this:

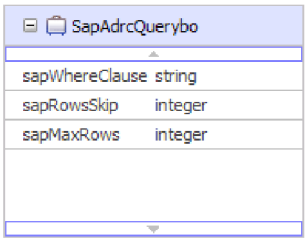


Figure 25. An example of an SQI query business object

The tables can be modeled as hierarchical business objects. You specify the parent-child relationship of the tables when you select the business objects during the enterprise service discovery process.

Tables are linked by a foreign key to form parent-child relationships. The child table business object has a foreign key that references a property in the parent query business object.

In the KNA1 business object, notice the reference to SapAdrc, a child business object. The SapAdrc table object, shown in the following figure, has a column named AddressNumber. This column has an associated property (ForeignKey) that contains a reference to the parent business object.

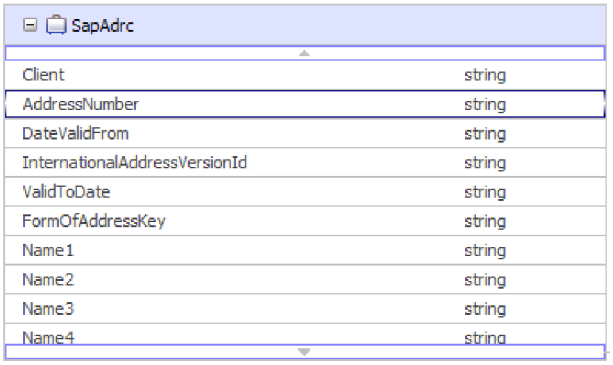


Figure 26. An example of a child table object

You can see the property by clicking **AddressNumber** and looking at the Properties tab.

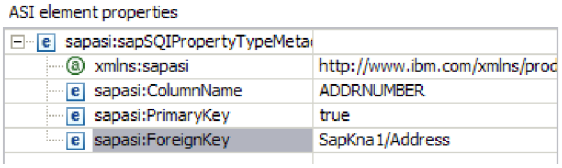


Figure 27. Example of the property metadata that links the child object to the parent object

The ForeignKey property contains a reference to the Address column of the SapKna1 table object.

The return from the SQI interface call for a RetrieveAll operation is a container of business graphs.

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 and WebSphere Enterprise Service Bus use the Windows standard format, but an enterprise information system exchanging data with WebSphere Process Server or WebSphere Enterprise Service Bus 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 before introducing the data to WebSphere Process Server or WebSphere Enterprise Service Bus .

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

Table 6. 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 is 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 or WebSphere Enterprise Service Bus.

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.

Table 7. Bidirectional property types

Property type	Data transformations
EIS	Controls the format for content data (data that is sent by the enterprise information system).
Metadata	Controls the format for metadata (data that provides information about the content data).
Skip	Identifies content or metadata to exclude from transformation.
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. Use the WebSphere Process Server administrative console to configure these properties.
- **Managed (J2C) connection factory properties:** These properties are used at run time to create an outbound connection instance with an enterprise information system. After the managed connection factory 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 administrative console of the server.

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 set values for EIS-type bidirectional properties for the resource adapter only, 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 (managed connection factory) 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 install WebSphere Adapter for SAP Software, make sure you have the correct environment and prerequisite software. Also consider any performance implications of the installation.

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.js.

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.

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.

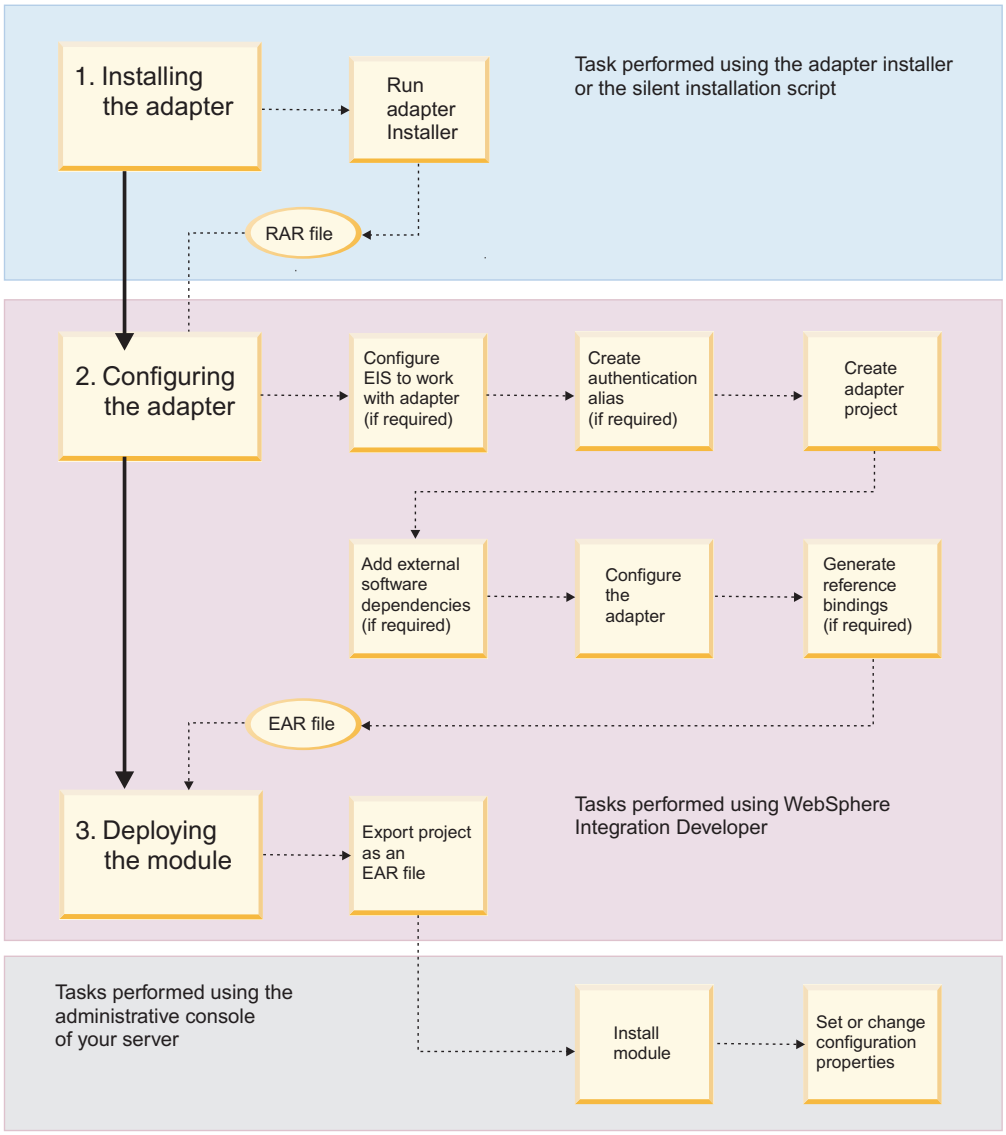


Figure 28. Roadmap for installing, configuring, and deploying the adapter

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 the installation steps common to all adapters, and perform additional installation steps specific to WebSphere Adapter for SAP. You can then update and migrate information from your existing adapter installation.

Installation prerequisites

Before installing Adapter for SAP Software, you must meet all of the 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 WebSphere Adapter for SAP Software, 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 connector project. For more information on how to do this, see ″Adding jar files to WebSphere Integration Developer versions 6011 and earlier″ in the reference section.

Performing the installation

The basic 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. After performing the common installation steps, you must also perform installation steps specific to WebSphere Adapter for SAP Software.

Before you begin

Review the installation prerequisites.

How to perform this task

1. Install the adapter using the basic installation instructions, which are common to all adapters. These steps are located in the Installing section of Installing IBM WebSphere Adapters.

2. Perform the following steps, which are specific to WebSphere Adapter for SAP Software.
   a. Copy the dependencies libraries to the bin subdirectory of the WebSphere Process Server or WebSphere Enterprise Service Bus installation directory.

      The installation directory is typically in the runtimes\bi_v6 directory of the WebSphere Integration Developer installation directory.

      For z/OS®, add the files to the ${*WAS_INSTALL_ROOT*}/lib directory.

Table 8. Files to be installed

Operating system	Files to be installed
Windows	Any *.dll files that come with the SAP Jco download from the SAP Web site
Unix (including Unix System Services on z/OS)	Any *.so or *.o files that come with the SAP Jco download from the SAP Web site

   b. For Windows environments only, install the msvcp71.dll and msvcr71.dll files in the Windows system path.
   c. Install the SAP Java Connector interface (sapjco.jar) to the lib subdirectory of the WebSphere Process Server or WebSphere Enterprise Service Bus installation directory.

      For z/OS, add ${*WAS_INSTALL_ROOT*}/lib/sapjco.jar to WAS_SERVER_ONLY_server_region_classpath
3. Make sure you know the following information for accessing the SAP application:
   - SAP user name
   - SAP password
   - SAP host name (or IP address)
   - SAP system number (usually 00)
   - SAP client number (usually 100)

Result

The resource adapter archive (RAR) file is copied to the workstation where the adapter is installed. You can view the installed files and directories, all of which have the installation directory as their root. If you accepted the default installation location, the RAR file is placed in one of the following paths. The second path represents the version of the adapter that supports J2C local transactions.

- C:\Program Files\IBM\ResourceAdapters\SAP\adapter\SAP\deploy\CWYAP_SAPAdapter.rar
- C:\Program Files\IBM\ResourceAdapters\SAP\adapter\SAP\deploy\CWYAP_SAPAdapterTX.rar

What to do next

Configure the adapter.

Migrating to version 6.0.2

If you have adapter applications created with an earlier version of WebSphere Adapter for SAP Software and you are upgrading to WebSphere Adapter for SAP Software version 6.0.2, you might need to make changes to the existing applications. Become familiar with the features that are being deprecated and make any required changes to your applications based on changes in version 6.0.2.

Migration considerations

WebSphere Adapter for SAP Software version 6.0.2 includes changes to earlier versions of the adapter that might affect your existing adapter applications. Review information on changed features (such as the change in the way the event recovery is implemented) and deprecated features (such as changes to the ALE business-object structure) so that you can decide whether you need to make changes to your applications. Note that there has been no change to the BAPI business-object structure.

Changed event-recovery implementation

The event recovery feature has changed. For event recovery, the WebSphere Adapter for SAP Software now uses the data source configured through the WebSphere Process Server or WebSphere Enterprise Service Bus administrative console.

Deprecated features

A deprecated feature is one that is supported but no longer recommended and that might become obsolete. Features from earlier versions of WebSphere Adapter for SAP Software that have been deprecated in version 6.0.2 include certain enterprise service discovery, resource adapter, and adapter specification properties, as well as the existing IDoc business structure.

- Single IDoc business structure

  Earlier versions of the adapter included two business-object structures for the ALE interface–one for a single IDoc and one for an IDoc packet. For version 6.0.2, one ALE business-object structure supports both single IDocs and IDoc packets. The earlier single IDoc business object, shown in the following figure, has been deprecated.

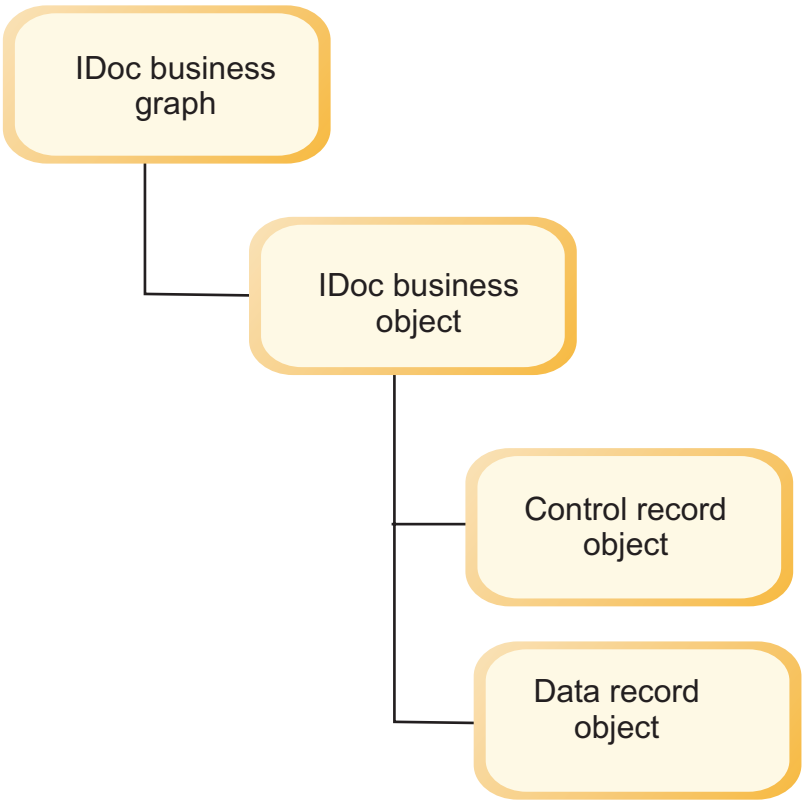


Figure 29. Deprecated ALE business object structure

For backward compatibility, the earlier business object structure is still supported.

- Enterprise service discovery property

  The ALE interface uses the same object definition for both inbound and outbound event processing. The Create ALE Wrapper property has therefore been deprecated.
- Resource adapters properties

  The PartnerCharSet adapter-level property has been deprecated. This property is now configured as a managed connection factory property or an activation specification property. This change is required for supporting multiple EIS systems.
- Activation specification properties

  The following activation specification properties have been deprecated:
  - EDTDriverName
  - EDTDatabaseName
  - EDTUserName
  - EDTUserPassword
  - EDTSchemaName
  - EDTURL
  - EDTServerName
  - EDTPortNumber
  - SplitIDocs

Performing the migration

To update your existing SAP adapter applications to include the newly installed adapter, you replace the existing RAR file with the WebSphere Adapter for SAP Software version 6.0.2 RAR file. To migrate inbound ALE applications, you modify the event recovery table and update activation specifications properties.

Updating adapter applications

To update an existing adapter application to include the newly installed adapter, you replace the RAR file from a previous version of the adapter with the RAR file installed as part of the WebSphere Adapter for SAP Software version 6.0.2 installation.

Before you begin

Make sure you have installed the new version of Adapter for SAP Software and have determined that your existing adapter application must be updated.

About this task

The way that you upgrade an adapter application depends on whether the application has already been deployed to a server (WebSphere Process Server or WebSphere Enterprise Service Bus) or whether the application is in the test environment of WebSphere Integration Developer. If the application has already been deployed, you use the administrative console to replace the adapter RAR file in the application. If the application is in the test environment of WebSphere Integration Developer, you import the new RAR file into the adapter project.

To update an adapter application, use the procedure that applies to your environment.

- If your adapter application has already been deployed to WebSphere Process Server or WebSphere Enterprise Service Bus, use the following procedure.
  1. From the administrative console of the server, click **Applications → Enterprise Applications**.
  2. Select the enterprise application that uses the adapter to be upgraded.
  3. From the Configuration page, under **Related Items**, click **Connector Modules**.
  4. On the Connector Modules page, select the check box next to the name of the RAR file, and then click **Update**.
  5. Select **Single Module**.
  6. Select **Local File System**, and browse for the new adapter RAR file. Then click **Next**.
  7. Click **Next** on the following page.
  8. On the Install New Application page, select **Step 4: Summary**, and then click **Finish**.
  9. Click **Save changes to Master Configuration**.
- If your adapter application is in a WebSphere Integration Developer test environment, use the following procedure.
  1. From the J2EE perspective of WebSphere Integration Developer, select the connector project to be updated.
  2. Import the new adapter RAR file by clicking **Import → RAR file**.
  3. Build and deploy any dependent applications as necessary.

Result

The adapter application contains the updated (WebSphere Adapter for SAP Software version 6.0.2) RAR file.

Migrating the event recovery table

To use the new event-recovery table mechanism, you can either create a new event recovery table or transfer data from an existing event recovery table. Use the procedures in a development environment and thoroughly test the applications before deploying them in production.

Creating a new event table:

To create a new event-recovery table for inbound events, you configure a new data source and run enterprise service discovery. During enterprise service discovery, the table is automatically created.

Before you begin

Make sure you have installed the new version of Adapter for SAP Software and that you have updated the adapter RAR file in WebSphere Integration Developer.

About this task

To create an event-recovery table, you create a new data source and use enterprise service discovery to generate new business objects. During enterprise service discovery, an event recovery table is automatically created.

To configure the data source and run enterprise service discovery, use the following procedure.

How to perform this task

1. Configure a new data source for ALE event recovery.

   See "Configuring the data source" on page 70 for the steps necessary to configure the data source.
2. Stop new events from being triggered and complete the existing flows for inbound ALE events.
3. Stop the application.

   When the migrated application is deployed and started, the adapter creates the event table based on the new structure.
4. Use enterprise service discovery to generate new service descriptions and business objects.
   a. In the Generate Artifacts window, make sure the inbound connection property **Auto Create Event Table** is selected.
   b. Make sure that a table with the name in the **Event recovery table name** field does not already exist in the database.
5. Fix any breakages in dependent components.
6. Deploy and start the application.

Result

A new event recovery table is created, and the associated adapter module is deployed.

Migrating properties from an existing table:

To migrate properties from an existing event-recovery table, you configure a new data source, create an event-recovery table, move the properties to the new table, and run enterprise service discovery.

Before you begin

Make sure you have installed the new version of Adapter for SAP Software and that you have updated the adapter RAR file in WebSphere Integration Developer.

About this task

To transfer data from an existing event-recovery table, you create a new data source, transfer data from the existing table to the new data source (using the information in Table 9, and use enterprise service discovery to generate new business objects.

To configure the data source and run enterprise service discovery, use the following procedure.

How to perform this task

1. Configure a new data source for ALE event recovery.

   See "Configuring the data source" on page 70 for the steps necessary to configure the data source.
2. Create the event recovery table in the data source.
3. Stop the application.
4. Transfer entries from the existing event recovery table to the new event recovery table, using the information in the following table as a guide.

Table 9. Mapping of new event recovery table entries to deprecated entries

Event persistence table column	Column type and size	Deprecated table column	Deprecated column type and size
EVNTID	VARCHAR(255)	TID	VARCHAR(255)
EVNTSTAT	INTEGER	Status	VARCHAR(255)
XID	VARCHAR (255)	N/A	N/A
BQTOTAL	INTEGER	NumIDocs	INTEGER
N/A	N/A	NumIDocsProcessed	INTEGER
BQPROC	INTEGER	CurrIDoc	INTEGER
EVNTDATA	VARCHAR(255)	N/A	N/A

Use the following mapping between the previous table and the new event recovery table.

Table 10. Mapping of the new event status entries to the deprecated entries

New event status (INTEGER)	Deprecated event status (VARCHAR)
0	CREATED
1	EXECUTED
3	PARTIAL
-1	ROLLBACK

5. Use enterprise service discovery to generate new service descriptions and business objects.

   In the Generate Artifacts window, make sure that **Event recovery table name** is set to the table that contains the migrated event-recovery data.
6. Fix any breakages in dependent components.
7. Deploy and start the application.

Result

A new event recovery table is created, and the associated adapter module is deployed.

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 264.

Chapter 7. Configuring the adapter for deployment

To configure WebSphere Adapter for SAP Software 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.

Creating the authentication alias

To create the authentication alias on the server, use the administrative console. From the administrative console, configure the global security and set the password for the authentication alias, which is used to process outbound requests.

Before you begin

You must have access to the administrative console.

About this task

To create an authentication alias, use the following procedure.

How to perform this task

1. In the administrative console, click **Security** → **Global security**.
2. Under **Authentication**, click **JAAS Configuration** → **J2C Authentication data**.
3. Create an authentication alias
   a. Click **New**.
   b. In the General properties window, type the name of the alias in the **Alias** field.
   c. Type the user ID and password that are required to connect to the SAP server.
   d. Click **OK**.

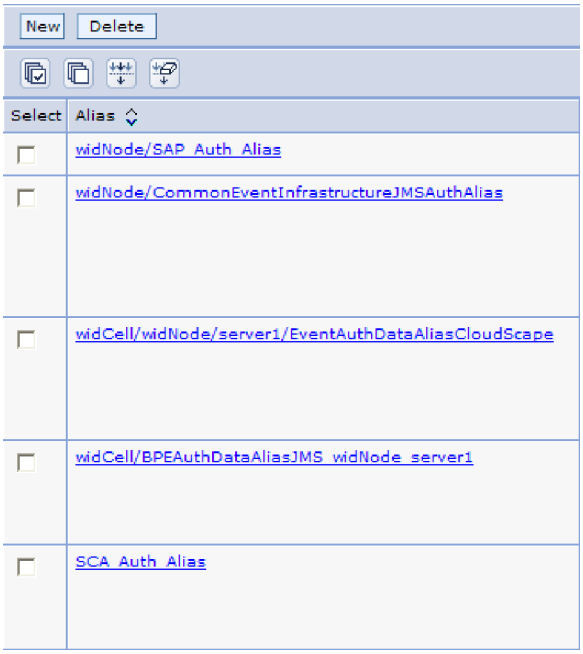


Figure 30. The list of aliases

Make note of the name as it appears in the Alias list. In the example, the name is **widNode/SAP_Auth_Alias**. This name is the one you use in subsequent configuration windows.

e. Click **Save**, and then click **Save** again.

Result

You have created an authentication alias, which you will use when you configure the adapter properties.

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 Adapter for SAP Software and that you have created an authentication alias.

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 V6.0.2** → **WebSphere Integration Developer V6.0.2**.
   b. If you are prompted to specify a workspace, accept the default value or select another workspace.

      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**.

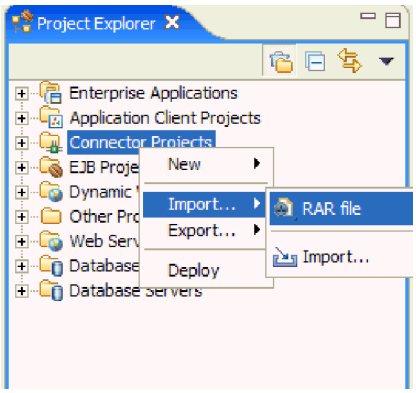


Figure 31. Importing the RAR file

4. From the Connector Import window, click **Browse** and navigate to the directory in which Adapter for SAP Software was installed.
5. Click either **CWYAP_SAPAdapter.rar** or **CWYAP_SAPAdapterTX.rar**, depending on which adapter you installed.

   The connector project has the same name as the RAR file.

   If a project named CWYAP_SAPAdapter.rar already exists in this workspace, the name in the **Connector project** field has a number appended to it (for example, CWYAP_SAPAdapter1).
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.

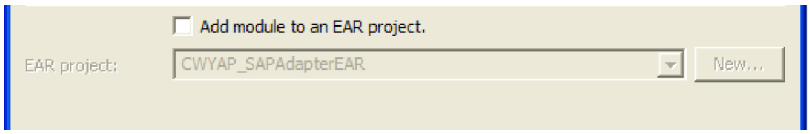


Figure 32. Clearing the Add module to an EAR project check box

Notice that the EAR project field becomes unavailable after you remove the check mark.

9. Click **Finish**.

Result

A new adapter project is created and is listed under **Connector Projects** in the Project Explorer window. To see its contents, expand the project. For example, if the project is named CWYAP_SAPAdapter, expand **CWYAP_SAPAdapter**.

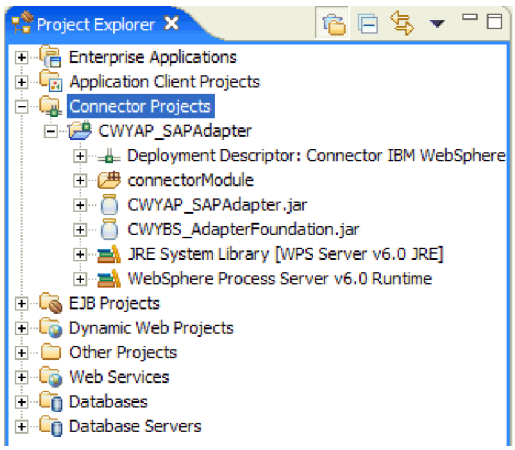


Figure 33. The Connector Projects section of the Project Explorer window

What to do next

Add the required external dependencies to the project.

Adding external software dependencies

To add the required sapjco.jar file to the Java build path of your adapter project, make sure you have installed sapjco.jar (and other files required to run the adapter) as part of the post-installation tasks. You then use WebSphere Integration Developer to import the sapjco.jar file into the adapter project.

Before you begin

Make sure you have installed the operating-system-dependent files required to run the adapter, as described in “Performing the installation” on page 41. Also make sure you have created the adapter project.

About this task

You import the sapjco.jar file into the Java build path of your adapter project.

To import the file, use the following procedure.

How to perform this task

1. Make sure you have copied the files for your operating system, as described in "Performing the installation" on page 41.
2. Import the sapjco.jar file into the adapter project.
   a. In the J2EE perspective of WebSphere Integration Developer, expand **Connector Projects**.
   b. Right-click **CWYAP_SAPAdapter** and click **Properties**.
   c. In the left side of the Properties for CWYAP_SAPAdapter window, click **Java Build Path**.

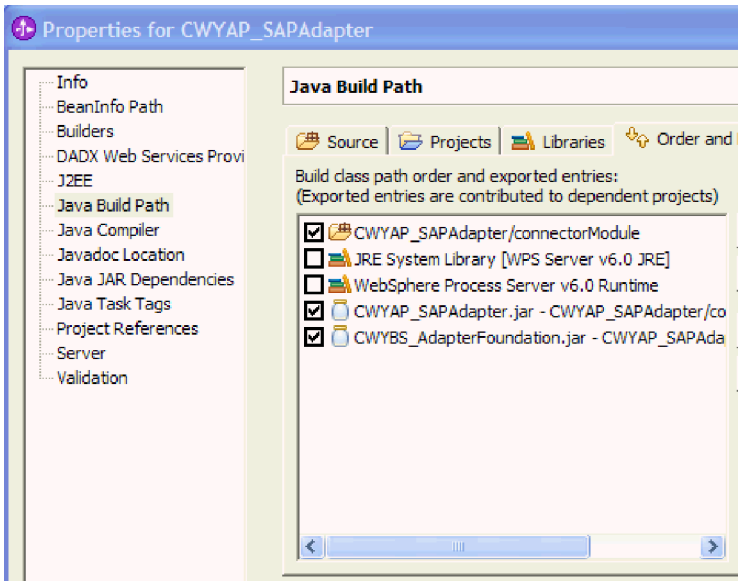


Figure 34. Selecting Java Build Path

   d. Click the **Libraries** tab, and click **Add External JARs**.
   e. Navigate to the directory on your local file system where the sapjco.jar file is located. Then select **sapjco.jar** and click **Open**.
   f. Click **OK**.

      The file sapjco.jar appears in the list of JARs and class folders in the build path.

Result

The sapjco.jar file is now part of your adapter project and appears in the Project Explorer window of WebSphere Integration Developer.

What to do next

Configure the adapter. The first step in the process of configuring the adapter is to specify information about the SAP server so that the enterprise service discovery wizard can connect to the server.

Configuring the adapter

To configure the adapter, use the enterprise service discovery wizard in WebSphere Integration Developer to set connection properties, select business objects or services from an SAP server, and generate business object definitions and related artifacts.

Configuring the adapter for the BAPI interface

To configure the adapter for BAPI outbound processing, you use the enterprise service discovery wizard to find a BAPI or set of BAPIs. You then configure the business objects that are generated and create a deployable module.

Setting connection properties for enterprise service discovery

To set connection properties for the enterprise service discovery wizard so that it can access the SAP server, specify such information as the user name and password you use to access the server as well as the name or IP address of the server.

Before you begin

Make sure you have successfully added the external dependencies.

About this task

Specify the connection properties that the enterprise service discovery wizard needs to connect to the SAP server and discover its business objects and services.

To specify the connection properties, use the following procedure.

How to perform this task

1. In WebSphere Integration Developer, start the enterprise service discovery wizard.
   a. Switch to the Business Integration Perspective by clicking **Window → Open Perspective → Other**. Then click **Business Integration (default)** and click **OK**.
   b. Right-click **New → Enterprise Service Discovery**.

      If **Enterprise Service Discovery** is not displayed, click **New → Other**, expand **Business Integration**, and click **Enterprise Service Discovery**. Then click **Next**.

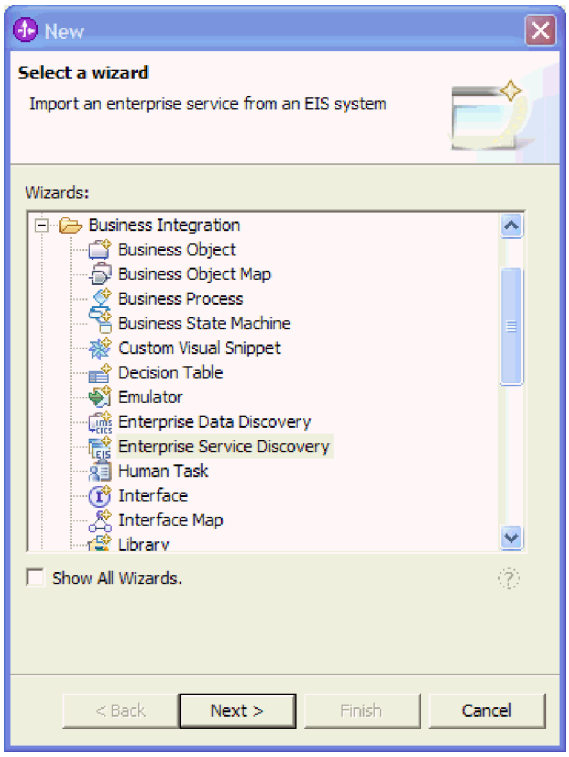


Figure 35. The expanded list of wizards

2. Select **IBM WebSphere Adapter for SAP Software** and click **Next**.

   If you previously ran the enterprise service discovery wizard, your connection properties appear when you expand the adapter name node by clicking the plus symbol (+). You can select the saved connection properties if you plan to connect to the SAP application you used the last time you ran the enterprise service discovery wizard.

   Note: Properties marked with an asterisk (*) are required.
3. Specify the configuration properties to initialize the discovery agent:
   a. Type the name and password you use to access the SAP system.

      The password is case-sensitive.
   b. Type your client ID.
   c. Optionally change the default settings for **Language**, **Codepage Number**, and **System Number**.
   d. In the **Application Server Host** field, type the name (or IP address) of your SAP server.

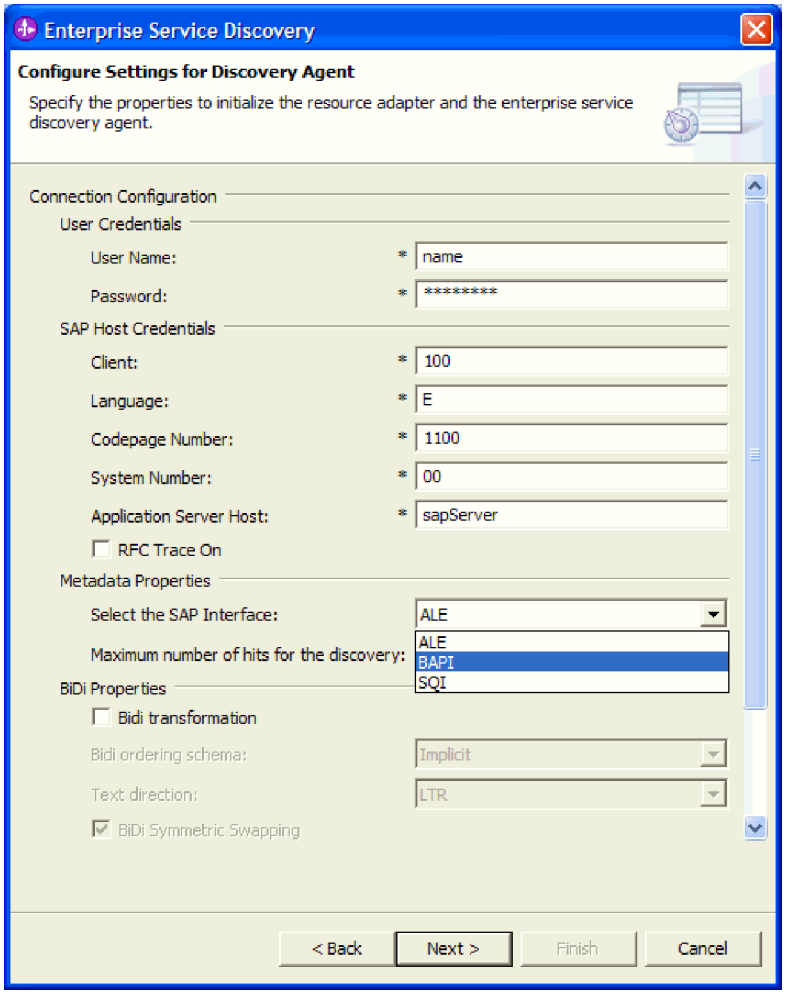


Figure 36. The Configure Settings for Discovery Agent window

4. Select **BAPI** from the **Select the SAP interface** list.
5. **Optional:** Indicate the number of functions you want returned by changing the value in the **Maximum number of hits for the discovery** field or by accepting the default value.
6. **Optional:** If you need to set bidirectional properties, perform the following steps:
   a. Select **Bidi transformation**.
   b. Set properties for your environment. See "Enterprise service discovery connection properties" on page 249 for more information about these properties.
7. **Optional:** To change the logging level for enterprise service discovery, 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.

      Note: This log pertains to enterprise service discovery only, not to the operation of the adapter.

8. Click **Next**.

Result

The enterprise service discovery wizard contacts the SAP server, using the information you provided (such as user name and password) to log in. You see the Find and Discover Enterprise Services window.

What to do next

Specify search criteria that the enterprise service discovery wizard uses to discover BAPI functions on the SAP server.

Selecting business objects and services

To specify which BAPI function you want to call and which data you want to process, you provide information in the enterprise service discovery wizard.

Before you begin

Make sure you have set the connection properties for enterprise service discovery.

About this task

Specify search criteria that the enterprise service discovery wizard uses to discover BAPI functions on the SAP server. The enterprise service discovery wizard returns a list of BAPI functions that meet the search criteria.

To specify the search criteria and select one or more BAPI functions, use the following procedure.

How to perform this task

1. In the Find and Discover Enterprise Services window, click **Execute Query**.
2. Under **Objects Discovered by query**, expand **RFC** or **BOR** and select **Discover By Name** or **Discover By Description**.

   The **Filter** button is now enabled.
3. Click **Filter**.

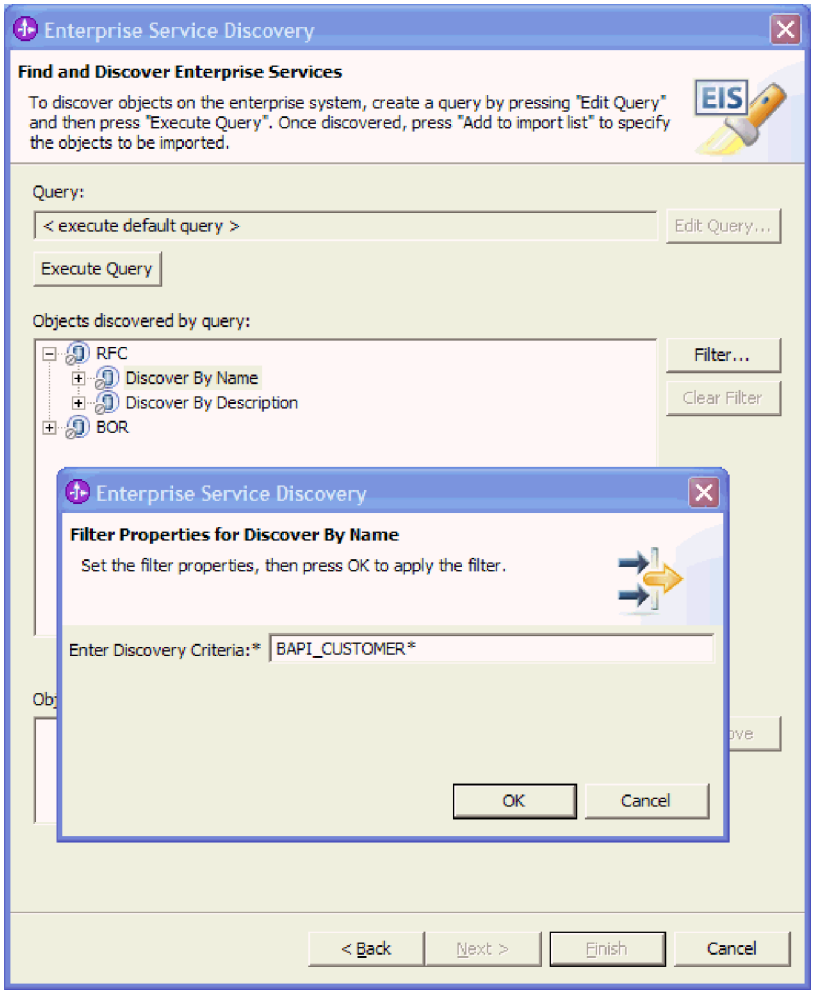


Figure 37. Find and Discover Enterprise Services window

4. Type a search string (for example, BAPI_CUSTOMER*) representing the BAPI you want to call.

   This is the name of the BAPI in SAP plus an asterisk as a wild card character to indicate that you want a list of all SAP application components that start with the phrase BAPI_CUSTOMER.
5. Click **OK**.
6. Expand **Discover By Name** or **Discover By Description** (whichever choice you made in the earlier step).
7. Click the BAPI you want to use. If you are working with multiple BAPI transactions, click the names of all the BAPIs.

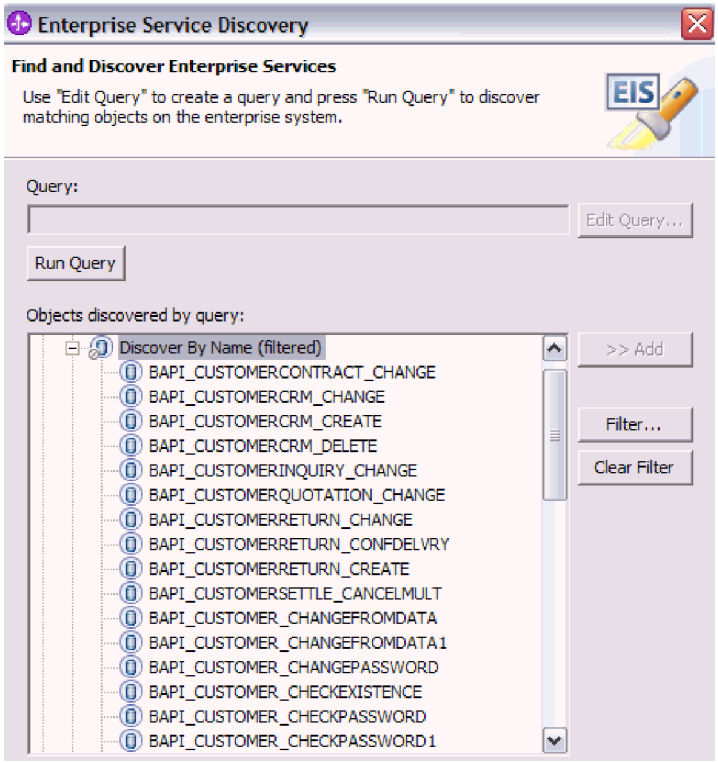


Figure 38. Objects discovered by query

8. Click **Add to import list**.
9. In the Configuration Parameters window, perform the following tasks for each BAPI to add it to the list of business objects to be imported:
   a. Optionally select the **Use Field Name to generate attribute(s)** check box. By default (when the check box is not selected), the field descriptions are used to generate attributes.
   b. If the BAPI has optional parameters associated with it, select the **Check if you want to select optional parameters for this interface** check box and then select the optional parameters you want included in your business object definition.

      By default, enterprise service discovery generates all the parameters required for the selected BAPI interface, so select this check box and then clear the check boxes for any parameters you do not want to include in your business object.

      For example, if you are adding the CUSTOMER_CHANGEFROMDATA BAPI, you have the option of adding the following parameters:

      PI_DIVISION

      PI_DISTR_CHAN

      Refer to the SAP documentation for a list and description of the optional parameters.
   c. Click **OK** to add the BAPI to the list of business objects to be imported.

   If you want to remove an object from the list, select the object name and click **Remove**.
10. Repeat step 9 until you have configured all BAPIs.
11. Click **Next**

Result

The enterprise service discovery wizard has returned the function or functions that match the search criteria, and you have selected the function or functions you want to work with.

What to do next

Specify a name for the business object and the directory to which it should be stored.

Configuring the selected objects

To configure the business object, you specify information about the object (such as the name of the object and the operation associated with the object).

Before you begin

Make sure you have selected and imported the BAPI function.

About this task

Configure the business object that was imported. Name the object, indicate where the object should be stored, and specify the operation associated with it.

To configure the business object, use the following procedure.

How to perform this task

1. In the Configure Objects window, type the name of the object location (where the object is stored)
2. In the **Namespace** field, use the default namespace (http://www.ibm.com/xmlns/prod/websphere/j2ca/sap) except in the following circumstance. If you are adding the business object to an existing module and the module already includes that business object (from an earlier run of the enterprise service discovery), change the namespace value.

   For example, you could change the namespace to http://www.ibm.com/xmlns/prod/websphere/j2ca/sap1.
3. Type a name for the business object.
4. If you are working with a BAPI transaction, perform the following tasks:
   a. Select the **Create a BAPI transaction object from the selected BAPIs** check box.
   b. Click **Add**.

      The BAPIs you previously selected and a COMMIT operation are listed.
   c. For each BAPI in the transaction, add the BAPI in the order in which it should be executed by selecting the BAPI and clicking **OK**.
   d. After you have added all the BAPIs, select **COMMIT**, and click **OK**.
5. Perform one of the following sets of tasks, depending on whether you have selected one BAPI, multiple BAPIs, or a BAPI transaction:
   - If you are working with a single BAPI, select an operation (for example, **Retrieve**).
   - If you are working with multiple BAPIs, select, for each operation, the BAPI you want associated with that operation.

     For example, if you have selected two BAPIs, one that is intended to perform a retrieve operation and one that is intended to perform a delete operation,

you would select the first BAPI from the list next to **Retrieve**, and you would select the second BAPI from the list next to **Delete**.

- If you are working with a BAPI transaction, select an operation from the **Choose the operation for this Transaction Business Object** list.

6. Click **Next**.

Result

You have associated an operation with the object and selected a name for the object. The Generate Artifacts window is displayed.

What to do next

Generate a deployable module that includes the adapter and the business object.

Generating artifacts

To generate the module, which is the artifact that is deployed on WebSphere Process Server or WebSphere Enterprise Service Bus, you create the module, include the adapter in the module, and specify an alias used to authenticate the caller to the SAP server.

Before you begin

Make sure you have configured the business object. The Generate Artifacts window should be displayed.

About this task

Generate the module, which includes the adapter and configured business object. The module is the artifact you deploy on the server.

To generate the module, use the following procedure.

How to perform this task

1. In the Generate Artifacts window, create a new module.
   a. Click **New**.
   b. Click **Create a module project** and click **Next**.
   c. Type a name for the module.

      As you type the name, it is added to the workplace specified in the **Directory** field.

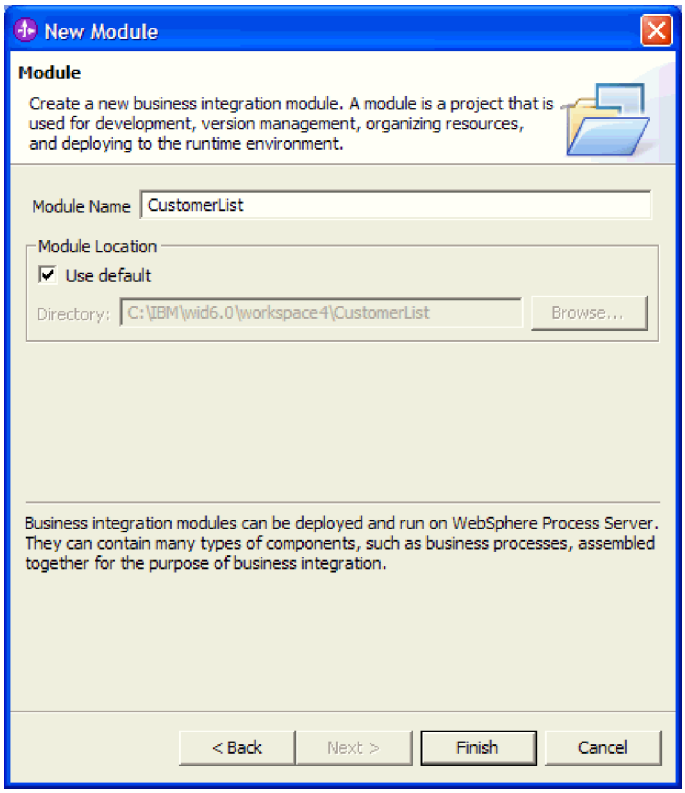


Figure 39. The New Module window

d. Click **Finish**.

2. If you want to change the default namespace, clear the **Use Default Namespace** check box and type a new path in the **Namespace** field.
3. In the Generate Artifacts window, specify the folder within the module where the service description should be saved by typing a name in the **Folder** field or browsing for a folder.
4. Optionally change the name of the interface and add a description.
5. Optionally select **Edit operations** if you want to change the default operation name. Then, in the Edit Operation Names window, type a new name and optional description, and click **OK**.
6. Select the **Deploy connector with module** check box.
7. In the **J2C Authentication Data Entry** field, enter the name you specified in the Security section of the administrative console.
8. Click **Use discovered connection properties** to set properties at this time.

   If you select **Use connection properties specified on server**, you can configure properties later, using the WebSphere Process Server or WebSphere Enterprise Service Bus administrative console.
9. Set or change any managed connection factory properties that apply to your configuration.

   See “Managed (J2C) connection factory properties” on page 252 for more information about these properties.

   Properties marked with an asterisk (*) are required.
10. Set or change any resource adapter properties that apply to your configuration.

See “Resource adapter properties” on page 251 for more information about these properties.

Properties marked with an asterisk (*) are required.

11. Click **Finish**.

Result

The new module is added to the Business Integration perspective.

What to do next

Export the module as an EAR file for deployment.

Configuring the adapter for ALE outbound processing

To configure the adapter for ALE outbound processing, you use the enterprise service discovery wizard to find an IDoc. You then configure the business objects that are generated and create a deployable module.

Setting connection properties for enterprise service discovery

To set connection properties for the enterprise service discovery wizard so that it can access the SAP server, specify such information as the user name and password you use to access the server as well as the name or IP address of the server.

Before you begin

Make sure you have successfully added the external dependencies.

About this task

Specify the connection properties that the enterprise service discovery wizard needs to connect to the SAP server and discover its business objects and services.

To specify the connection properties, use the following procedure.

How to perform this task

1. In WebSphere Integration Developer, start the enterprise service discovery wizard.
   a. Switch to the Business Integration Perspective by clicking **Window** → **Open Perspective** → **Other**. Then click **Business Integration (default)** and click **OK**.
   b. Right-click **New** → **Enterprise Service Discovery**.

      If **Enterprise Service Discovery** is not displayed, click **New** → **Other**, expand **Business Integration**, and click **Enterprise Service Discovery**. Then click **Next**.

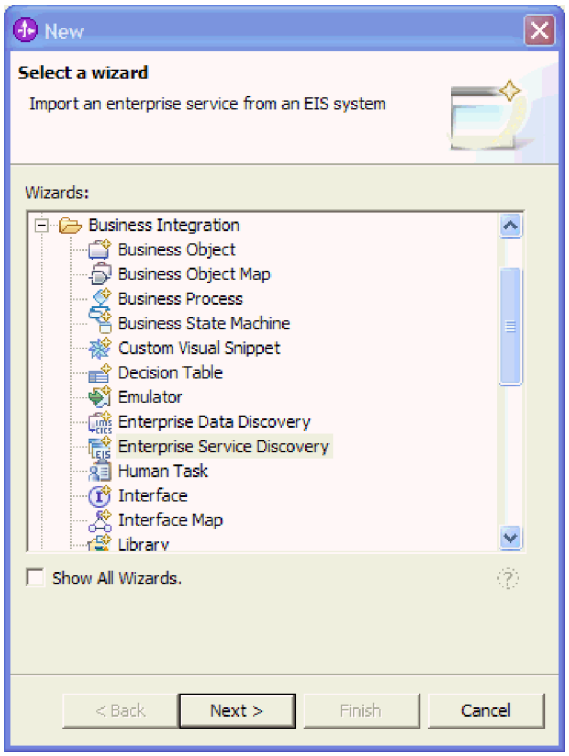


Figure 40. The expanded list of wizards

2. Select **IBM WebSphere Adapter for SAP Software** and click **Next**.

   If you previously ran the enterprise service discovery wizard, your connection properties appear when you expand the adapter name node by clicking the plus symbol (+). You can select the saved connection properties if you plan to connect to the SAP application you used the last time you ran the enterprise service discovery wizard.

   Note: Properties marked with an asterisk (*) are required.
3. Specify the configuration properties to initialize the discovery agent:
   a. Type the name and password you use to access the SAP system.

      The password is case-sensitive.
   b. Type your client ID.
   c. Optionally change the default settings for **Language**, **Codepage Number**, and **System Number**.
   d. In the **Application Server Host** field, type the name (or IP address) of your SAP server.

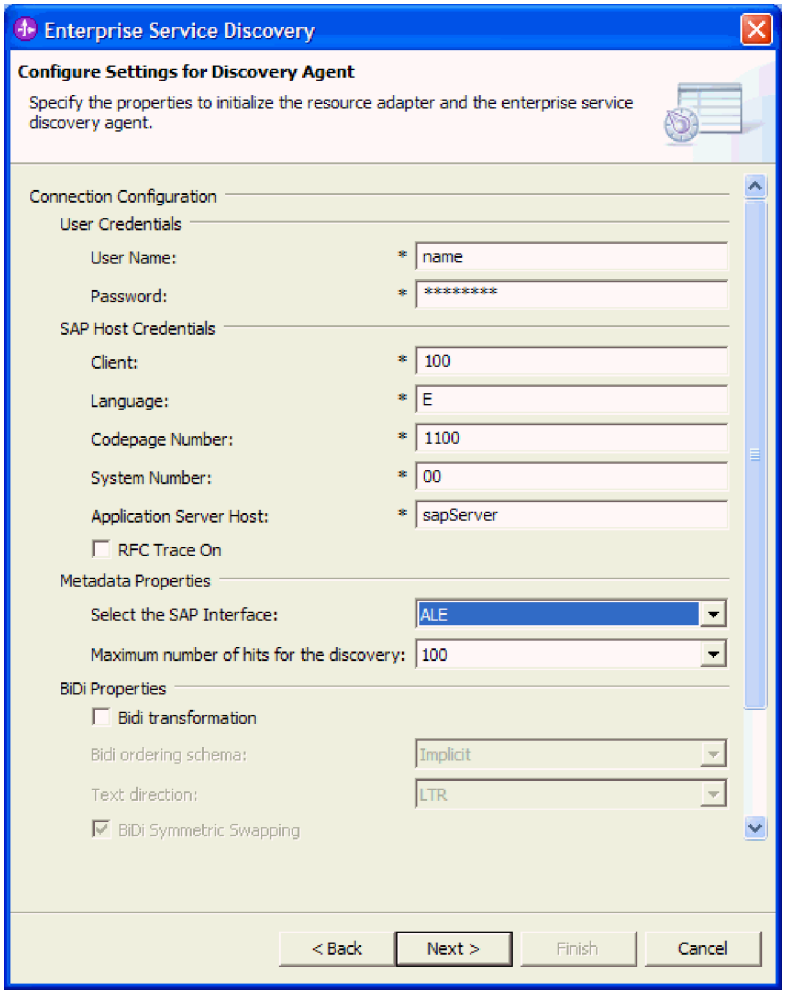


Figure 41. The Configure Settings for Discovery Agent window

4. **Optional:** For inbound processing, select the **RFC Trace On** check box if you want to generate a text file that details the RFC activity for event listeners. Use these text files in a development environment only, because the fields can grow rapidly.
5. Select **ALE** from the **Select the SAP interface** list.
6. **Optional:** Indicate the number of IDocs you want returned by changing the value in the **Maximum number of hits for the discovery** field or by accepting the default value.
7. **Optional:** If you need to set bidirectional properties, perform the following steps:
   a. Select **Bidi transformation**.
   b. Set properties for your environment. See "Enterprise service discovery connection properties" on page 249 for more information about these properties.
8. **Optional:** To change the logging level for enterprise service discovery, 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.

Note: This log pertains to enterprise service discovery only, not to the operation of the adapter.

9. Click **Next**.

Result

The enterprise service discovery wizard contacts the SAP server, using the information you provided (such as user name and password) to log in. You see the Find and Discover Enterprise Services window.

What to do next

Specify search criteria that the enterprise service discovery wizard uses to discover ALE functions on the SAP server.

Selecting business objects and services

To specify the IDoc you want to process, you provide information in the enterprise service discovery wizard.

Before you begin

Make sure you have set the connection properties for enterprise service discovery.

About this task

Specify search criteria that the enterprise service discovery wizard uses to discover ALE IDocs on the SAP server. The enterprise service discovery wizard returns a list of IDocs that meet the search criteria.

To specify the search criteria and select one or more IDocs, use the following procedure.

How to perform this task

1. In the Find and Discover Enterprise Services window, click **Execute Query**.
2. Expand **ALE**.

   The objects discovered by the query are grouped into two categories: Basic IDocs and Extension IDocs.
3. Expand either **Basic IDocs** or **Extension IDocs**, depending on the type of IDoc you want to import.
4. Click either **Discover By Name** or **Discover By Description**.

   The **Filter** button is now enabled.
5. Click **Filter**.
6. Type a search string (for example, `ALEREQ*`) representing the IDoc you want to call.
7. Expand **Discover By Name** or **Discover By Description** (whichever choice you made in the earlier step).
8. Click the IDoc you want to use. If you are working with multiple IDocs, click the names of all the IDocs.
9. Click **Add to import list**.

10. In the Configuration Parameters window, perform the following tasks to add the IDoc to the list of business objects to be imported.
    a. Select the **Use SAP Field Name to generate attribute(s)** check box.
    b. In the **Enter the Release** field, specify the SAP release number to identify the IDoc type you want the enterprise service discovery wizard to use for creating business objects.

       Note: You can specify an earlier release than the one you are currently using if you need to create business objects based on earlier versions of the IDoc type. If the earlier version of the IDoc type has fewer segments than the current version, the enterprise service discovery wizard might create a definition with missing segments or it might display an error indicating that the generation of the business object definition was unsuccessful because different versions of SAP require different API calls.
    c. Click **OK**.
11. Click **Next**.

Result

The enterprise service discovery wizard has returned a list of the IDoc or IDocs that match the search criteria, and you have selected the IDoc or IDocs you want to work with.

What to do next

Specify a name for the business object and the directory to which it should be stored.

Configuring the selected objects

To configure the business object, you specify information about the object (such as the name of the object and the operation associated with the object).

Before you begin

Make sure you have selected and imported the ALE IDoc.

About this task

Configure the business object that was imported. Name the object, indicate where the object should be stored, and specify the operation associated with the object.

To configure the business object, use the following procedure.

How to perform this task

1. In the Configure Objects window, type the name of the object location (where the object is stored).
2. In the **Namespace** field, use the default namespace (http://www.ibm.com/xmlns/prod/websphere/j2ca/sap) except in the following circumstance. If you are adding the business object to an existing module and the module already includes that business object (for example, if you previously ran the enterprise service discovery), change the namespace value.

   For example, you could change the namespace to http://www.ibm.com/xmlns/prod/websphere/j2ca/sap1.

3. Indicate that you are configuring the object for outbound processing by selecting **Outbound** from the **ServiceType** field.
4. Associate the Execute operation with the object by clicking **Add**, selecting **Execute**, and clicking **OK**.
5. Click **Next**.

Result

You have selected a location where the object is stored, optionally changed the namespace, indicated that this is an outbound operation, and associated the Execute operation with the object. The Generate Artifacts window is displayed.

What to do next

Generate a deployable module that includes the adapter and the business object.

Generating artifacts

To generate the module, which is the artifact that is deployed on WebSphere Process Server or WebSphere Enterprise Service Bus, you create the module, include the adapter in the module, and specify an alias used to authenticate the caller to the SAP server.

Before you begin

Make sure you have configured the business object. The Generate Artifacts window should be displayed.

About this task

Generate the module, which includes the adapter and configured business object. The module is the artifact you deploy on the server.

To generate the module, use the following procedure.

How to perform this task

1. In the Generate Artifacts window, create a new module.
   a. Click **New**.
   b. Click **Create a module project** and click **Next**.
   c. Type a name for the module.

      As you type the name, it is added to the workplace specified in the **Directory** field.

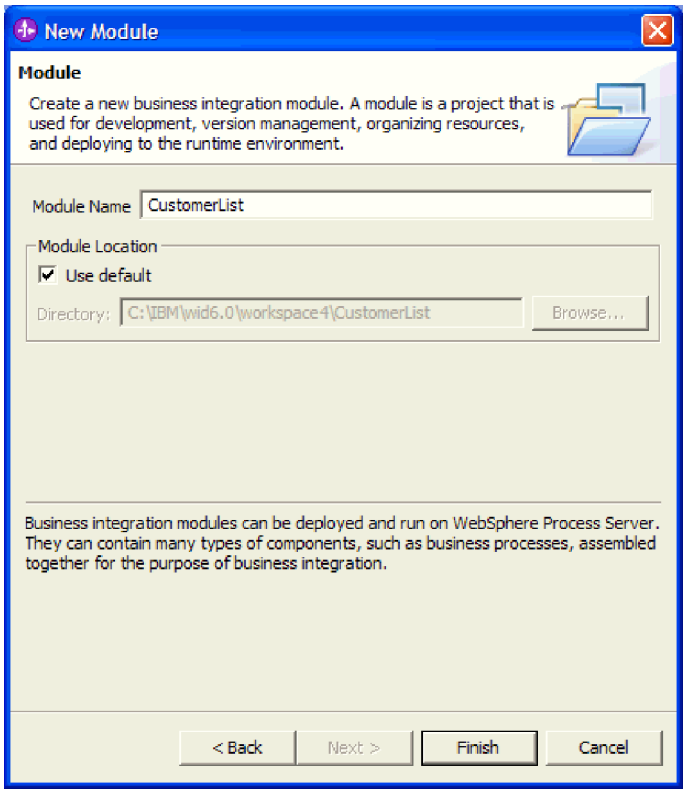


Figure 42. The New Module window

d. Click **Finish**.

2. If you want to change the default namespace, clear the **Use Default Namespace** check box and type a new path in the **Namespace** field.
3. In the Generate Artifacts window, specify the folder within the module where the service description should be saved by typing a name in the **Folder** field or browsing for a folder.
4. Optionally change the name of the interface and add a description.
5. Optionally select **Edit operations** if you want to change the default operation name. Then, in the Edit Operation Names window, type a new name and optional description, and click **OK**.
6. Select the **Deploy connector with module** check box.
7. In the **J2C Authentication Data Entry** field, enter the name you specified in the Security section of the administrative console.
8. Click **Use discovered connection properties** to set properties at this time.

   If you select **Use connection properties specified on server**, you can configure properties later, using the WebSphere Process Server or WebSphere Enterprise Service Bus administrative console.
9. Set or change any managed connection factory properties that apply to your configuration.

   See “Managed (J2C) connection factory properties” on page 252 for more information about these properties.

   Properties marked with an asterisk (*) are required.
10. Set or change any resource adapter properties that apply to your configuration.

See “Resource adapter properties” on page 251 for more information about these properties.

Properties marked with an asterisk (*) are required.

11. Click **Finish**.

Result

The new module is added to the Business Integration perspective.

What to do next

Export the module as an EAR file for deployment.

Configuring the adapter for ALE inbound processing

To configure the adapter for ALE inbound processing, you create a data source to hold the event-recovery table. You then use the enterprise service discovery wizard to find an IDoc, configure the business objects that are generated, and create a deployable module.

Configuring the data source

To create a data source, which is used for event tracking and recovery during inbound processing, you use the administrative console. You select a JDBC provider and then create a data source in the JDBC provider.

1. In the administrative console, select a JDBC provider.
   a. Click **Resources** → **JDBC Providers**.
   b. Select a JDBC provider.

      The example administrative console windows shown in this topic use the Cloudscape JDBC provider.
2. Select **Data sources**.

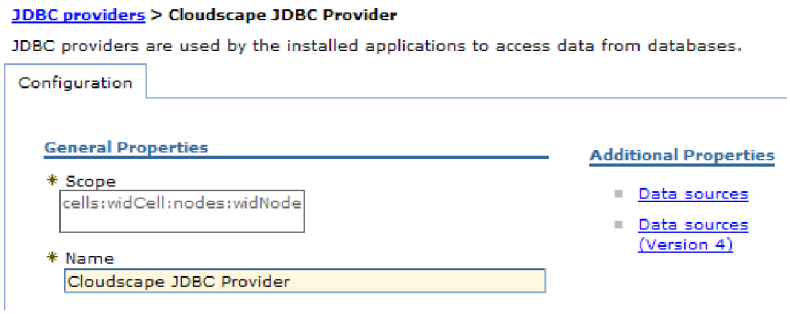


Figure 43. The Cloudscape JDBC provider Configuration tab

3. Create a new data source by clicking **New**.
4. Type values for the required fields.

Option	Description
Name	The name of the event table. Example: `ALEEventRecoveryDS`
JNDI Name	The JNDI name of the data source configured for event recovery. Example: `jdbc/ALEEventRecovery`

Option	Description
Database name	Example: `ALEEventRecoveryDB`

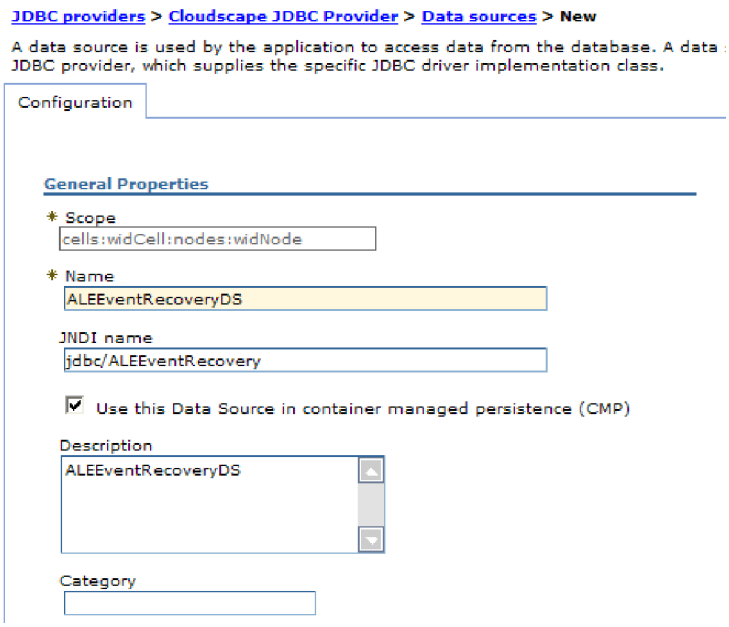


Figure 44. Creating the data source

5. Click **Apply**. After the changes are applied, **Custom properties** becomes active.
6. Click **Custom properties**.
7. Scroll down and click **createDatabase**.

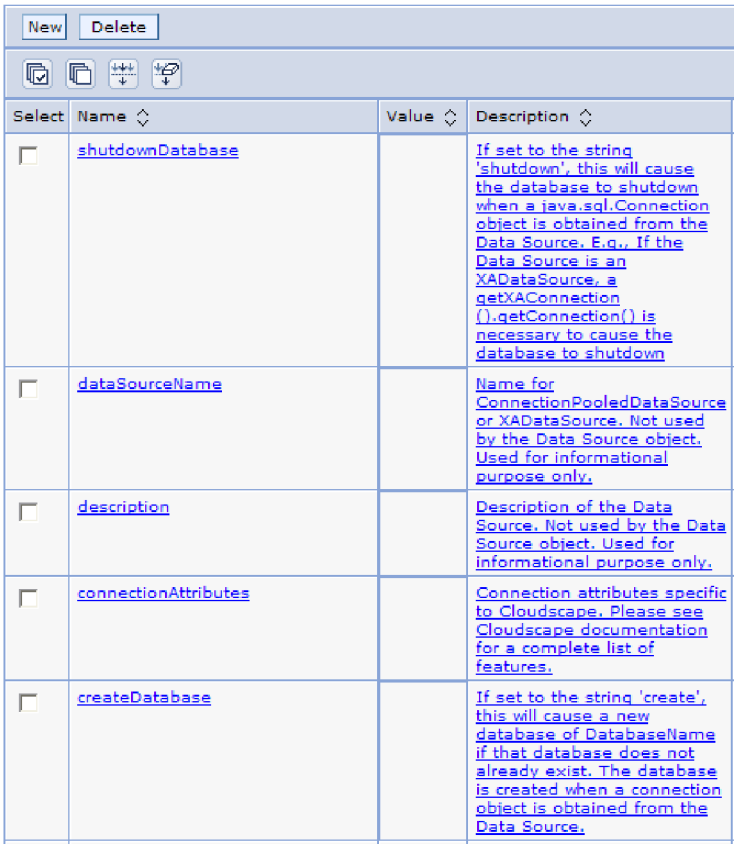


Figure 45. Selecting the createDatabase entry

8. Type `create` in the **value** field, and click **Apply**.
9. Save your configurations.

Result

A new data source is displayed in the list of data sources.

Setting connection properties for enterprise service discovery

To set connection properties for the enterprise service discovery wizard so that it can access the SAP server, specify such information as the user name and password you use to access the server as well as the name or IP address of the server.

Before you begin

Make sure you have successfully added the external dependencies.

About this task

Specify the connection properties that the enterprise service discovery wizard needs to connect to the SAP server and discover its business objects and services.

To specify the connection properties, use the following procedure.

How to perform this task

1. In WebSphere Integration Developer, start the enterprise service discovery wizard.
   a. Switch to the Business Integration Perspective by clicking **Window** → **Open Perspective** → **Other**. Then click **Business Integration (default)** and click **OK**.
   b. Right-click **New** → **Enterprise Service Discovery**.

      If **Enterprise Service Discovery** is not displayed, click **New** → **Other**, expand **Business Integration**, and click **Enterprise Service Discovery**. Then click **Next**.

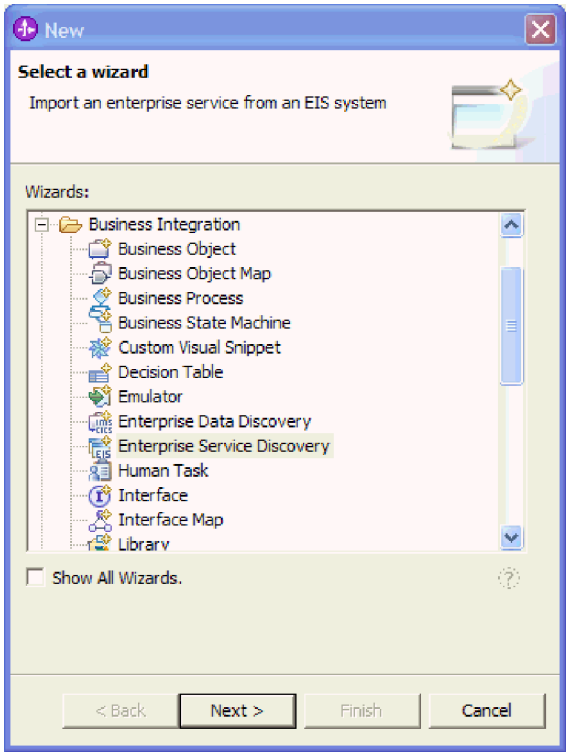


Figure 46. The expanded list of wizards

2. Select **IBM WebSphere Adapter for SAP Software** and click **Next**.

   If you previously ran the enterprise service discovery wizard, your connection properties appear when you expand the adapter name node by clicking the plus symbol (+). You can select the saved connection properties if you plan to connect to the SAP application you used the last time you ran the enterprise service discovery wizard.

   Note: Properties marked with an asterisk (*) are required.
3. Specify the configuration properties to initialize the discovery agent:
   a. Type the name and password you use to access the SAP system.

      The password is case-sensitive.
   b. Type your client ID.
   c. Optionally change the default settings for **Language**, **Codepage Number**, and **System Number**.
   d. In the **Application Server Host** field, type the name (or IP address) of your SAP server.

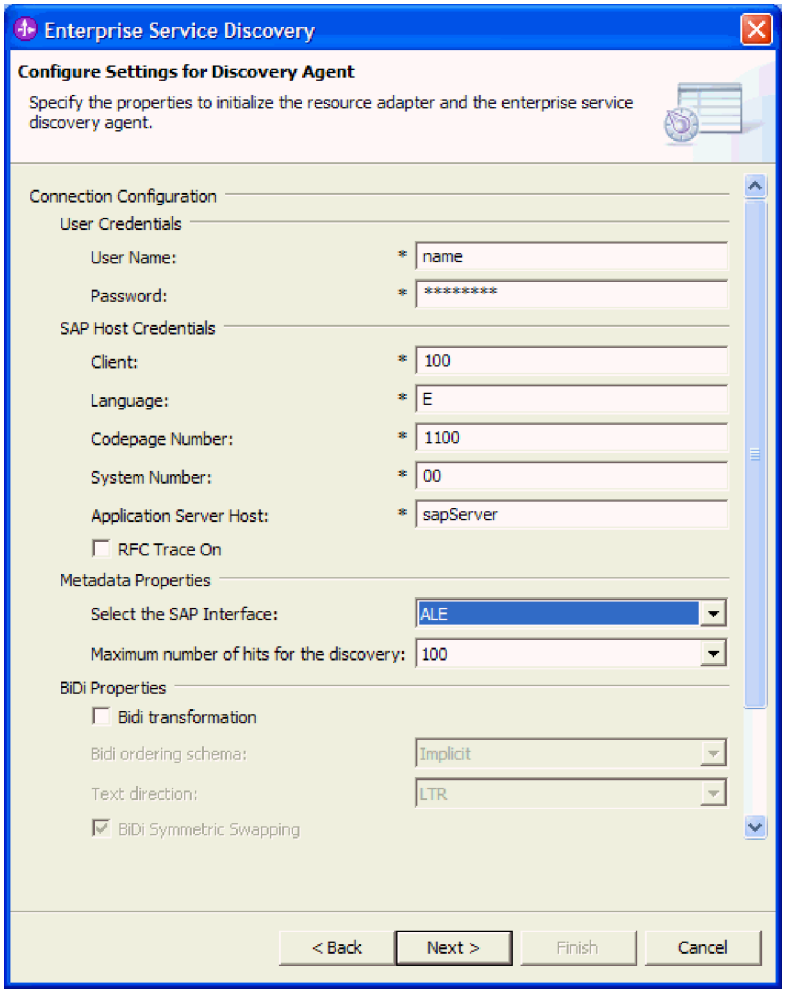


Figure 47. The Configure Settings for Discovery Agent window

4. **Optional:** For inbound processing, select the **RFC Trace On** check box if you want to generate a text file that details the RFC activity for event listeners. Use these text files in a development environment only, because the fields can grow rapidly.
5. Select **ALE** from the **Select the SAP interface** list.
6. **Optional:** Indicate the number of IDocs you want returned by changing the value in the **Maximum number of hits for the discovery** field or by accepting the default value.
7. **Optional:** If you need to set bidirectional properties, perform the following steps:
   a. Select **Bidi transformation**.
   b. Set properties for your environment. See "Enterprise service discovery connection properties" on page 249 for more information about these properties.
8. **Optional:** To change the logging level for enterprise service discovery, 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.

Note: This log pertains to enterprise service discovery only, not to the operation of the adapter.

9. Click **Next**.

Result

The enterprise service discovery wizard contacts the SAP server, using the information you provided (such as user name and password) to log in. You see the Find and Discover Enterprise Services window.

What to do next

Specify search criteria that the enterprise service discovery wizard uses to discover ALE functions on the SAP server.

Selecting business objects and services

To specify which IDoc you want to process, you provide information in the enterprise service discovery wizard.

Before you begin

Make sure you have set the connection properties for enterprise service discovery.

About this task

Specify search criteria that the enterprise service discovery wizard uses to discover ALE IDocs on the SAP server. The enterprise service discovery wizard returns a list of ALE IDocs that meet the search criteria.

To specify the search criteria and select one or more IDocs, use the following procedure.

How to perform this task

1. In the Find and Discover Enterprise Services window, click **Execute Query**.
2. Expand **ALE**.

   The objects discovered by the query are grouped into two categories: Basic IDocs and Extension IDocs.
3. Expand either **Basic IDocs** or **Extension IDocs**, depending on the type of IDoc you want to import.
4. Click either **Discover By Name** or **Discover By Description**.

   The **Filter** button is now enabled.
5. Click **Filter**.
6. Type a search string (for example, `ALEREQ*`) representing the IDoc you want to call.
7. Expand **Discover By Name** or **Discover By Description** (whichever choice you made in the earlier step).
8. Click the IDoc you want to use. If you are working with multiple IDocs, click the names of all the IDocs.
9. Click **Add to import list**.

10. In the Configuration Parameters window, perform the following tasks to add the IDoc to the list of business objects to be imported.
    a. If you want to send an IDoc packet and specify that the packet not be split, select the **Send an IDoc Packet as one Business Object** check box.
    b. If you want to send the IDoc in an unparsed form (so that the client application, rather than the adapter, parses the data), select the **Send IDoc with Unparsed Data** check box.
    c. Select the **Use SAP Field Name to generate attribute(s)** check box.
    d. In the **Enter the Release** field, specify the SAP release number to identify the IDoc type you want the enterprise service discovery wizard to use for creating business objects.

       Note: You can specify an earlier release than the one you are currently using if you need to create business objects based on earlier versions of the IDoc type. If the earlier version of the IDoc type has fewer segments than the current version, the enterprise service discovery wizard might create a definition with missing segments or it might display an error indicating that the generation of the business object definition was unsuccessful because different versions of SAP require different API calls.
    e. Click **OK**.
11. Click **Next**.

Result

The enterprise service discovery wizard has returned a list of the function or functions that match the search criteria, and you have selected the function or functions you want to work with.

What to do next

Specify a name for the business object and the directory to which it should be stored.

Configuring the selected objects

To configure the business object, you specify information about the object (such as the name of the object and the operation associated with the object).

Before you begin

Make sure you have selected and imported the ALE IDoc.

About this task

Configure the business object that was imported. Name the object, indicate where the object should be stored, and specify the operation associated with the object.

To configure the business object, use the following procedure.

How to perform this task

1. In the Configure Objects window, type the name of the object location (where the object is stored).
2. In the **Namespace** field, use the default namespace (http://www.ibm.com/xmlns/prod/websphere/j2ca/sap) except in the following circumstance. If you

are adding the business object to an existing module and the module already includes that business object (for example, if you previously ran the enterprise service discovery), change the namespace value.

For example, you could change the namespace to http://www.ibm.com/xmlns/prod/websphere/j2ca/sap1.

3. Make sure that the default value of **Inbound** is selected in the **ServiceType** field.
4. Indicate which operation you want performed on the IDoc.
   a. Click **Add**.
   b. Select **Create**, **Updatewithdelete**, or **Delete**.
   c. Click **OK**.
5. Click **Next**.

Result

You have selected a location where the object is stored, optionally changed the namespace, indicated that this is an inbound operation, and associated an operation with the object. The Generate Artifacts window is displayed.

What to do next

Generate a deployable module that includes the adapter and the business object.

Generating artifacts

To generate the module, which is the artifact that is deployed on WebSphere Process Server or WebSphere Enterprise Service Bus, you create the module, include the adapter in the module, and specify an alias used to authenticate the caller to the SAP server.

Before you begin

Make sure you have configured the business object. The Generate Artifacts window should be displayed.

About this task

Generate the module, which includes the adapter and configured business object. The module is the artifact you deploy on the server.

To generate the module, use the following procedure.

How to perform this task

1. In the Generate Artifacts window, create a new module.
   a. Click **New**.
   b. Click **Create a module project** and click **Next**.
   c. Type a name for the module.

      As you type the name, it is added to the workplace specified in the **Directory** field.

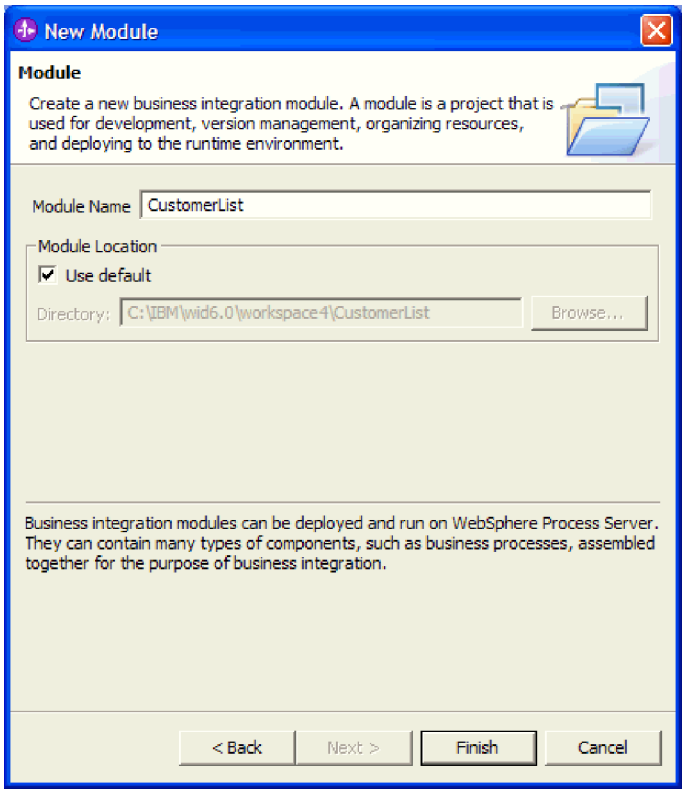


Figure 48. The New Module window

d. Click **Finish**.

2. If you want to change the default namespace, clear the **Use Default Namespace** check box and type a new path in the **Namespace** field.
3. In the Generate Artifacts window, specify the folder within the module where the service description should be saved by typing a name in the **Folder** field or browsing for a folder.
4. Optionally change the name of the interface and add a description.
5. Optionally select **Edit operations** if you want to change the default operation name. Then, in the Edit Operation Names window, type a new name and optional description, and click **OK**.
6. Select the **Deploy connector with module** check box.
7. Click **Use discovered connection properties** to set properties at this time.

   If you select **Use connection properties specified on server**, you can configure properties later, using the WebSphere Process Server or WebSphere Enterprise Service Bus administrative console.
8. In the **J2C Authentication Data Entry** field, enter the name you specified in the Security section of the administrative console.
9. Set the required inbound connection properties.

   As shown in the following figure, required properties are indicated with an asterisk (*).

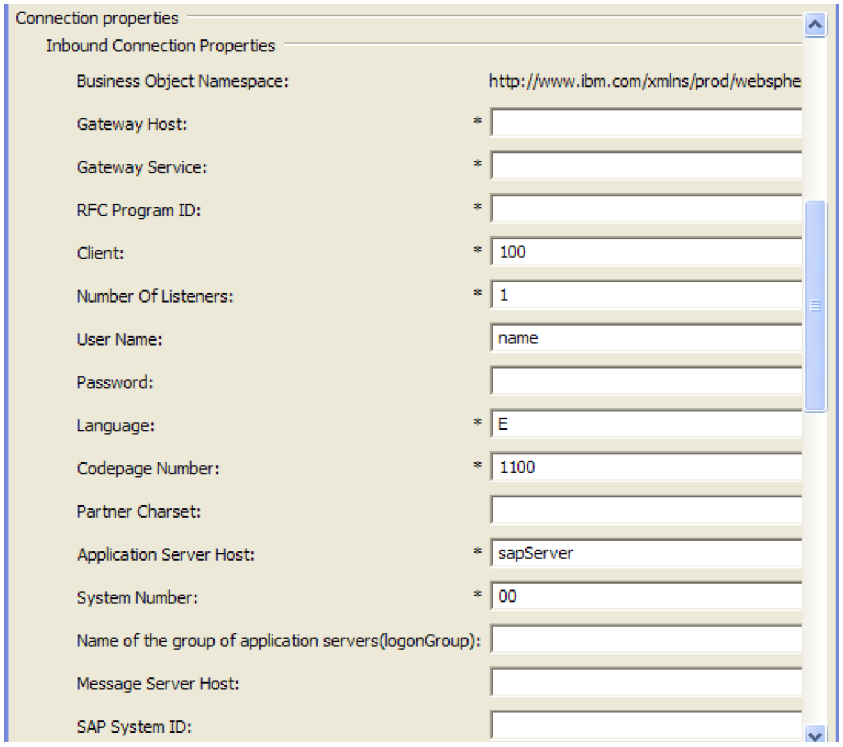


Figure 49. Connection properties

The fields are described in the following table.

Option	Description
Gateway Host	Specify the SAP gateway host where the gateway service is running.
Gateway Service	Specify the gateway server identifier. This value is often `sapgw00`.
RFC Program ID	Specify the program identifier under which the RFC server program registers.
Client	Use the value already filled in for your client number, or change it (if necessary).
Number of Listeners	Use the value already filled in for the number of listeners, or change it (if necessary).
Language	Use the value already filled in for language, or change it (if necessary).
Codepage Number	Use the value already filled in for codepage number, or change it (if necessary).
Application Server Host	Use the value already filled in for your application server host, or change it (if necessary).
System Number	Use the value already filled in for system number, or change it (if necessary).
Auto Create Event Table	Select this check box if you want the event table automatically created. If you do not select the check box, you must create the event table manually.

Option	Description
Event Recovery Table Name	Type the name you specified when you created the data source.
Event Recovery DataSource (JNDI) Name	Type the name you specified when you created the data source.
Username to connect to Event Datasource	Type the name you use to access the data source.
Password to connect to Event Datasource	Type the password you use to access the data source.

10. Optionally, change or set any of the other inbound connection properties.
    See "Activation specification properties" on page 256 for more information about these properties.
11. Set or change any resource adapter properties that apply to your configuration.
    See "Resource adapter properties" on page 251 for more information about these properties.
    Properties marked with an asterisk (*) are required.

Result

The new module is added to the Business Integration perspective.

What to do next

Generate reference bindings..

Generating reference bindings

To notify the adapter of eligible event listeners, you generate reference bindings. For inbound processing, the adapter uses these event listeners to receive events from SAP before forwarding them to the endpoint (a message driven bean).

Before you begin

Make sure you have completed enterprise service discovery.

How to perform this task

1. In the Business Integration Perspective of WebSphere Integration Developer, right-click the module, and select **Open With → Assembly Editor**.
2. In the Assembly Diagram window, create a new component by clicking the top icon in the left pane and then clicking the top icon in the resulting menu (it has hover help that reads **Component (with no implementation type)**.

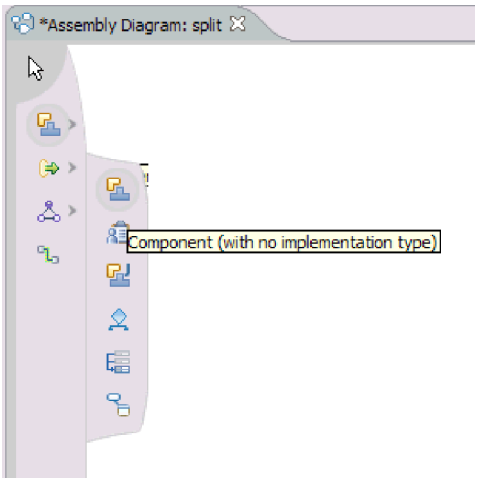


Figure 50. Selecting the new component icon

The cursor changes to the placement icon.

3. Click the palette to add the new component to the Assembly Diagram window.
4. Click and drag the Export component to the new component.

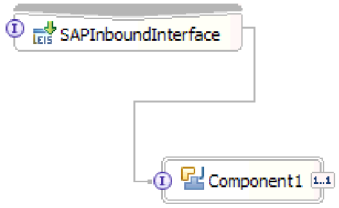


Figure 51. Wiring the components

5. In the Add Wire window, click **OK**.
6. Create a Java component to act as an endpoint by right-clicking the new component and selecting **Generate Implementation → Java**.
7. In the Generate Implementation window, select the package in which the Java code will be created, and click **OK**.
8. In the Java file editor, make any desired changes to the Java file. For example, you might wish to write code to print trace and log messages.
9. Save the Java file.

Result

A new reference binding has been generated.

What to do next

Export the module as an EAR file for deployment.

Configuring the adapter for SQI processing

To configure the adapter for SQI outbound processing, you use the enterprise service discovery wizard to find data in an SAP table. You then configure the business objects that are generated and create a deployable module.

Setting connection properties for enterprise service discovery

To set connection properties for the enterprise service discovery wizard so that it can access the SAP server, specify such information as the user name and password you use to access the server as well as the name or IP address of the server.

Before you begin

Make sure you have successfully added the external dependencies.

About this task

Specify the connection properties that the enterprise service discovery wizard needs to connect to the SAP server and discover its business objects and services.

To specify the connection properties, use the following procedure.

How to perform this task

1. In WebSphere Integration Developer, start the enterprise service discovery wizard.
   a. Switch to the Business Integration Perspective by clicking **Window** → **Open Perspective** → **Other**. Then click **Business Integration (default)** and click **OK**.
   b. Right-click **New** → **Enterprise Service Discovery**.

      If **Enterprise Service Discovery** is not displayed, click **New** → **Other**, expand **Business Integration**, and click **Enterprise Service Discovery**. Then click **Next**.

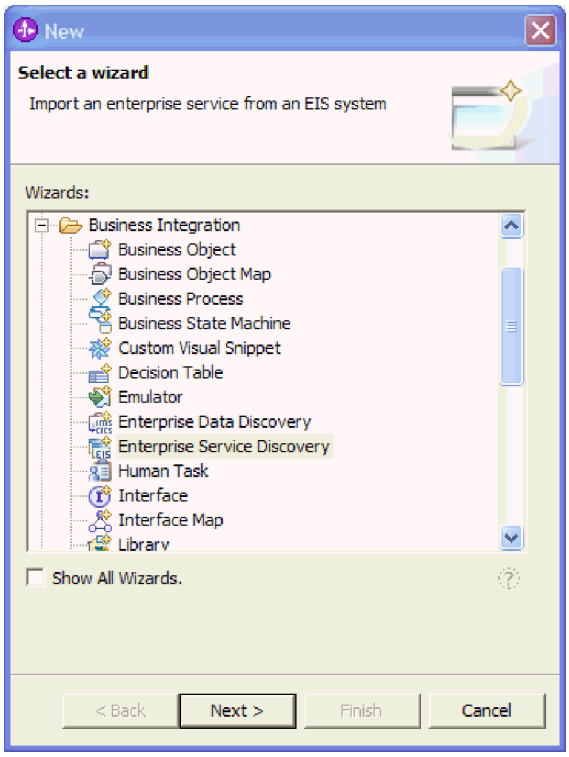


Figure 52. The expanded list of wizards

2. Select **IBM WebSphere Adapter for SAP Software** and click **Next**.

   If you previously ran the enterprise service discovery wizard, your connection properties appear when you expand the adapter name node by clicking the plus symbol (+). You can select the saved connection properties if you plan to connect to the SAP application you used the last time you ran the enterprise service discovery wizard.

   Note: Properties marked with an asterisk (*) are required.
3. Specify the configuration properties to initialize the discovery agent:
   a. Type the name and password you use to access the SAP system.

      The password is case-sensitive.
   b. Type your client ID.
   c. Optionally change the default settings for **Language**, **Codepage Number**, and **System Number**.
   d. In the **Application Server Host** field, type the name (or IP address) of your SAP server.

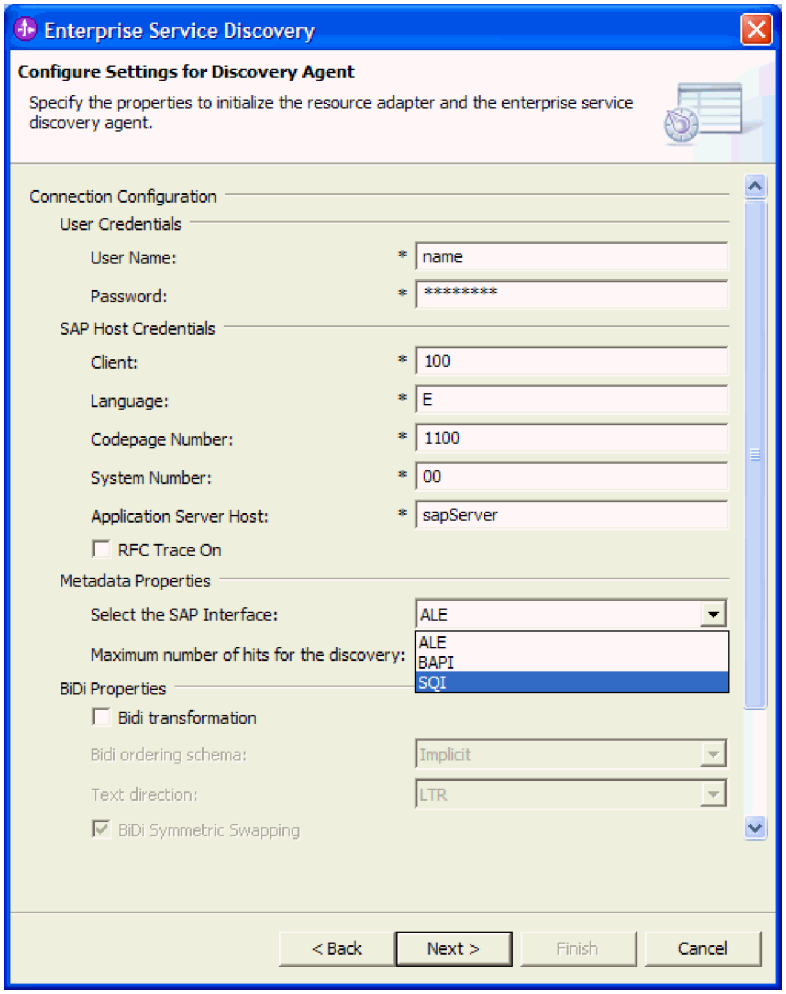


Figure 53. The Configure Settings for Discovery Agent window

4. Select **SQI** from the **Select the SAP interface** list.
5. **Optional:** Indicate the number of functions you want returned by changing the value in the **Maximum number of hits for the discovery** field or by accepting the default value.
6. **Optional:** If you need to set bidirectional properties, perform the following steps:
   a. Select **Bidi transformation**.
   b. Set properties for your environment. See "Enterprise service discovery connection properties" on page 249 for more information about these properties.
7. **Optional:** To change the logging level for enterprise service discovery, 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.

      Note: This log pertains to enterprise service discovery only, not to the operation of the adapter.

8. Click **Next**.

Result

The enterprise service discovery wizard contacts the SAP server, using the information you provided (such as user name and password) to log in. You see the Find and Discover Enterprise Services window.

What to do next

Specify search criteria that the enterprise service discovery wizard uses to query data on the SAP server.

Selecting business objects and services

To specify which data you want to query, you provide information in the enterprise service discovery wizard.

Before you begin

Make sure you have set the connection properties for enterprise service discovery.

About this task

Specify search criteria that the enterprise service discovery wizard uses to query data on the SAP server. The enterprise service discovery wizard returns the data that meet the search criteria.

To specify the search criteria, use the following procedure.

How to perform this task

1. From the Find and Discover Enterprise Services window, begin the process of searching for services by clicking **Execute Query**.
2. Expand **SQI**.
3. Click either **Discover By Name** or **Discover By Description**.
   The **Filter** button is now enabled.
4. Click **Filter**.
5. Enter information about the table.
   a. In the Filter Properties for Discover By Name or Filter Properties for Discover By Description window, type the name of the table.
      You can type part of the name and use the wildcard (*) character.
   b. Click **OK**.
   c. Expand **Discover By Name (filtered)** or **Discover By Description (filtered)**.

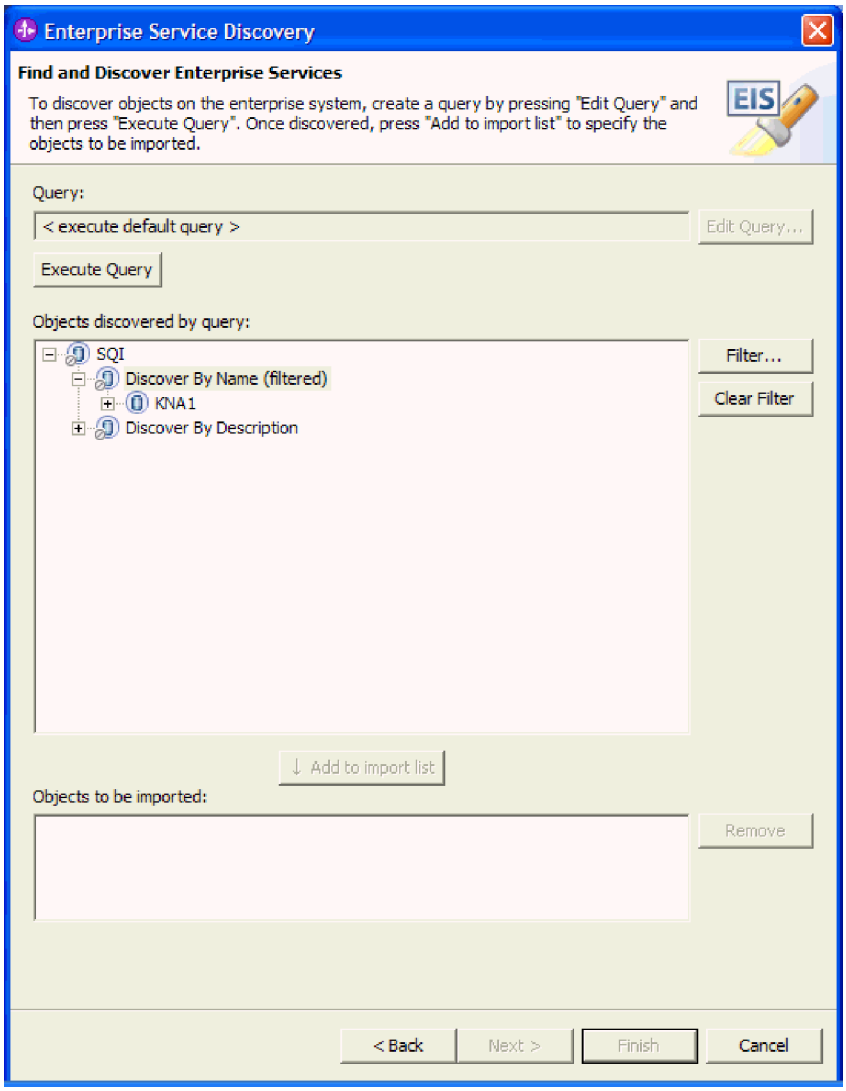


Figure 54. The Find and Discover Enterprise Services window

d. Select the table and click **Add to import list**.
   In the example shown in the previous figure, the table name is KNA1.
e. In the Configuration Parameters for *table* window, indicate which columns you want included in the query and click **OK**.

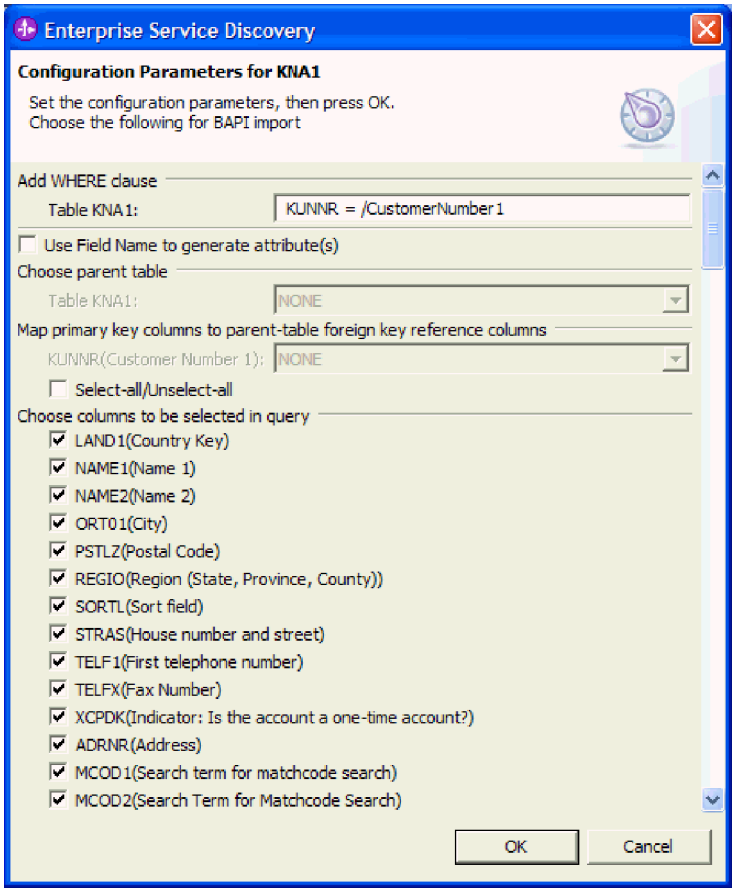


Figure 55. The Configuration Parameters window

6. To include another table in the query, perform the following tasks:
   a. Click **Filter** and type the name of the table (for example, ADRC). Then click **OK**.
   b. Expand **Discover By Name (filtered)** or **Discover By Description (filtered)** by clicking **+**.
   c. Select the table and click **Add**.
   d. Under **Choose Parent Table**, select the first table (KNA1 in the example).
   e. Under **Map primary key columns to parent-table foreign key reference columns**, select a value to link the tables.

      For example, you might select **ADRNR** for **ADDRNUMBER**.
   f. Indicate which columns you want included in the query and click **OK**.
7. Click **Next**.

Result

The enterprise service discovery wizard has returned the data that matches the search criteria.

What to do next

Specify a name for the business object and the directory to which it should be stored.

Configuring the selected objects

To configure the business object, you specify information about the object (such as the name of the object and the operation associated with the object).

Before you begin

Make sure you have selected and imported the business object.

About this task

Configure the business object that was imported. Indicate where the object should be stored, and specify the operation associated with the object.

To configure the business object, use the following procedure.

How to perform this task

1. In the Configure Objects window, type the name of the object location (where the object is stored).
2. In the **Namespace** field, use the default namespace (http://www.ibm.com/xmlns/prod/websphere/j2ca/sap) except in the following circumstance. If you are adding the business object to an existing module and the module already includes that business object (for example, if you previously ran the enterprise service discovery), change the namespace value.

   For example, you could change the namespace to http://www.ibm.com/xmlns/prod/websphere/j2ca/sap1.
3. Click **Next**.

Result

You have selected a location where the object is stored and optionally changed the namespace. The Generate Artifacts window is displayed.

What to do next

Generate a deployable module that includes the adapter and the business object.

Generating artifacts

To generate the module, which is the artifact that is deployed on WebSphere Process Server or WebSphere Enterprise Service Bus, you create the module, include the adapter in the module, and specify an alias used to authenticate the caller to the SAP server.

Before you begin

Make sure you have configured the business object. The Generate Artifacts window should be displayed.

About this task

Generate the module, which includes the adapter and configured business object. The module is the artifact you deploy on the server.

To generate the module, use the following procedure.

How to perform this task

1. In the Generate Artifacts window, create a new module.
   a. Click **New**.
   b. Click **Create a module project** and click **Next**.
   c. Type a name for the module.

      As you type the name, it is added to the workplace specified in the **Directory** field.

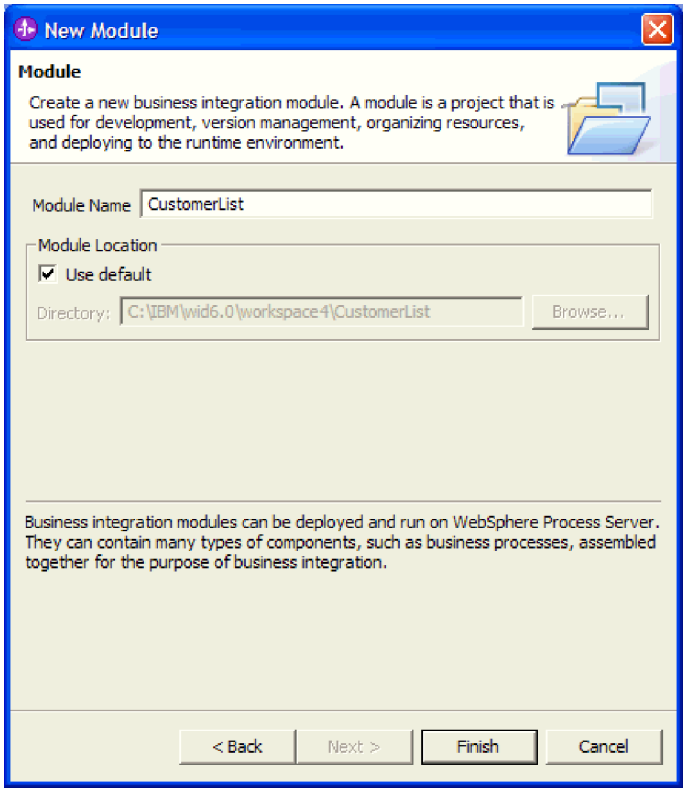


Figure 56. The New Module window

   d. Click **Finish**.
2. If you want to change the default namespace, clear the **Use Default Namespace** check box and type a new path in the **Namespace** field.
3. In the Generate Artifacts window, specify the folder within the module where the service description should be saved by typing a name in the **Folder** field or browsing for a folder.
4. Optionally change the name of the interface and add a description.
5. Optionally select **Edit operations** if you want to change the default operation name. Then, in the Edit Operation Names window, type a new name and optional description, and click **OK**.
6. Select the **Deploy connector with module** check box.
7. In the **J2C Authentication Data Entry** field, enter the name you specified in the Security section of the administrative console.
8. Click **Use discovered connection properties** to set properties at this time.

   If you select **Use connection properties specified on server**, you can configure properties later, using the WebSphere Process Server or WebSphere Enterprise Service Bus administrative console.

9. Set or change any managed connection factory properties that apply to your configuration.

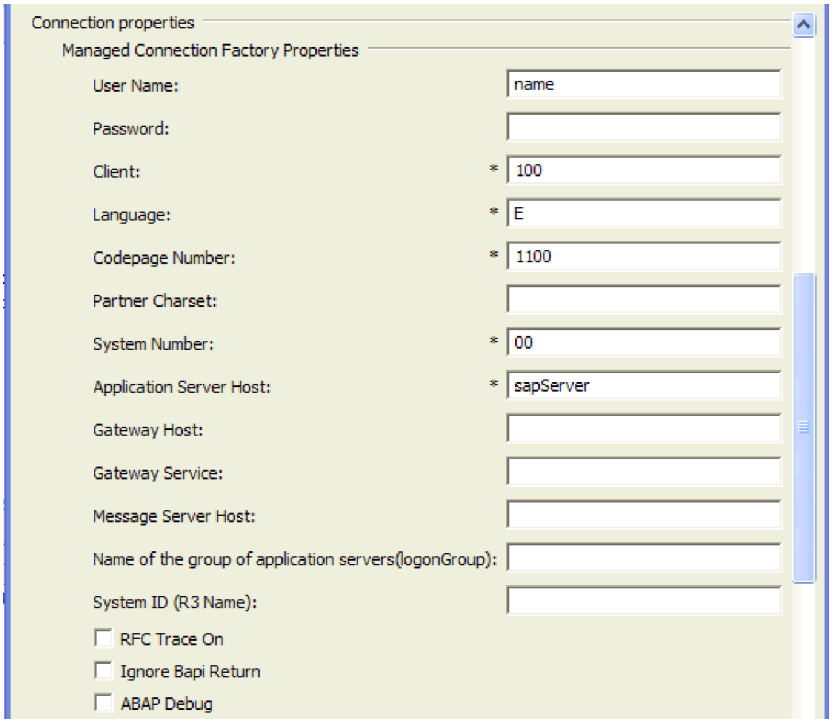


Figure 57. Managed connection factory properties

See "Managed (J2C) connection factory properties" on page 252 for more information about these properties.

Properties marked with an asterisk (*) are required.

10. Set or change any resource adapter properties that apply to your configuration.

See "Resource adapter properties" on page 251 for more information about these properties.

Properties marked with an asterisk (*) are required.

11. Click **Finish**.

Result

The new module is added to the Business Integration perspective.

What to do next

Export the module as an EAR file for deployment.

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 enterprise archive resource (EAR) file

To deploy the project, you must export it as an EAR file. The EAR file is created during the deployment process.

Before you begin

Make sure you have created a module. The module should be displayed in the Business Perspective.

About this task

Export the module as an EAR file so that it can be deployed to WebSphere Process Server or WebSphere Enterprise Service Bus. When you export the module as an EAR file, **App** is appended to the name, indicating that the module is a deployable application.

To export the module, use the following procedure.

How to perform this task

1. In the J2EE perspective window of WebSphere Integration Developer, right-click the application you want to export, and select **Export**.
2. Select **EAR file** from the Export window.
3. In the EAR Export window, select the EAR project and the destination directory (the directory, including the EAR file name, where the project should be exported).
4. If the Save Resources window appears, click **OK**.
5. Click **Next**.
6. Click **Finish**.

Result

The EAR file has been created and saved and is ready to be deployed.

What to do next

Install the application on WebSphere Process Server or WebSphere Enterprise Service Bus.

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

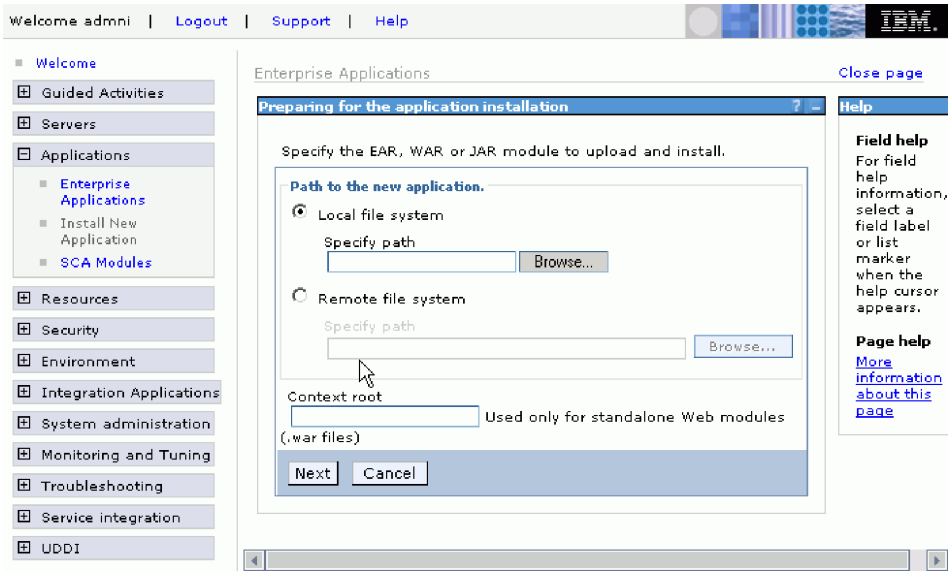
You must have exported your project module as an EAR file before installing the adapter project.

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.

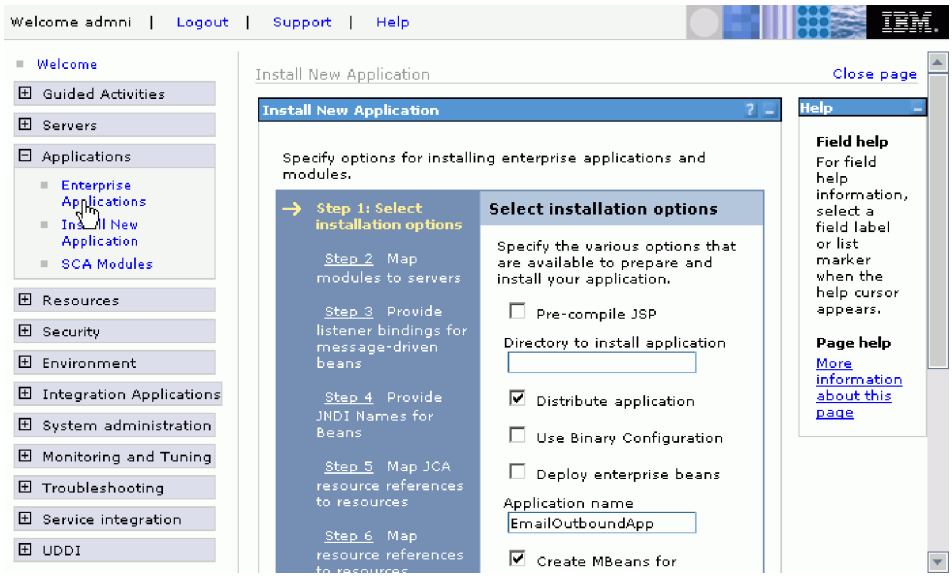
How to perform this task

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.
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 configuration properties from the administrative console

To set or change configuration properties after you deploy a module, you use the administrative console. You can update managed (J2C) connection factory properties (which are used for outbound processing) and activation specification properties (which are used for inbound processing).

Setting resource adapter properties

To set resource adapter properties for your adapter module after it has been deployed, use the administrative console. You select the name of the property you want to configure and then change or set the value as desired.

Before you begin

Your adapter module must be deployed on the WebSphere Process Server or WebSphere Enterprise Service Bus.

About this task

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

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

How to perform this task

1. Start the administrative console.
2. Under **Applications**, select **Enterprise Applications**.
3. From the Enterprise Applications list, click the name of the adapter application whose properties you want to change.
4. Scroll to the bottom of the window. Under **Related Items**, click **Connector Modules**.
5. Click **CYWAP_SapAdapter.rar** file.

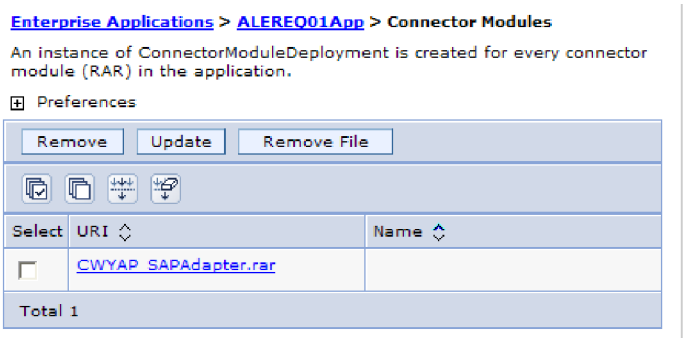


Figure 58. The RAR file in the Connector Modules window

6. Click **Resource Adapter**.

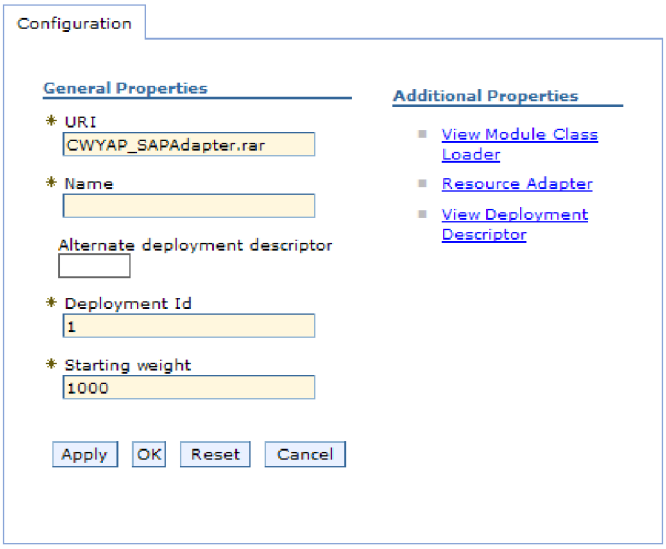


Figure 59. The Configuration tab for the RAR file

7. Click **Custom properties**.

Figure 60. Additional Properties selections

8. For each property you want to change, perform the following steps. See "Resource adapter properties" on page 251 for more information about these properties.
   a. Click the name of the property.
   b. Change the contents of the **Value** field value or type a value, if the field is empty.
   c. Click **OK**.
9. Click the **Save** link in the **Messages** box at the top of the window.

Result

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

Setting managed (J2C) connection factory properties

To set managed connection factory properties for your adapter module after it has been deployed, use the administrative console. You select the name of the property you want to configure and then change or set the value as desired.

Before you begin

Your adapter module must be deployed on the WebSphere Process Server or WebSphere Enterprise Service Bus.

About this task

You use managed connection factory properties to configure the target SAP server instance.

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

How to perform this task

1. Start the administrative console.
2. Under **Applications**, select **Enterprise Applications**.
3. From the Enterprise Applications list, click the name of the adapter application whose properties you want to change.
4. Scroll to the bottom of the window. Under **Related Items**, click **Connector Modules**.
5. Under **Additional Properties**, select **J2C connection factories**.
6. Click **CYWAP_SapAdapter.rar** file.

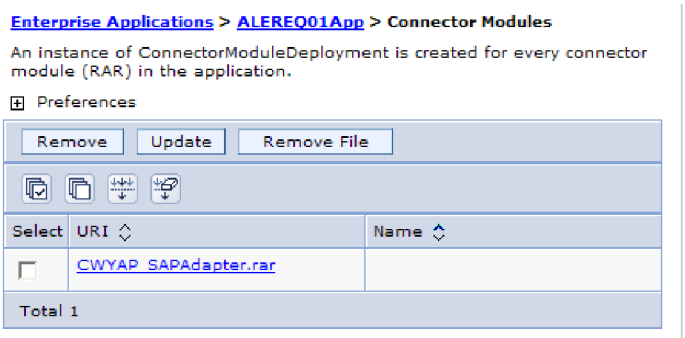


Figure 61. The RAR file in the Connector Modules window

7. Click **Resource Adapter**.

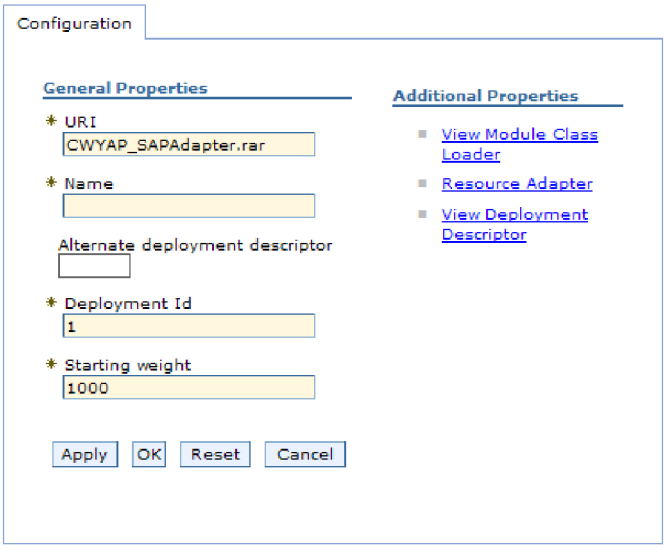


Figure 62. The Configuration tab for the RAR file

8. Click **J2C connection factories**.

Figure 63. Additional Properties selections

9. Click the name of the J2C connection factory you want to configure.

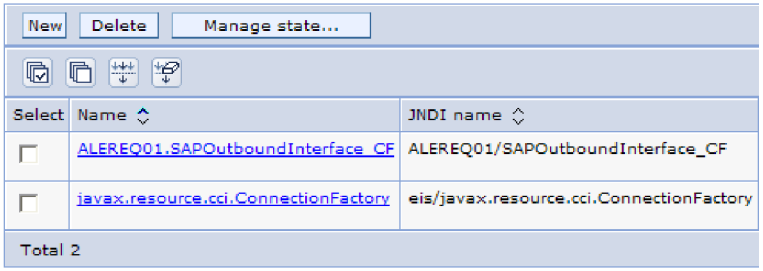


Figure 64. Selecting the connection factory

10. Click **Custom properties**.

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

11. For each property you want to change, perform the following steps. See "Managed (J2C) connection factory properties" on page 252 for more information about these properties.
    a. Click the name of the property.
    b. Change the contents of the **Value** field value or type a value, if the field is empty.
    c. Click **OK**.
12. Click the **Save** link in the **Messages** box at the top of the window.

Result

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

Setting activation specification properties

To set activation specification properties for your inbound adapter module after it has been deployed, use the administrative console. You select the name of the message endpoint property you want to configure and then change or set the value as desired.

Before you begin

Your adapter module must be deployed on the WebSphere Process Server or WebSphere Enterprise Service Bus.

About this task

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

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

How to perform this task

1. Start the administrative console.
2. Under **Applications**, select **Enterprise Applications**.
3. From the Enterprise Applications list, click the name of the adapter application whose properties you want to change.
4. Scroll to the bottom of the window. Under **Related Items**, click **Connector Modules**.
5. Click **CYWAP_SapAdapter.rar** file.

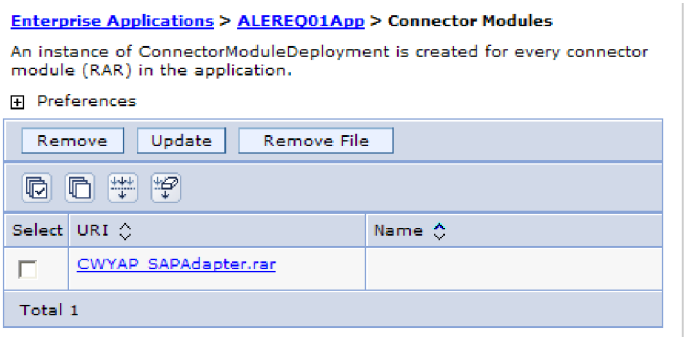


Figure 65. The RAR file in the Connector Modules window

6. Click **Resource Adapter**.

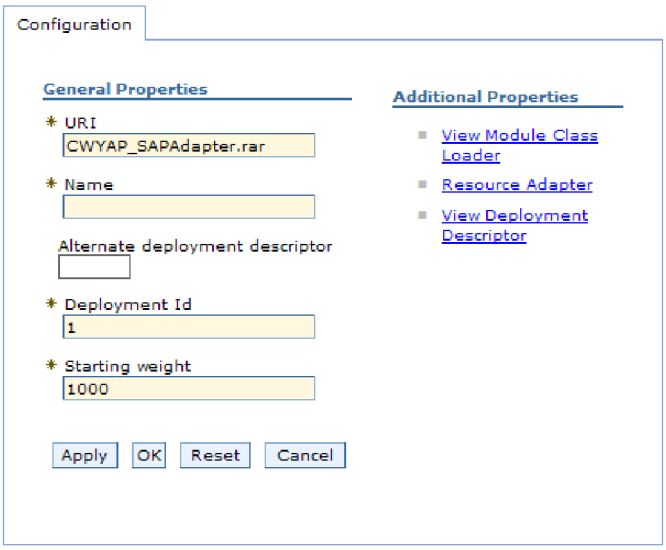


Figure 66. The Configuration tab for the RAR file

7. Click **J2C Activation specifications**.

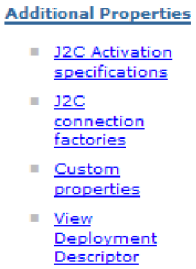


Figure 67. Additional Properties selections

8. Click the name of the adapter application you want to configure.

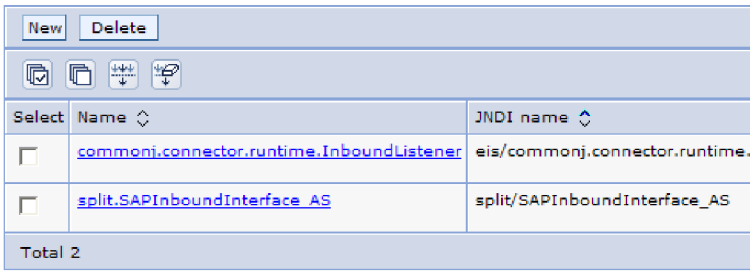


Figure 68. Selecting the application

9. Click **J2C activation specification custom properties**.
10. For each property you want to change, perform the following steps. See "Activation specification properties" on page 256 for more information about these properties.
    a. Click the name of the property.
    b. Change the contents of the **Value** field value or type a value, if the field is empty.
    c. Click **OK**.
11. Click the **Save** link in the **Messages** box at the top of the window.

Result

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

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.

Detecting errors during outbound processing

To detect errors such as invalid data or invalid state that occur during outbound processing, you set up business-object application-specific data.

Before you begin

Make sure you have determined which errors you want to detect.

About this task

During outbound processing, the adapter can automatically detect errors generated by the SAP Jco interface. To detect other types of errors returned by the RFC interface (for example, to be able to validate the data that is returned) you must define values for application-specific data (metadata) at the business-object level.

To set up the business-level metadata to detect errors, use the following procedure.

How to perform this task

1. Identify the parameters that define RFC error codes and their possible values.
2. Add the application-specific information for ErrorParameter, ErrorCode, and ErrorDetail to the business object.
   - ErrorParameter is the XPATH to the property that returns the error codes.
   - ErrorCode contains all possible values (for example, `E`, `ERROR`, and `NODATA`) returned in the property referred to by ErrorParameter.
   - ErrorDetail is the XPATH to the property that contains details about the error.

   If the values defined in the ErrorCode property match the error parameter values after RFC executes the call, an error message with detailed information is generated. The detail is derived from the ErrorDetail property.

   Error handling application-specific information must be manually maintained. You can use the Business Object Designer to add the application-specific information.

Result

Your top-level business object contains properties that enable it to detect RFC errors.

Resolving memory-related issues

You can increase the WebSphere Process Server or WebSphere Enterprise Service Bus memory limit if you encounter memory-related issues.

Increase the memory limit if you encounter the following problems:

- You see an out-of-memory error when a very large IDoc is sent from the SAP server to WebSphere Process Server or WebSphere Enterprise Service Bus.

- You see the error message `JCO Server could not unmarshall tables`.

To increase the memory limit, use the Jvm arguments for the initial (ms) and maximum (mx) size (for example, `-mx512m -mx256m`) in the server startup command.

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.

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 SAP Software. 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 11. 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 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 tutorial provides a complete set of instructions for configuring the adapter so that it can be used by a J2EE component (in this case, an SCA module) to send requests to the SAP server or by the SAP server to send requests to a J2EE component.

In the tutorials, you use WebSphere Integration Developer (and its enterprise service discovery wizard) to configure the adapter, connect to the SAP server, and retrieve information about a service on the SAP server. Enterprise service discovery then creates the business objects and interface information needed to interact with the service. The business objects and interface information, along with the adapter, are built into a deployable module.

Six tutorials are provided.

- Tutorial 1: Invoking a simple BAPI function

  The first BAPI outbound tutorial demonstrates how to create business objects based on a BAPI customer-related function call and how to create a module containing all the information needed to invoke the BAPI function.
- Tutorial 2: Invoking a BAPI transaction

  The second BAPI outbound tutorial demonstrates how to create business objects based on a BAPI transaction (a series of ordered BAPI function calls) and how to create a module containing all the information needed to invoke the transaction.
- Tutorial 3: Posting an IDoc to an SAP application

  The ALE outbound tutorial demonstrates how to create business objects based on an IDoc and how to create a module containing all the information needed to invoke an operation that posts an IDoc to an SAP application.
- Tutorial 4: Receiving a split IDoc packet

  The first ALE inbound tutorial demonstrates how the adapter receives events from the SAP server in the form of a split IDoc packet. The adapter converts the packet into a business object. You configure the adapter and set up an endpoint to receive the object.
- Tutorial 5: Receiving a non-split IDoc packet

  The second ALE inbound tutorial demonstrates how the adapter receives events from the SAP server in the form of a non-split IDoc packet. A non-split packet is treated as one unit of work. You configure the adapter and set up an endpoint to receive the object.
- Tutorial 6: Querying data in an SAP table

  The SQI tutorial demonstrates how to create a query object and send the query to the SAP server.

Learning objectives

After completing the outbound tutorials, 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 SAP server and make them part of the adapter project
- Create a deployable module that you install in the WebSphere Process Server test environment
- Test the module to ensure that it operates correctly and to see the results of running the module

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

- Create an adapter project in WebSphere Integration Developer
- Create a data source on the SAP server
- Discover services and associated business objects from the SAP server and make them part of the adapter project
- Create a deployable module that you install in the WebSphere Process Server test environment
- Test the module to ensure that it operates correctly and to see the results of running the module

Time required

The following table lists the approximate time it takes (after you have performed the prerequisite tasks and obtained the necessary data to run the tutorial) to complete each tutorial.

Table 12. Time required to complete the tutorials

Tutorial	Time to complete
Tutorial 1: Invoking a simple BAPI function	30 minutes
Tutorial 2: Invoking a BAPI transaction	45 minutes
Tutorial 3: Posting an IDoc to an SAP application	60 minutes
Tutorial 4: Receiving a split IDoc packet	60 minutes
Tutorial 5: Receiving a non-split IDoc packet	60 minutes
Tutorial 6: Querying data in an SAP table	30 minutes

Audience

The tutorials are intended for the integration developer who will be configuring Adapter for SAP Software for deployment on WebSphere Process Server or WebSphere Enterprise Service Bus.

Prerequisites

Before you begin the tutorials, make sure you have performed the following tasks:

- Install all prerequisite software
- Install Adapter for SAP Software

Also make sure you have all the information (such as user ID and password) needed to access the SAP server.

Values needed to set up inbound processing

When you configure your adapter for inbound processing, you set up values related to the SAP server and to the data on the server. If necessary, obtain this information from your SAP administrator.

For example, you must obtain actual values for **Partner Number of Sender** and **Partner Number of Recipient**.

The data that you must obtain is shown in the following table.

Table 13. Data required to set up the inbound environment

Tutorial	Obtain values for these fields
Tutorial 3: Posting an IDoc to an SAP application	Client IDoc Number Sender Port Partner Number of Sender Receiver Port Partner Number of Recipient
Tutorial 4: Receiving a split IDoc packet and Tutorial 5: Receiving a non-split IDoc packet	RFC Program ID Receiver Port Partner Type of Sender Name of Basic Type Partner Number of Sender Name of Table Structure Client Logical Message Type Partner Number of Recipient Sender Port IDoc Number Partner Number of Recipient

Values needed to test the tutorials

To test some of the tutorials, you need to obtain information from your SAP administrator about the data on the SAP server. For example, in the Querying data in an SAP table tutorial, you must use an existing customer number when you test the tutorial.

The data that you must obtain before you can run the tutorials is shown in the following table.

Table 14. Data required to complete the tutorials

Tutorial	Obtain values for these fields
Tutorial 2: Invoking a BAPI transaction	Sales Organization Distribution Channel Division Reference Customer
Tutorial 3: Posting an IDoc to an SAP application	Client IDoc Number Sender Port Partner Number of Sender Receiver Port Partner Number of Recipient
Tutorial 4: Receiving a split IDoc packet and Tutorial 5: Receiving a non-split IDoc packet	RFC Program ID Receiver Port Partner Type of Sender Name of Basic Type Partner Number of Sender Name of Table Structure Client Logical Message Type Partner Number of Recipient Sender Port IDoc Number Partner Number of Recipient
Tutorial 6: Querying data in an SAP table	Customer Number

Tutorial 1: Invoking a simple BAPI function

To create a module that invokes a simple BAPI function, you create an adapter project, use the enterprise service discovery wizard to generate business objects based on the function, and create a module that contains WebSphere Adapter for SAP Software and the newly generated business objects. You then deploy the module to the test environment of WebSphere Integration Developer.

Creating the authentication alias

To create an authentication alias, display the WebSphere Process Server administrative console and specify the user ID and password you use to access the SAP server. The user ID and password are then associated with the authentication alias.

1. Launch WebSphere Integration Developer by clicking **Start → Programs → IBM WebSphere → Integration Developer V6.0.2 → WebSphere Integration Developer V6.0.2**.
2. If you are prompted to specify a workspace, accept the default value.

   The workspace is a directory where WebSphere Integration Developer stores your project.
3. When the WebSphere Integration Developer window is displayed, close the Welcome page.

4. Switch to the Business Integration Perspective by clicking **Window** → **Open Perspective**. Then click **Business Integration (default)** and click **OK**.
5. Display the administrative console.
   a. Click the **Servers** tab.
   b. If **WebSphere Process Server v6.0** does not show a status of **Started**, right-click **WebSphere Process Server v6.0** and click **Start**.
   c. Right-click **WebSphere Process Server v6.0** and click **Run administrative console**.

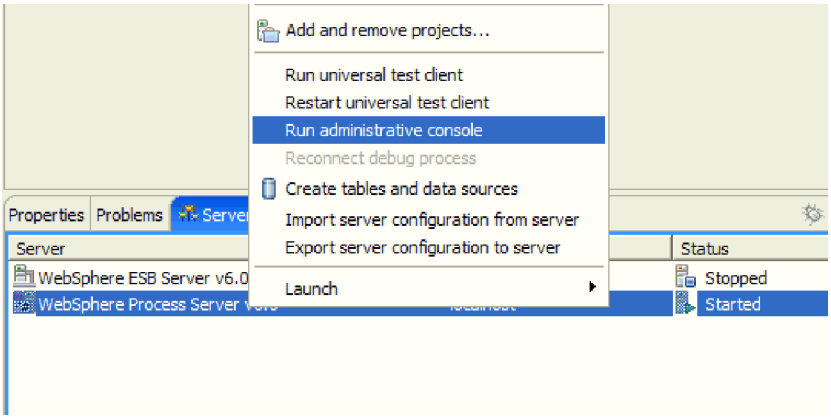


Figure 69. Selecting Run administrative console for the WebSphere Process Server

   d. Log in to the administrative console by typing **admin** and clicking **Log in**.
6. In the WebSphere Process Server administrative console, click **Security** → **Global security**.

Figure 70. The Security item on the administrative console

7. Under **Authentication**, click **JAAS Configuration** → **J2C Authentication data**.

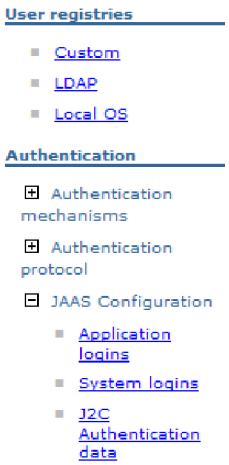


Figure 71. The Authentication section of the administrative console

8. If an alias named **SAP_Auth_Alias** does not already exist, create it now.
   a. Determine from your SAP administrator whether the authentication alias is case-sensitive (for example, whether the alias must be entered in uppercase).
   b. Click **New**.
   c. In the General properties window, type `SAP_Auth_Alias` in the **Alias** field.

      Note: If your SAP server requires that the alias be entered in a specific format (for example, all uppercase), type the alias according to that format.
   d. Type the user ID and password that are required to connect to the SAP server.

      Note: If your SAP server requires that the password be entered in a specific format (for example, all uppercase), type the password according to that format.
   e. Click **OK**.

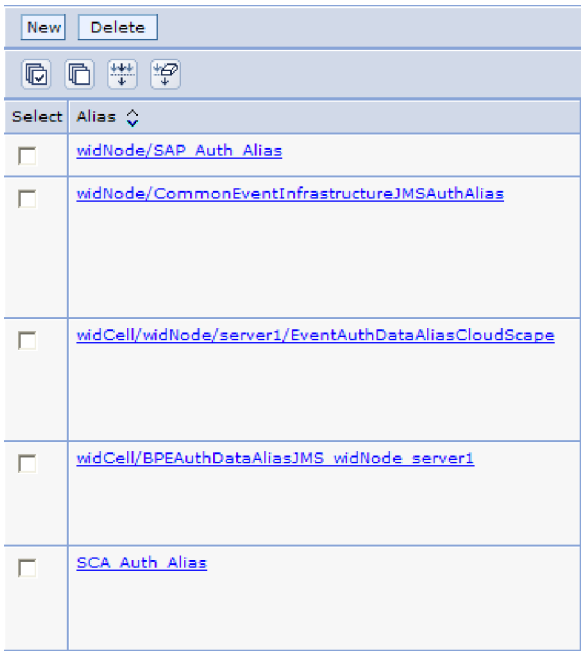


Figure 72. The list of aliases, including the newly created SAP_Auth_Alias

Make note of the name as it appears in the Alias list. In the example, the name is **widNode/SAP_Auth_Alias**. This name is the one you will use in subsequent configuration windows.

f. Click **Save**.

Result

You have created an authentication alias, which you will use when you configure the adapter properties.

Creating the adapter project in WebSphere Integration Developer

To begin the process of creating a module to communicate with an SAP service, you create an adapter project. The adapter project (called a *connector project* in WebSphere Integration Developer) 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.

About this task

You can use the same adapter project for multiple tutorials. If you have already created an adapter project by importing the adapter RAR file, you do not need to create it again, unless you want to have separate adapter projects for each tutorial.

How to perform this task

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**.

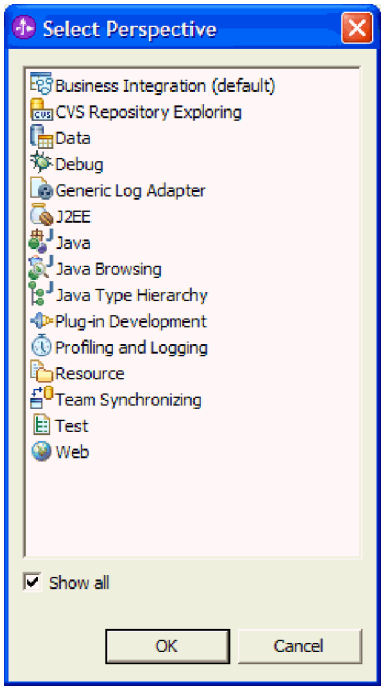


Figure 73. Selecting J2EE from the Select Perspective list

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

d. Click **OK**.

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

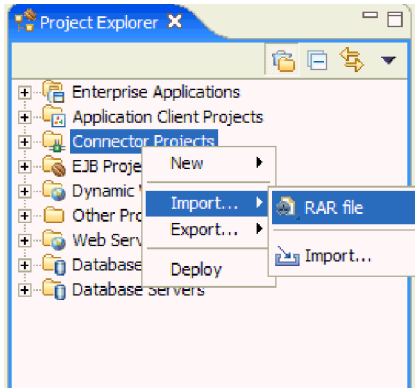


Figure 74. Importing the RAR file

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

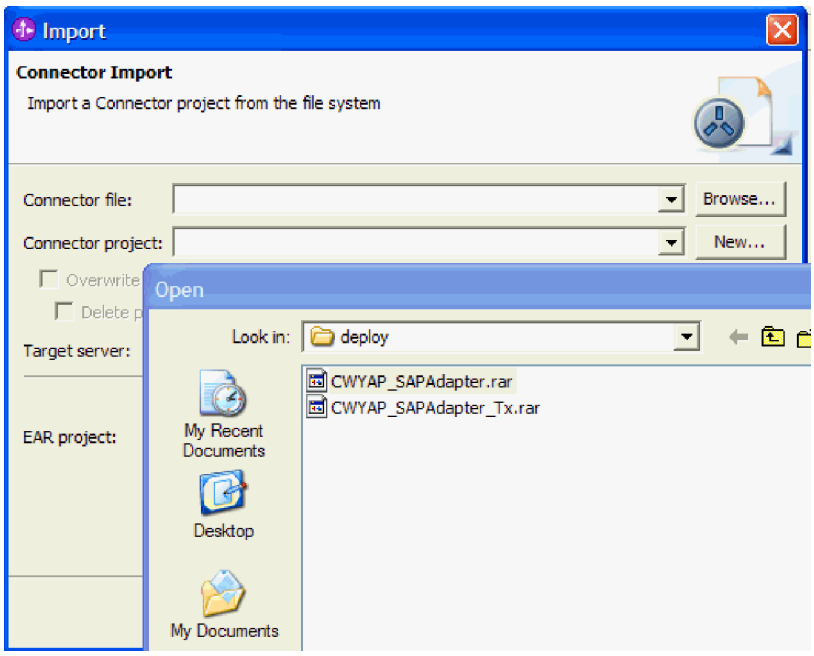


Figure 75. Selecting the RAR file from the installation directory

4. Select the RAR file and click **Open**.
5. Accept the default setting (**CWYAP_SAPAdapter**) for **Connector project**.

   The connector project has the same name as the RAR file.

   If a project named CWYAP_SAPAdapter.rar already exists in this workspace, the name in the **Connector project** field has a number appended to it (for example, CWYAP_SAPAdapter1).
6. 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.
7. Clear the **Add module to an EAR project** check box.

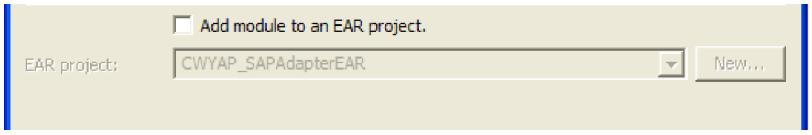


Figure 76. Clearing the Add module to an EAR project check box

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

Result

A new adapter project, named CWYAP_SAPAdapter, is created. To see its contents, expand **CWYAP_SAPAdapter**.

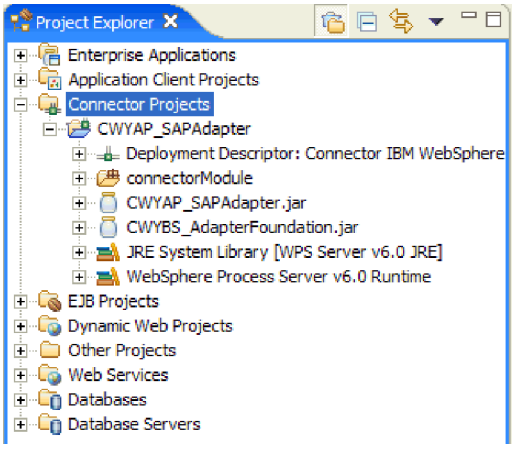


Figure 77. The CWYAP_SAPAdapter project in the Project Explorer window

Adding external dependencies

To add the required external dependency files, you copy the files, including the sapjco.jar file, to directories within the WebSphere Integration Developer directory. You then add the sapjco.jar file to the adapter project you created.

1. If you have not already done so as part of the installation of the adapter or as part of running another tutorial, copy the required files as outlined in the following steps.
   a. Obtain the files for your operating system from your SAP administrator or from the SAP Web site.

Table 15. Files to be installed

Operating system	Files to be installed
Windows	Any *.dll files that come with the SAP Jco download from the SAP Web site
Unix (including Unix System Services on z/OS)	Any .so and .o files that come with the SAP Jco download from the SAP Web site

   b. Copy the files to the following locations in the WebSphere Integration Developer installation directory:
      - \runtimes\bi_v6\java\bin
      - \eclipse\jre\bin

      For z/OS, add the files to the ${*WAS_INSTALL_ROOT*}/lib directory.
   c. For Windows environments only, obtain the msvcp71.dll and msvcr71.dll files from your SAP administrator or the SAP Web site.
   d. For Windows environments only, install the msvcp71.dll and msvcr71.dll files in the Windows system path.
   e. Obtain the sapjco.jar file from your SAP administrator or the SAP Web site.
   f. Copy sapjco.jar to the following location in the WebSphere Integration Developer installation directory: \runtimes\bi_v6\lib

      For z/OS, add ${*WAS_INSTALL_ROOT*}/lib/sapjco.jar to WAS_SERVER_ONLY_server_region_classpath
2. Import the sapjco.jar file into the adapter project.

a. In the J2EE perspective of WebSphere Integration Developer, expand **Connector Projects**.
b. Right-click **CWYAP_SAPAdapter** and click **Properties**.

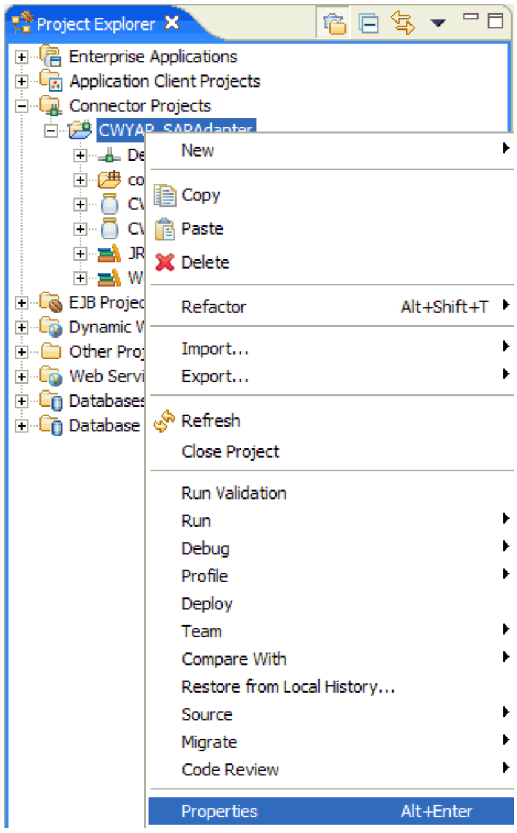


Figure 78. The CWYAP_SAPAdapter project, displayed in the Project Explorer

c. In the left side of the Properties for CWYAP_SAPAdapter window, click **Java Build Path**.

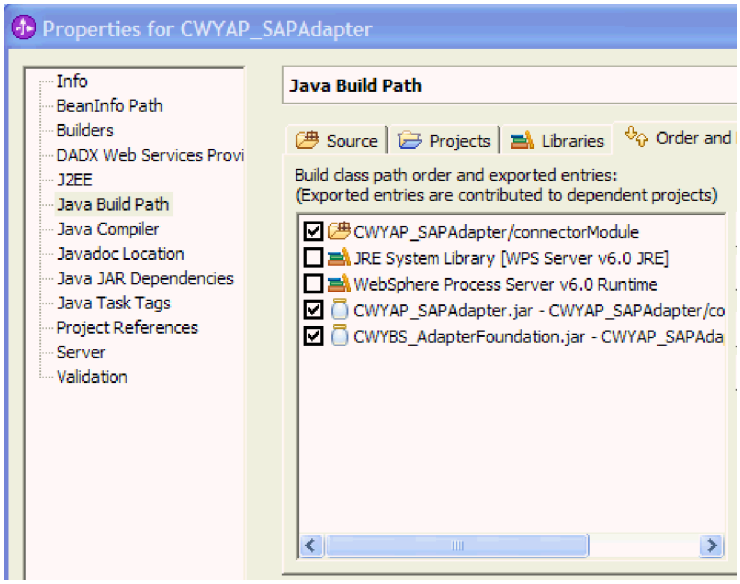


Figure 79. Selecting Java Build Path

d. Click the **Libraries** tab, and click **Add External JARs**.

e. Navigate to the directory on your local file system where the sapjco.jar file is located. Then select **sapjco.jar** and click **Open**.

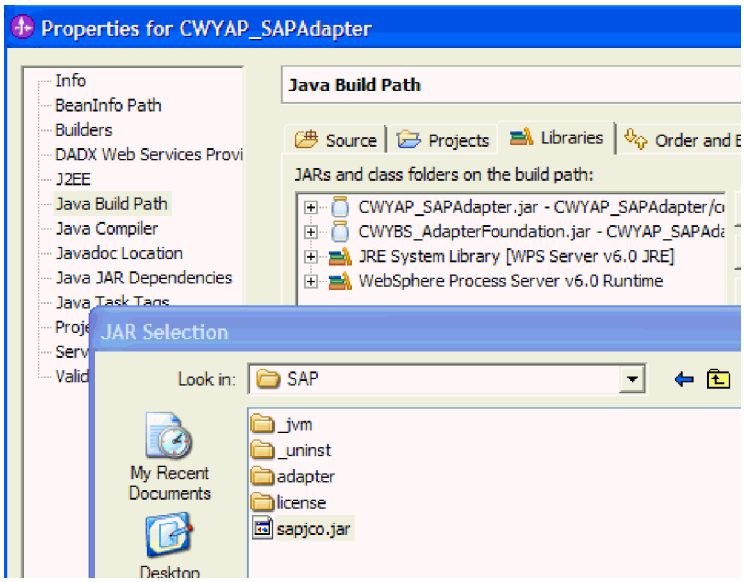


Figure 80. The JAR Selection window, with the sapjco.jar file highlighted for selection

f. Click **OK**.

The file sapjco.jar appears in the list of JARs and class folders in the build path.

Result

The sapjco.jar file is now part of your connector project and appears in the Project Explorer window of WebSphere Integration Developer.

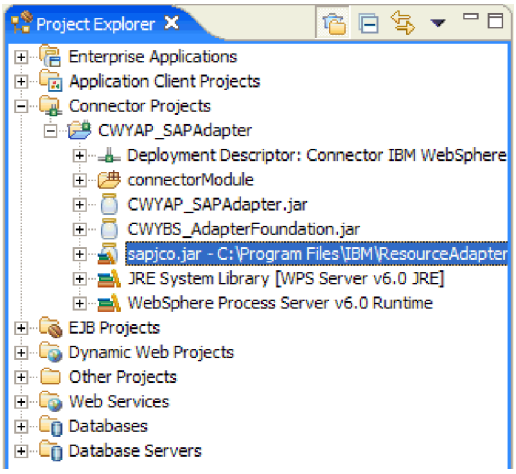


Figure 81. The Project Explorer window of WebSphere Integration Developer

Configuring the adapter for outbound processing

To configure the adapter, set the connection properties for enterprise service discovery. Then use the enterprise service discovery wizard to select and configure the necessary business objects and to generate a deployable module.

Setting connection properties for enterprise service discovery

To set connection properties for the enterprise service discovery wizard so that it can access the SAP server, specify such information as the user name and password you use to access the server as well as the name or IP address of the server.

1. In WebSphere Integration Developer, start the enterprise service discovery wizard.
   a. Switch to the Business Integration Perspective by clicking **Window** → **Open Perspective** → **Other**. Then click **Business Integration (default)** and click **OK**.
   b. Right-click **New** → **Enterprise Service Discovery**.

      If **Enterprise Service Discovery** is not displayed, click **New** → **Other**, expand **Business Integration**, and click **Enterprise Service Discovery**. Then click **Next**.

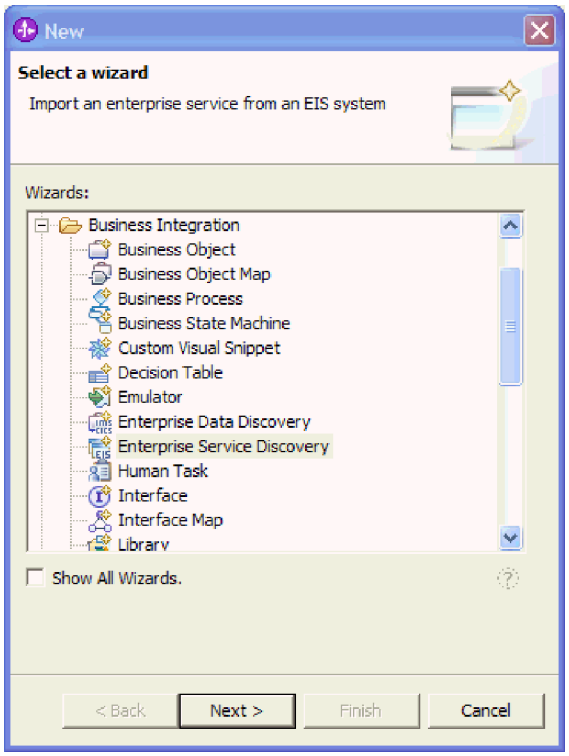


Figure 82. The expanded list of wizards

2. In the Select an Enterprise Service Resource Adapter window, make sure **IBM WebSphere Adapter for SAP Software** is selected, and click **Next**.
3. In the Configure Settings for Discovery Agent window, specify the configuration properties needed to connect to the SAP server.
   a. Type the name and password you use to access the SAP server.

      The password is case-sensitive.
   b. Type your client ID.

      This is typically `100`.
   c. In the **Application Server Host** field, type the name (or IP address) of your SAP server.

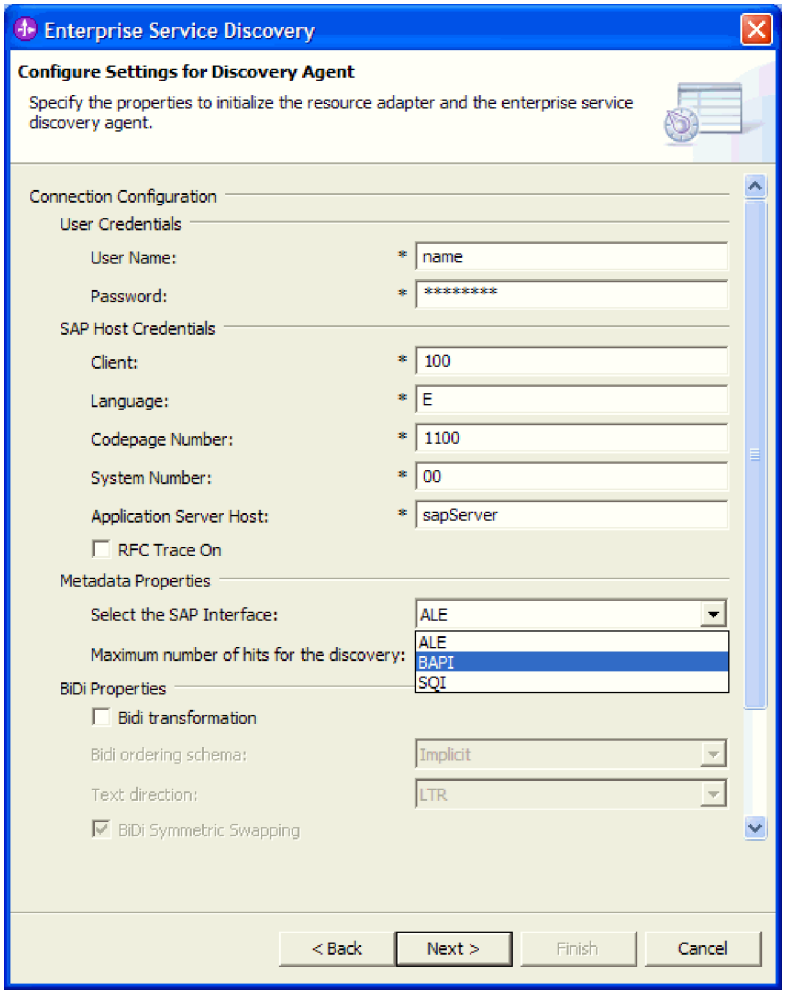


Figure 83. The Configure Settings for Discovery Agent window

4. Indicate which SAP interface you want to work with by selecting **BAPI** from the **Select the SAP interface** list.
5. Set the logging level so that you can see any errors that might arise during enterprise service discovery.
   a. At the bottom of the Configure Settings for Discovery Agent window, click **Show Advanced**.

      The button changes to **Hide Advanced**.

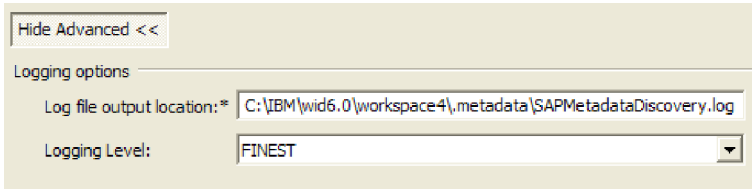


Figure 84. The Logging options displayed when you select Show Advanced

   b. For **Logging Level**, select **FINEST**.
6. Click **Next**.

Result

The enterprise service discovery wizard contacts the SAP server, using the information you provided (such as user name and password) to log in. You see the Find and Discover Enterprise Services window.

Selecting business objects and services

To select the BAPI function, specify search criteria (such as the name of the function). The enterprise service discovery wizard uses the search criteria to find the function on the SAP server.

1. From the Find and Discover Enterprise Services window, begin the process of searching for services by clicking **Execute Query**.
2. Indicate you are searching by the name of the function by expanding **RFC** and clicking **Discover By Name**.

   The **Filter** button is now enabled.
3. Click **Filter**.
4. In the Filter Properties for Discover By Name window, indicate that you want to see all BAPI functions that begin with the phrase BAPI_CUSTOMER by typing `BAPI_CUSTOMER*`.

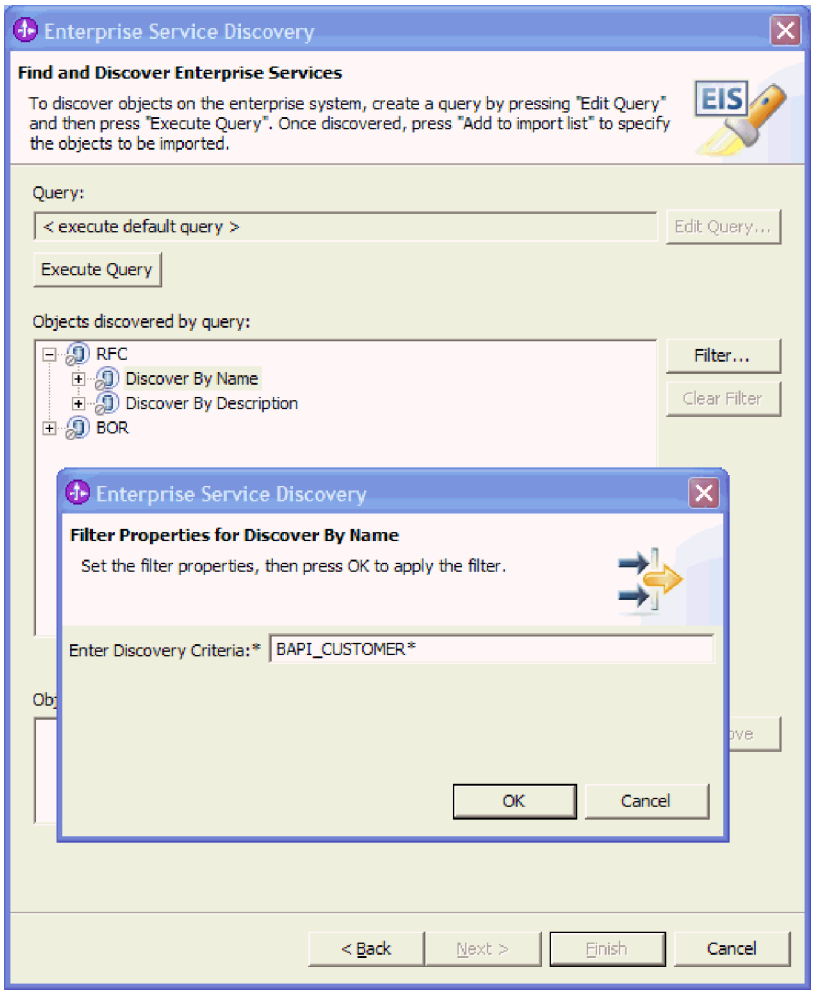


Figure 85. Entering the search criteria in the Filter Properties for Discover By Name window

5. Click **OK**.

6. Display a list of all the functions that meet the search criteria by expanding **Discover By Name (filtered)**.

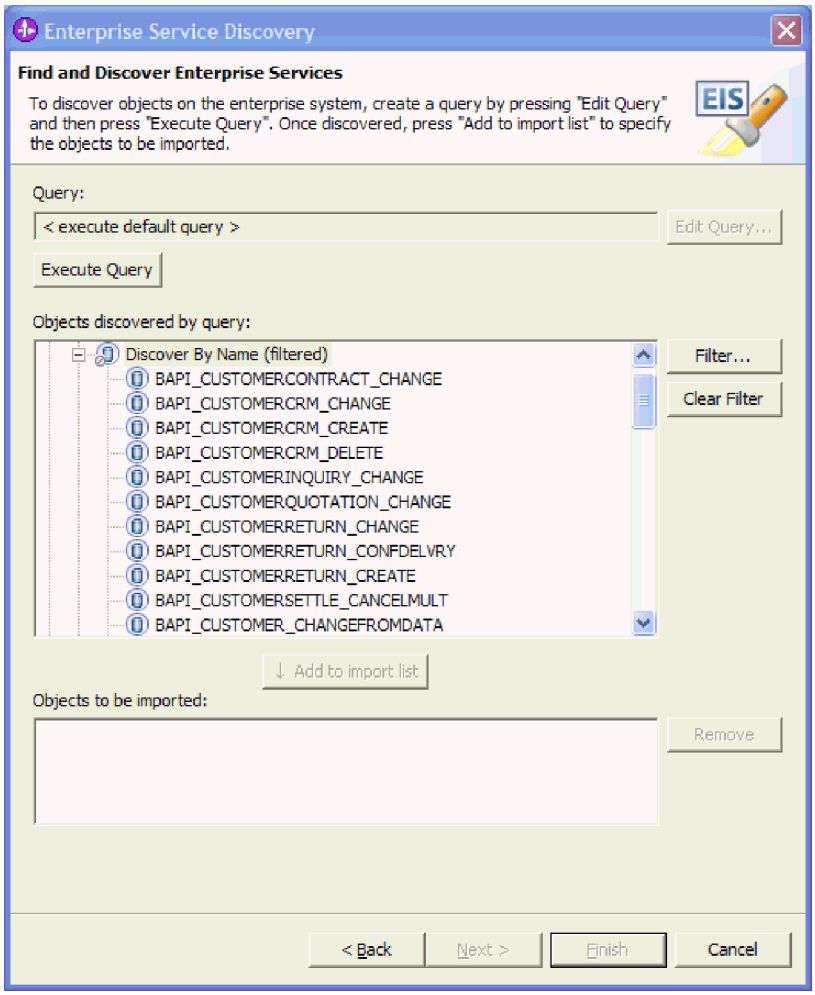


Figure 86. The list of BAPI functions that meet the search criteria

7. Scroll down and select **BAPI_CUSTOMER_GETLIST**, and then click **Add to import list**.

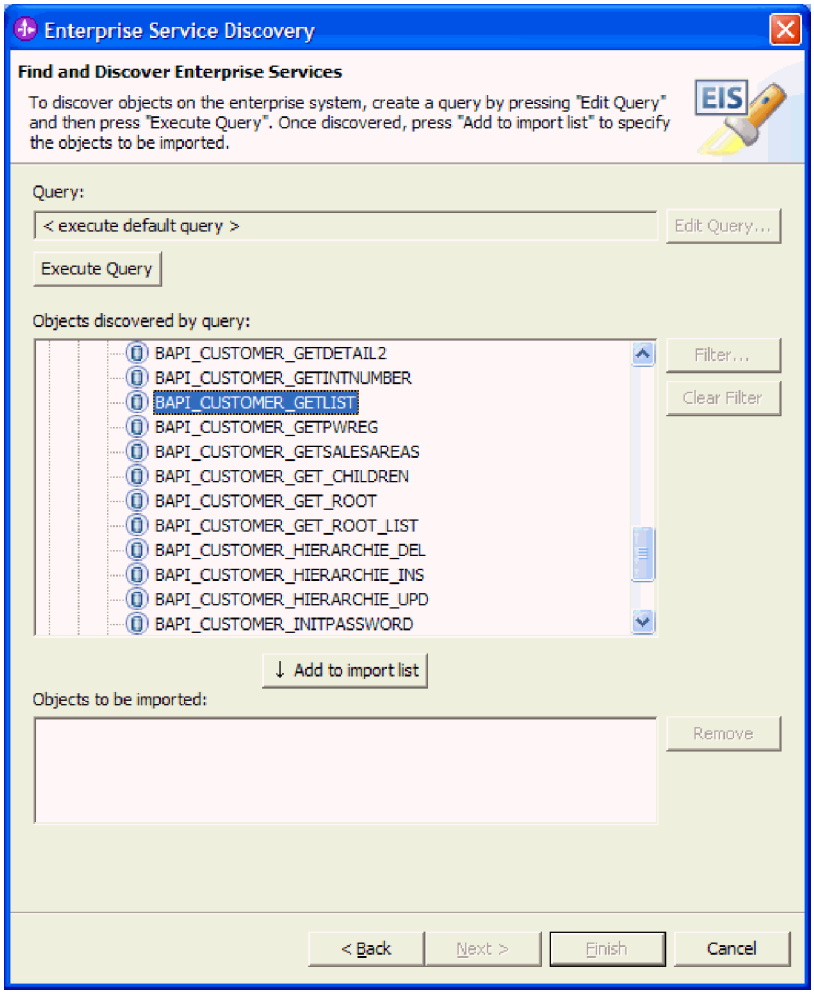


Figure 87. Selecting BAPI_CUSTOMER_GETLIST

8. In the Configuration Parameters for BAPI_CUSTOMER_GETLIST window, accept the defaults by clicking **OK**.

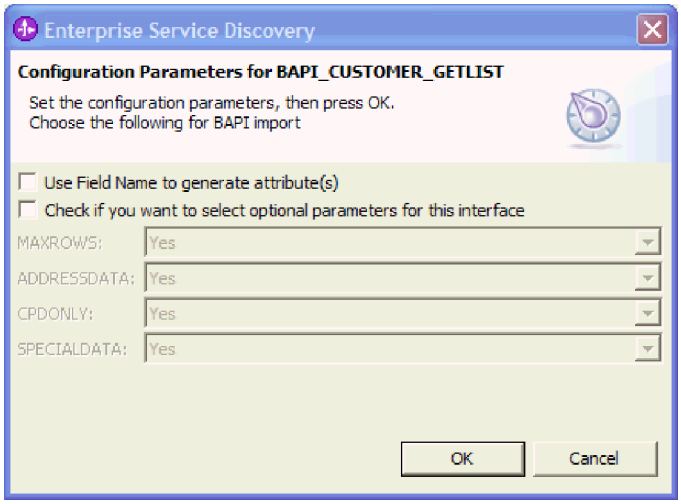


Figure 88. The Configuration Parameters window, which you leave blank to accept the defaults

BAPI_CUSTOMER_GETLIST is displayed under **Objects to be imported**.

9. Click **Next**.

Result

The enterprise service discovery wizard has found all functions whose names start with BAPI_CUSTOMER, and you have selected the BAPI_CUSTOMER_GETLIST function as the one you want to work with.

Configuring the selected objects

To configure the business object, you specify information about the object (such as the name of the object and the operation associated with the object).

1. In the Configure Objects window, specify the name of the object and the directory where it should be stored.
   a. In the **Object Location (Enter relative Path)** field, type `bodefs` as the name of the directory.
   b. Type `CustomerList` as the name of the business object.

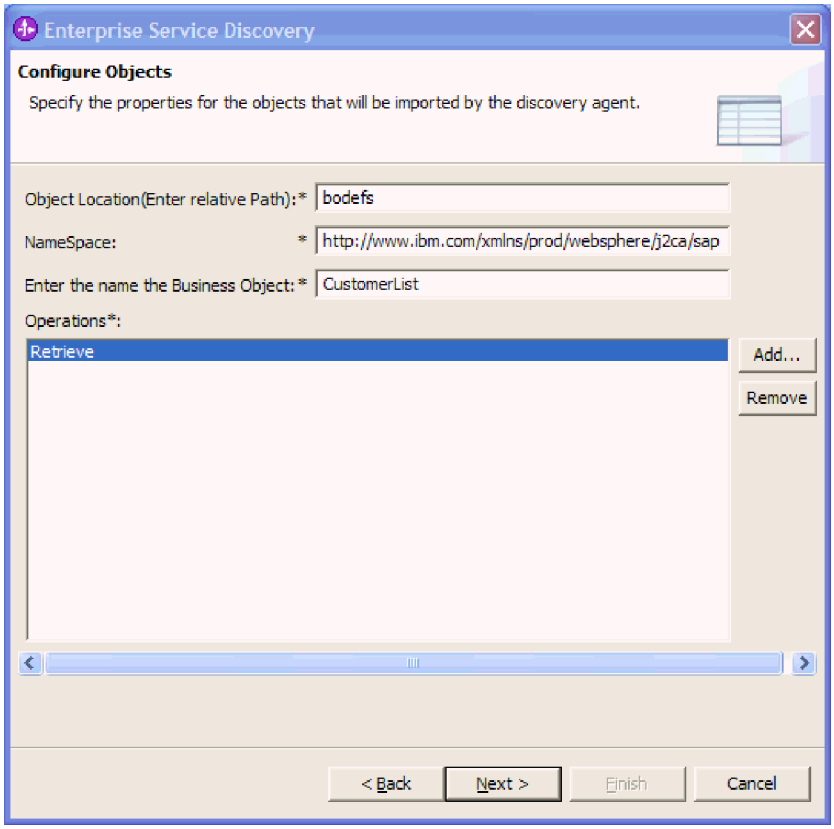


Figure 89. The Configure Objects window, with the sample values entered

2. Indicate which operation you want performed on the BAPI by clicking **Add**, selecting **Retrieve**, and clicking **OK**.
3. Click **Next**.

Result

You have associated an operation (Retrieve) with the object and selected a name for the object and the location where it is stored. The Generate Artifacts window is displayed.

Generating artifacts

To generate the module, which is the artifact that can be exported to an EAR file for deployment, you create the module, include the adapter in the module, and specify an alias used to authenticate the caller to the SAP server.

1. In the Generate Artifacts window, create a new module.
   a. Click **New**.
   b. Click **Create a module project** and click **Next**.
   c. Type `CustomerList`.

      As you type the name of the module, the name is added to the path next to **Directory**.

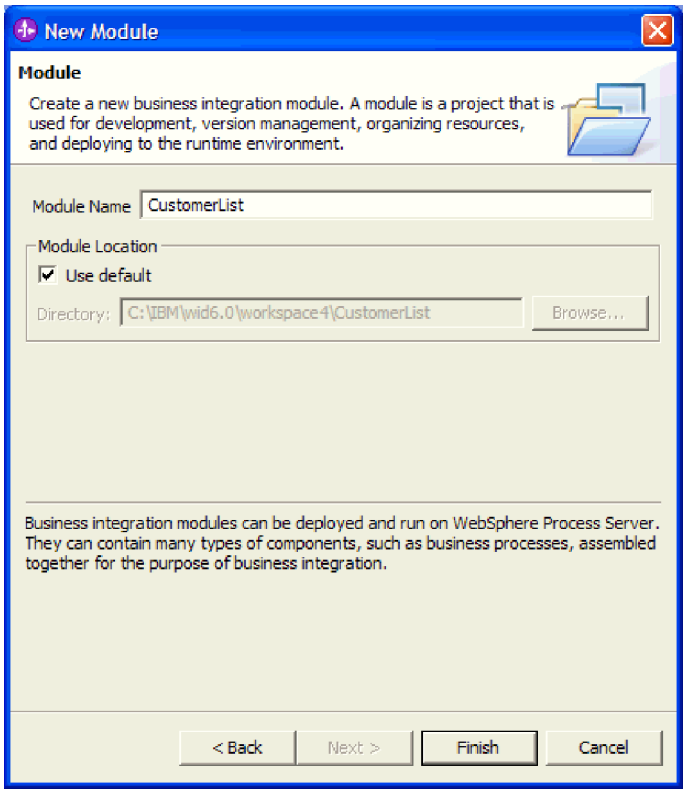


Figure 90. The New Module window

d. Click **Finish**.

2. Accept the default value for **Name** and **Deploy connector with module**.
3. Indicate the authentication alias to use by typing the alias that you created (in the beginning of the tutorial) in the administrative console. In the example shown earlier, the alias is `widNode/SAP_Auth_Alias`.
4. Select **Use discovered connection properties**.

   When you select **Use discovered connection properties**, the entries you made earlier (such as user name and IP address) are displayed at the bottom of the window.

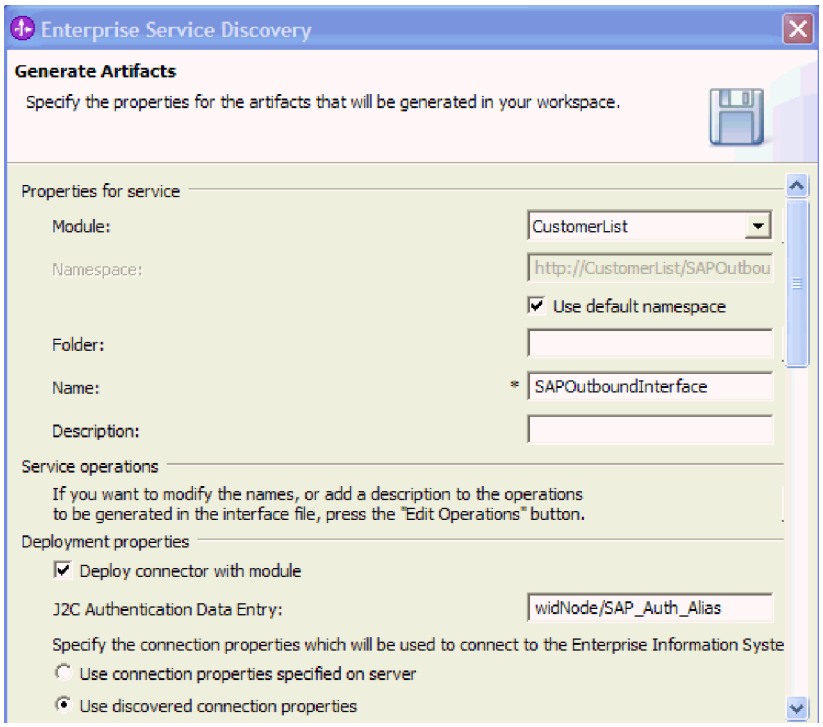


Figure 91. The Generate Artifacts window

5. Click **Finish**.

Result

The new CustomerList module is added to the Business Integration perspective.

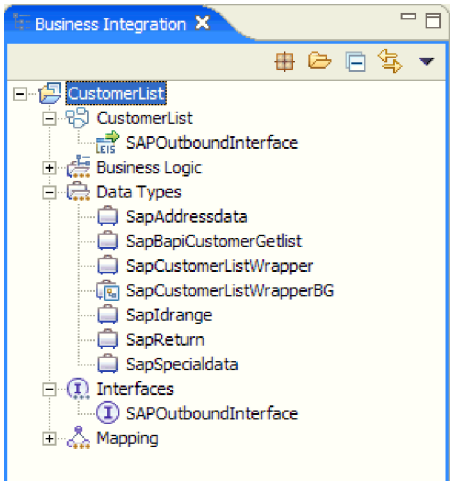


Figure 92. The CustomerList module displayed in the Business Integration perspective

Deploying the module for testing

To deploy the module to the test environment of WebSphere Process Server, you start the server and add the module (CustomerListApp) to it. "App" is appended to the name of the module to indicate that the module is a deployable application.

1. Select the test environment server.
   a. Click the **Servers** tab.
   b. Right-click **WebSphere Process Server v6.0**.

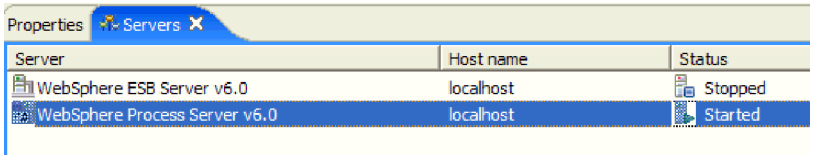


Figure 93. Selecting the WebSphere Process Server test environment from the Servers tab

2. Click **Add and remove projects**.
3. Select **CustomerListApp** and click **Add**.
4. Click **Finish**.

Result

You see status messages in the **Console** tab as CustomerListApp is deployed to the server.

Testing the module

Test the module to make sure you can retrieve a list of customers from the SAP server. You enter search criteria, and the list of customers meeting that criteria is returned.

1. In the Business Integration perspective, begin the testing procedure by right-clicking **CustomerList** and clicking **Test** → **Test Module**.

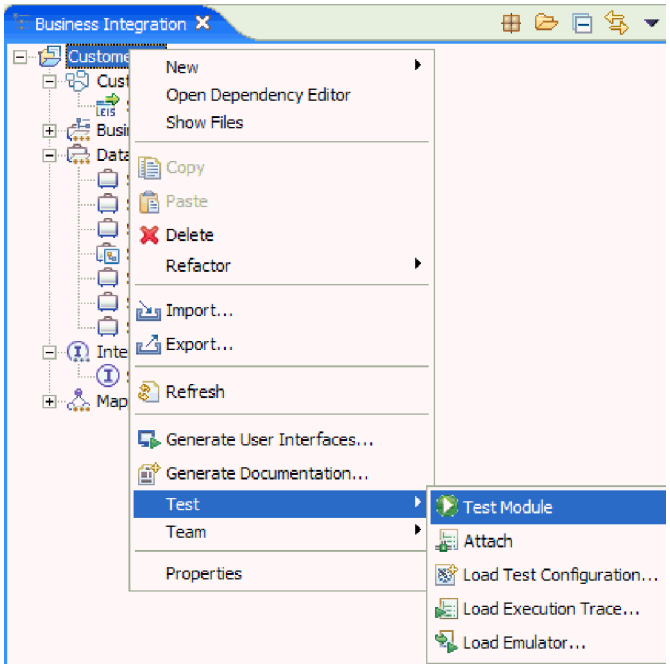


Figure 94. Specifying that you want to test CustomerListApp

2. In the Initial request parameters window, scroll to the **SapIdrange** array.

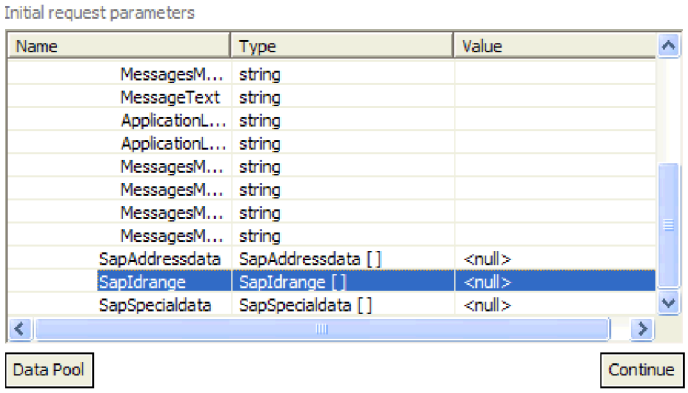


Figure 95. The Initial request parameters section of the test environment

3. Right-click **SapIdrange** and select **Add element**.
   If you are prompted to enter the number of elements to add, select **1** and click **OK**.
4. Select customer records from 1 through 100:
   a. Expand the new **SapIdrange** element.
   b. Type `E` for **InclusionExclusionCriterion**.
   c. Type `1` for **CustomerNumber1**.
   d. Type `100` for **CustomerNumber12217378**.

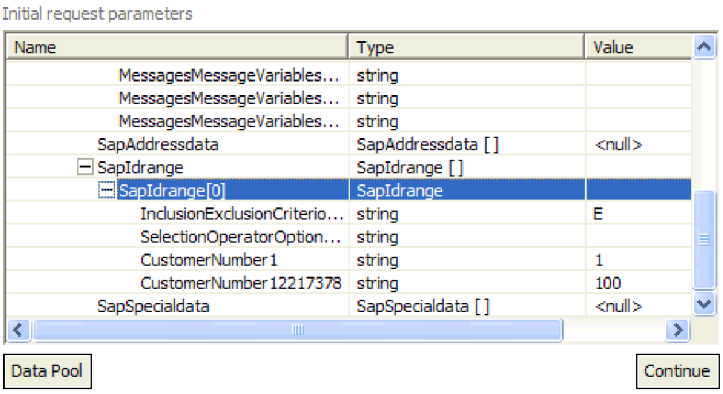


Figure 96. The Initial request parameters section of the test environment, with the sample values filled in

5. Click **Continue**.
6. Click **Finish**. The integration test client invokes CustomerListApp.
7. Scroll down to the Return parameters window to see the list of customers.

Result

The list of customers meeting the search criteria is displayed, indicating that CustomerListApp successfully completed.

Tutorial 2: Invoking a BAPI transaction

To create a module that invokes a BAPI transaction, you create an adapter project, use the enterprise service discovery wizard to generate business objects based on the functions of the transaction, and create a module that contains WebSphere Adapter for SAP Software and the newly generated business objects. You then deploy the module to the test environment of WebSphere Integration Developer.

Creating the authentication alias

To create an authentication alias, display the WebSphere Process Server administrative console and specify the user ID and password you use to access the SAP server. The user ID and password are then associated with the authentication alias.

1. Launch WebSphere Integration Developer by clicking **Start** → **Programs** → **IBM WebSphere** → **Integration Developer V6.0.2** → **WebSphere Integration Developer V6.0.2**.
2. If you are prompted to specify a workspace, accept the default value.

   The workspace is a directory where WebSphere Integration Developer stores your project.
3. When the WebSphere Integration Developer window is displayed, close the Welcome page.
4. Switch to the Business Integration Perspective by clicking **Window** → **Open Perspective**. Then click **Business Integration (default)** and click **OK**.
5. Display the administrative console.
   a. Click the **Servers** tab.
   b. If **WebSphere Process Server v6.0** does not show a status of **Started**, right-click **WebSphere Process Server v6.0** and click **Start**.
   c. Right-click **WebSphere Process Server v6.0** and click **Run administrative console**.

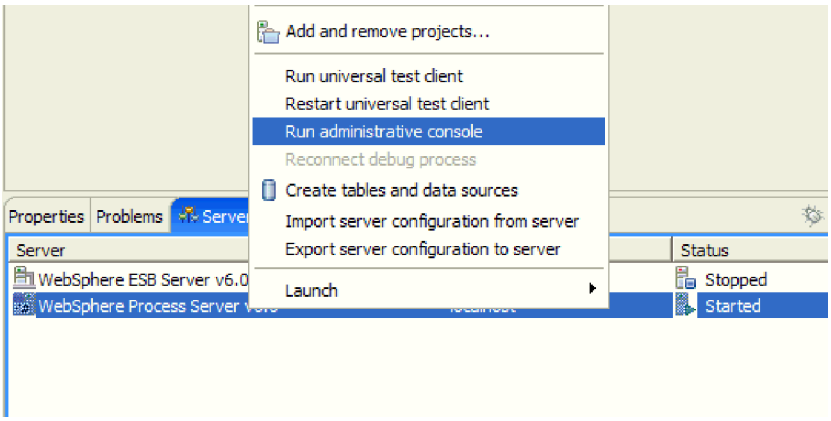


Figure 97. Selecting Run administrative console for the WebSphere Process Server

   d. Log in to the administrative console by typing **admin** and clicking **Log in**.
6. In the WebSphere Process Server administrative console, click **Security** → **Global security**.

Figure 98. The Security item on the administrative console

7. Under **Authentication**, click **JAAS Configuration** → **J2C Authentication data**.

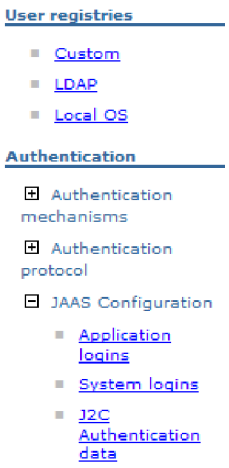


Figure 99. The Authentication section of the administrative console

8. If an alias named **SAP_Auth_Alias** does not already exist, create it now.
   a. Determine from your SAP administrator whether the authentication alias is case-sensitive (for example, whether the alias must be entered in uppercase).
   b. Click **New**.
   c. In the General properties window, type `SAP_Auth_Alias` in the **Alias** field.

      Note: If your SAP server requires that the alias be entered in a specific format (for example, all uppercase), type the alias according to that format.
   d. Type the user ID and password that are required to connect to the SAP server.

Note: If your SAP server requires that the password be entered in a specific format (for example, all uppercase), type the password according to that format.

e. Click **OK**.

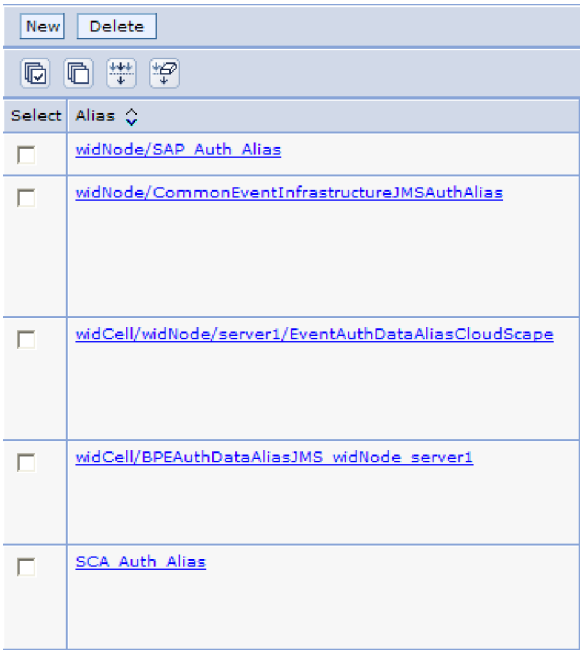


Figure 100. The list of aliases, including the newly created SAP_Auth_Alias

Make note of the name as it appears in the Alias list. In the example, the name is **widNode/SAP_Auth_Alias**. This name is the one you will use in subsequent configuration windows.

f. Click **Save**.

Result

You have created an authentication alias, which you will use when you configure the adapter properties.

Creating the adapter project in WebSphere Integration Developer

To begin the process of creating a module to communicate with an SAP service, you create an adapter project. The adapter project (called a *connector project* in WebSphere Integration Developer) 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.

About this task

You can use the same adapter project for multiple tutorials. If you have already created an adapter project by importing the adapter RAR file, you do not need to create it again, unless you want to have separate adapter projects for each tutorial.

How to perform this task

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**.

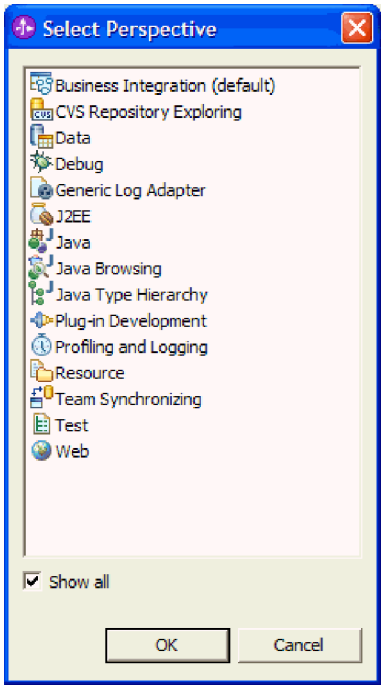


Figure 101. Selecting J2EE from the Select Perspective list

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

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

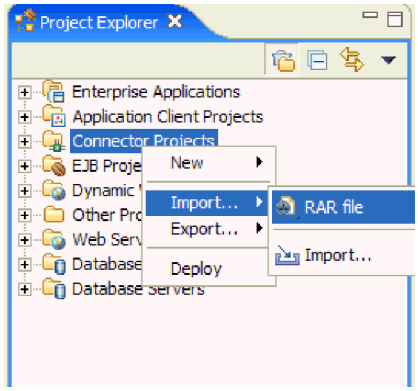


Figure 102. Importing the RAR file

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

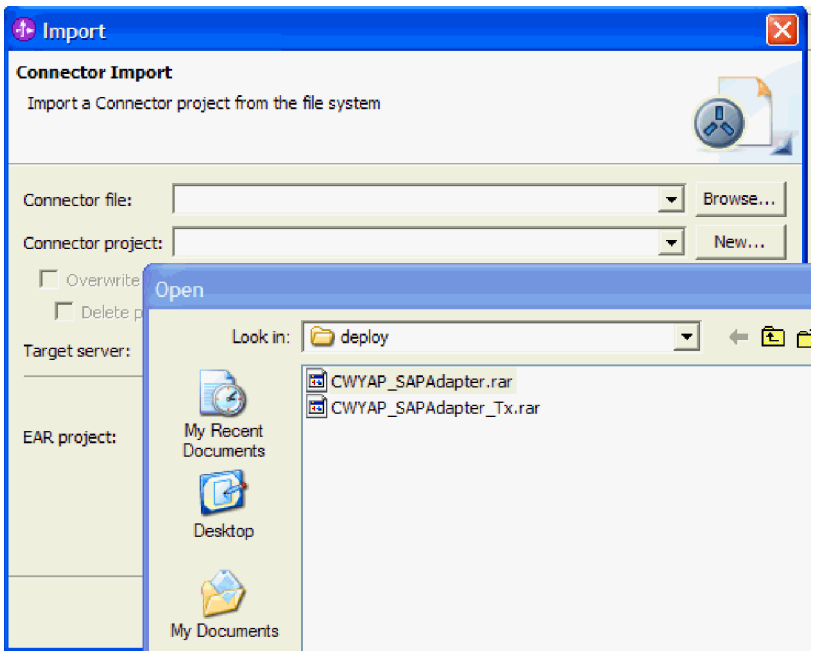


Figure 103. Selecting the RAR file from the installation directory

4. Select the RAR file and click **Open**.
5. Accept the default setting (**CWYAP_SAPAdapter**) for **Connector project**.

   The connector project has the same name as the RAR file.

   If a project named CWYAP_SAPAdapter.rar already exists in this workspace, the name in the **Connector project** field has a number appended to it (for example, CWYAP_SAPAdapter1).
6. 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.
7. Clear the **Add module to an EAR project** check box.

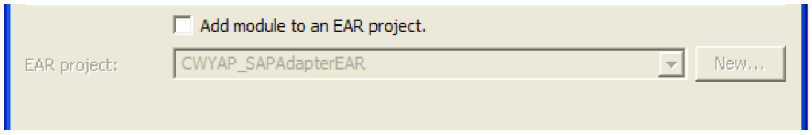


Figure 104. Clearing the Add module to an EAR project check box

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

Result

A new adapter project, named CWYAP_SAPAdapter, is created. To see its contents, expand **CWYAP_SAPAdapter**.

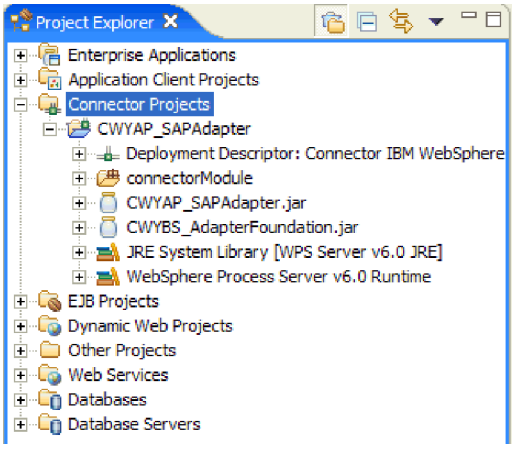


Figure 105. The CWYAP_SAPAdapter project in the Project Explorer window

Adding external dependencies

To add the required external dependency files, you copy the files, including the sapjco.jar file, to directories within the WebSphere Integration Developer directory. You then add the sapjco.jar file to the adapter project you created.

1. If you have not already done so as part of the installation of the adapter or as part of running another tutorial, copy the required files as outlined in the following steps.
   a. Obtain the files for your operating system from your SAP administrator or from the SAP Web site.

Table 16. Files to be installed

Operating system	Files to be installed
Windows	Any *.dll files that come with the SAP Jco download from the SAP Web site
Unix (including Unix System Services on z/OS)	Any .so and .o files that come with the SAP Jco download from the SAP Web site

   b. Copy the files to the following locations in the WebSphere Integration Developer installation directory:
      - \runtimes\bi_v6\java\bin
      - \eclipse\jre\bin

      For z/OS, add the files to the ${*WAS_INSTALL_ROOT*}/lib directory.
   c. For Windows environments only, obtain the msvcp71.dll and msvcr71.dll files from your SAP administrator or the SAP Web site.
   d. For Windows environments only, install the msvcp71.dll and msvcr71.dll files in the Windows system path.
   e. Obtain the sapjco.jar file from your SAP administrator or the SAP Web site.
   f. Copy sapjco.jar to the following location in the WebSphere Integration Developer installation directory: \runtimes\bi_v6\lib

      For z/OS, add ${*WAS_INSTALL_ROOT*}/lib/sapjco.jar to WAS_SERVER_ONLY_server_region_classpath
2. Import the sapjco.jar file into the adapter project.

a. In the J2EE perspective of WebSphere Integration Developer, expand **Connector Projects**.
b. Right-click **CWYAP_SAPAdapter** and click **Properties**.

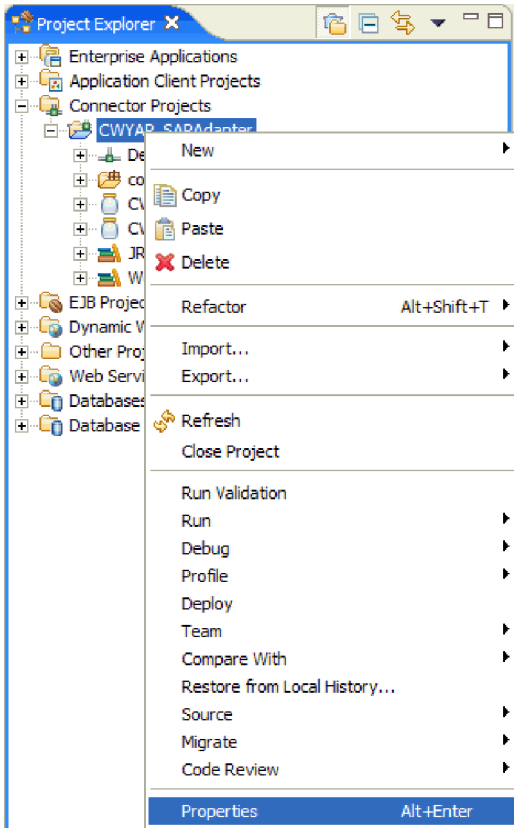


Figure 106. The CWYAP_SAPAdapter project, displayed in the Project Explorer

c. In the left side of the Properties for CWYAP_SAPAdapter window, click **Java Build Path**.

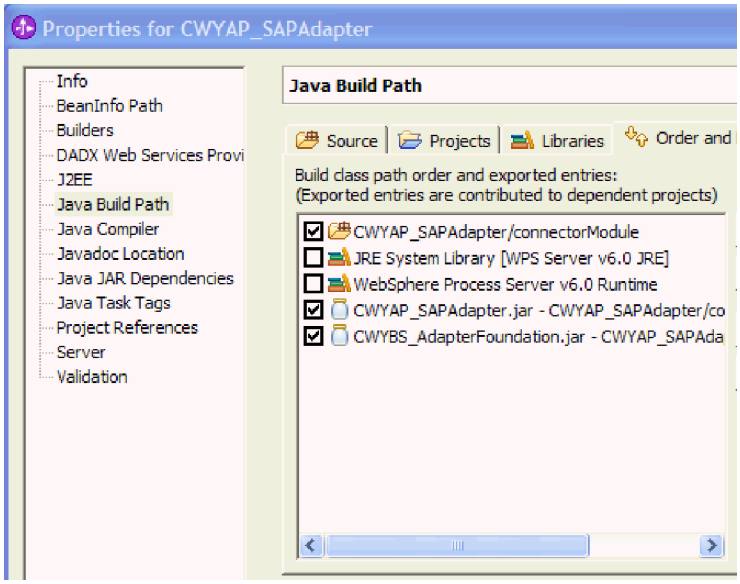


Figure 107. Selecting Java Build Path

d. Click the **Libraries** tab, and click **Add External JARs**.

e. Navigate to the directory on your local file system where the sapjco.jar file is located. Then select **sapjco.jar** and click **Open**.

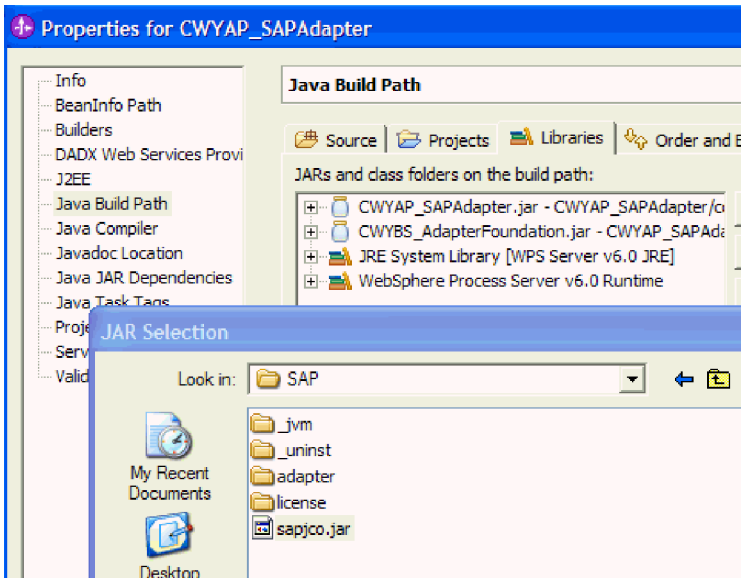


Figure 108. The JAR Selection window, with the sapjco.jar file highlighted for selection

f. Click **OK**.

The file sapjco.jar appears in the list of JARs and class folders in the build path.

Result

The sapjco.jar file is now part of your connector project and appears in the Project Explorer window of WebSphere Integration Developer.

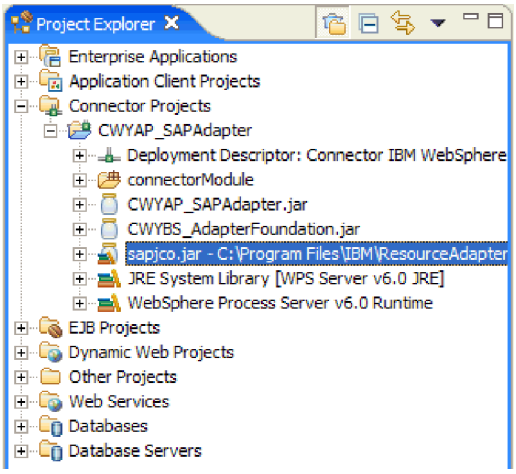


Figure 109. The Project Explorer window of WebSphere Integration Developer

Configuring the adapter for outbound processing

To configure the adapter, set the connection properties for enterprise service discovery. Then use the enterprise service discovery wizard to select and configure the necessary business objects and to generate a deployable module.

Setting connection properties for enterprise service discovery

To set connection properties for the enterprise service discovery wizard so that it can access the SAP server, specify such information as the user name and password you use to access the server as well as the name or IP address of the server.

1. In WebSphere Integration Developer, start the enterprise service discovery wizard.
   a. Switch to the Business Integration Perspective by clicking **Window** → **Open Perspective** → **Other**. Then click **Business Integration (default)** and click **OK**.
   b. Right-click **New** → **Enterprise Service Discovery**.

      If **Enterprise Service Discovery** is not displayed, click **New** → **Other**, expand **Business Integration**, and click **Enterprise Service Discovery**. Then click **Next**.

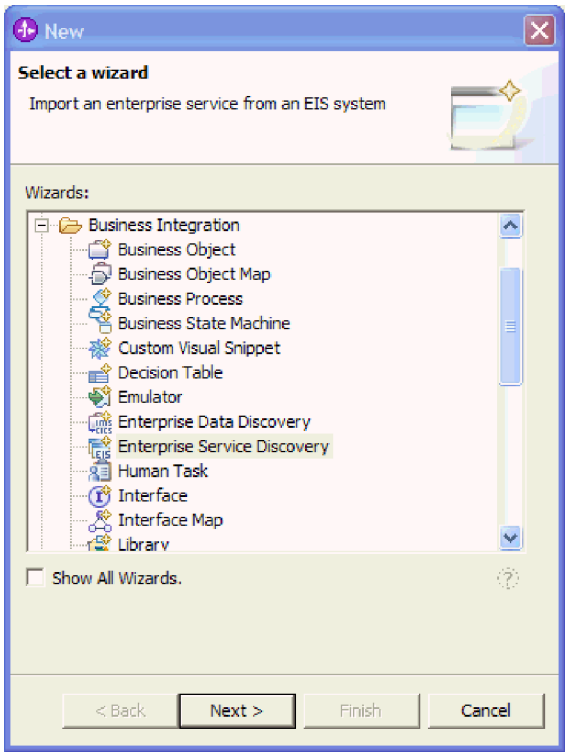


Figure 110. The expanded list of wizards

2. In the Select an Enterprise Service Resource Adapter window, make sure **IBM WebSphere Adapter for SAP Software** is selected, and click **Next**.
3. In the Configure Settings for Discovery Agent window, specify the configuration properties needed to connect to the SAP server.
   a. Type the name and password you use to access the SAP server.

      The password is case-sensitive.
   b. Type your client ID.

      This is typically `100`.
   c. In the **Application Server Host** field, type the name (or IP address) of your SAP server.

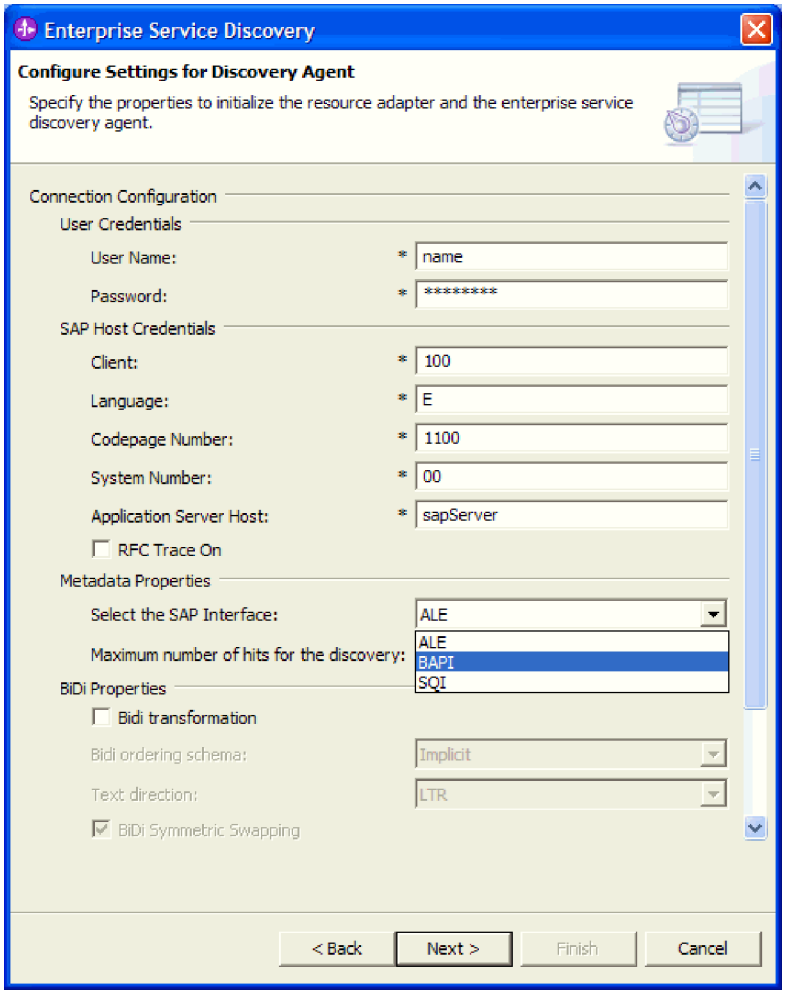


Figure 111. The Configure Settings for Discovery Agent window

4. Indicate which SAP interface you want to work with by selecting **BAPI** from the **Select the SAP interface** list.
5. Set the logging level so that you can see any errors that might arise during enterprise service discovery.
   a. At the bottom of the Configure Settings for Discovery Agent window, click **Show Advanced**.

      The button changes to **Hide Advanced**.

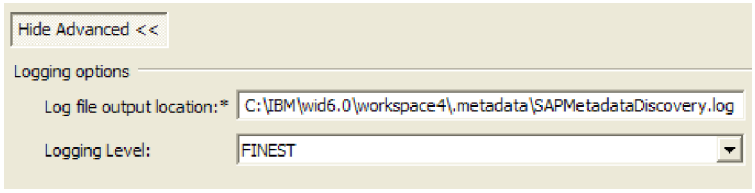


Figure 112. The Logging options displayed when you select Show Advanced

   b. For **Logging Level**, select **FINEST**.
6. Click **Next**.

Result

The enterprise service discovery wizard contacts the SAP server, using the information you provided (such as user name and password) to log in. You see the Find and Discover Enterprise Services window.

Selecting business objects and services

To select the BAPI functions that make up a transaction, specify search criteria (such as the name of the functions). The enterprise service discovery wizard uses the search criteria to find the functions on the SAP server.

1. From the Find and Discover Enterprise Services window, begin the process of searching for services by clicking **Execute Query**.
2. Indicate you are searching by the name of the functions by expanding **RFC** and clicking **Discover By Name**.

   The **Filter** button is now enabled.
3. Click **Filter**.
4. In the Filter Properties for Discover By Name window, indicate that you want to see all BAPI functions that begin with the phrase BAPI_CUSTOMER by typing `BAPI_CUSTOMER*`.

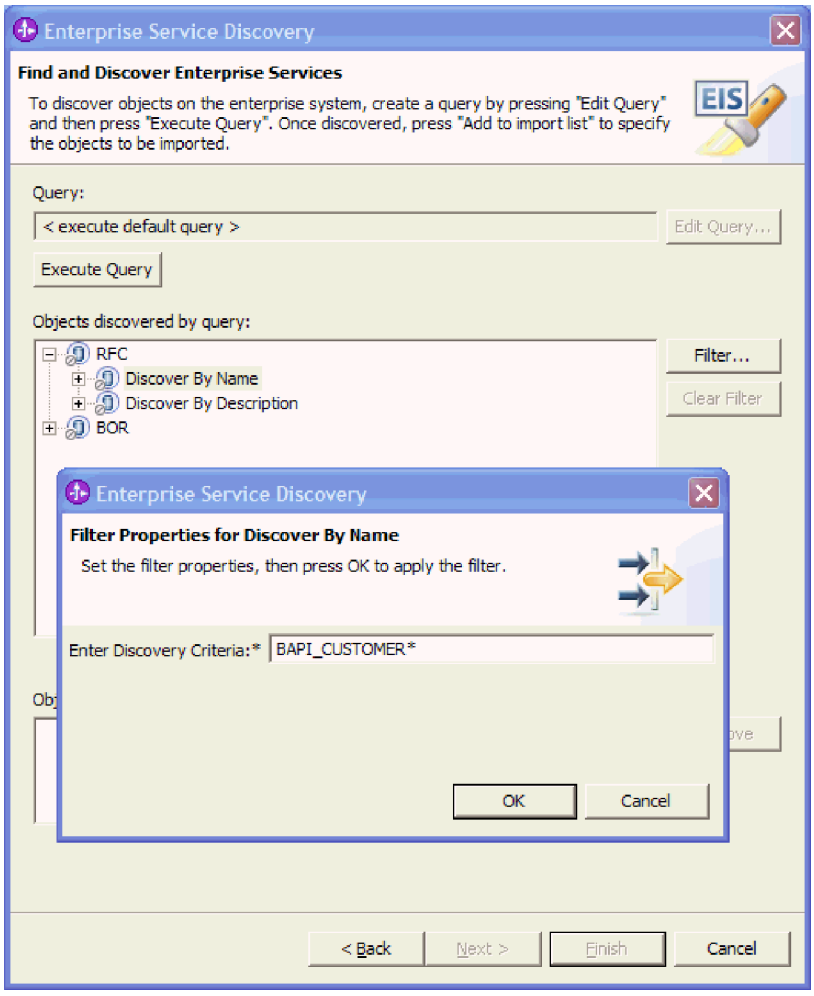


Figure 113. Entering the search criteria in the Filter Properties for Discover By Name window

5. Click **OK**.

6. Display a list of all the functions that meet the search criteria by expanding **Discover By Name (filtered)**.

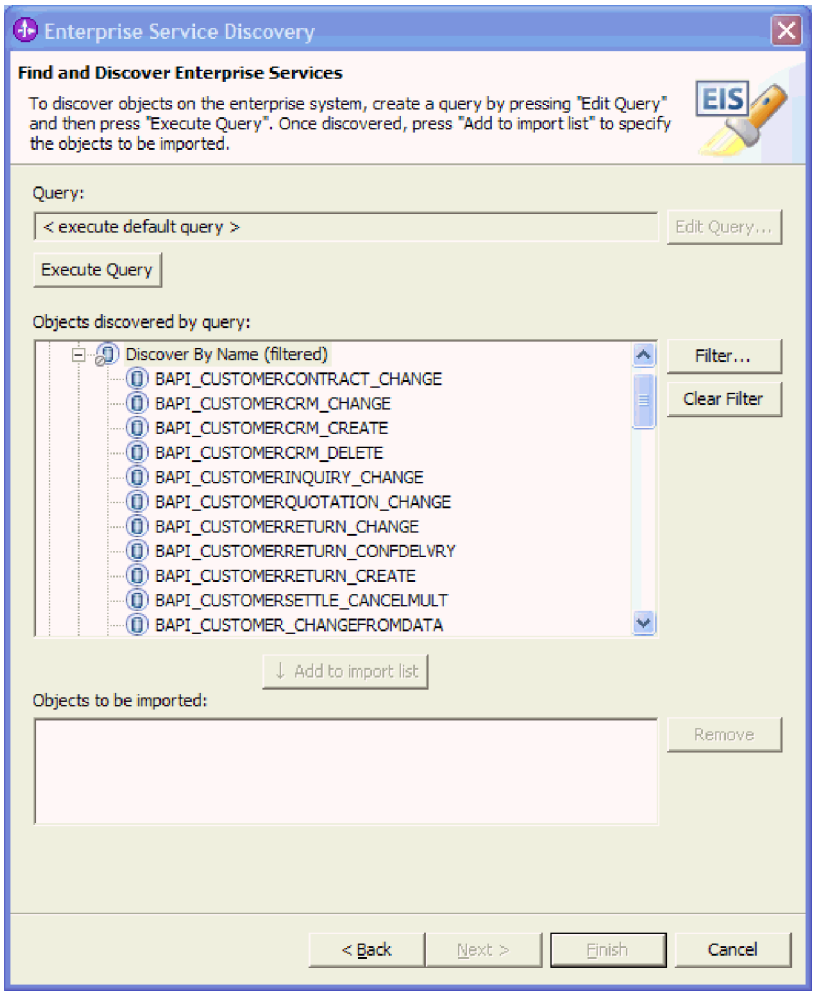


Figure 114. The list of BAPI functions that meet the search criteria

7. Select **BAPI_CUSTOMER_CREATEFROMDATA1** and **BAPI_CUSTOMER_CHANGEFROMDATA**, and then click **Add to import list**.

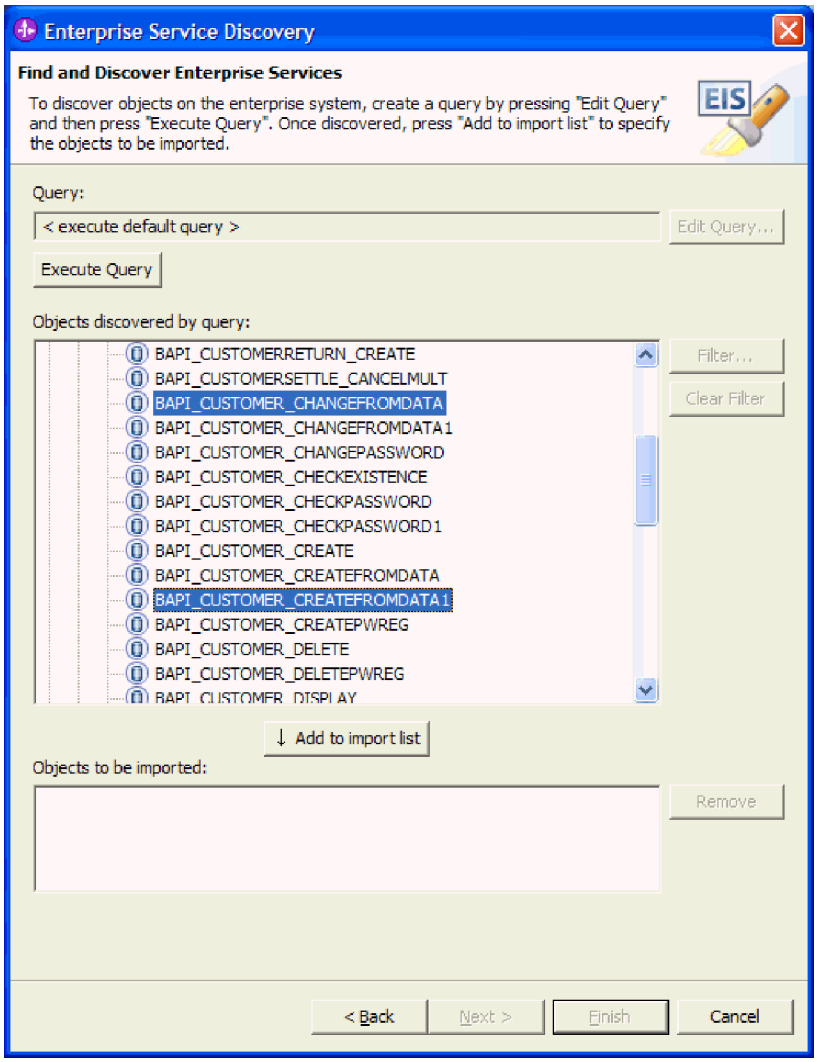


Figure 115. Selecting the BAPI functions

8. In the Configuration Parameters window, accept the defaults for the two BAPIs by clicking **OK**.

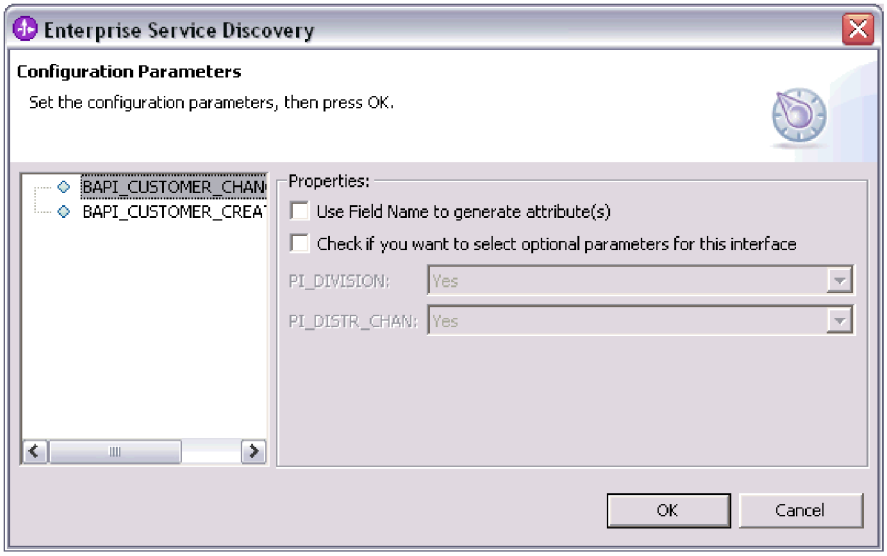


Figure 116. The Configuration Parameters window

9. Click **Next**.

Result

The enterprise service discovery wizard has found all functions whose names start with BAPI_CUSTOMER, and you have selected two functions to be part of the transaction.

Configuring the selected objects

To configure the transaction business object, you specify information about the object (such as the name of the object and the operation associated with the transaction).

1. In the Configure Objects window, specify the name of the object and the directory where it should be stored, and indicate that it is a transaction object.
   a. For **Object Location (Enter relative path)**, type `Customer`.
   b. For **Enter the name of Business Object**, type `Customer`.
   c. Select the **Create a BAPI transaction object from the selected BAPIs** check box.

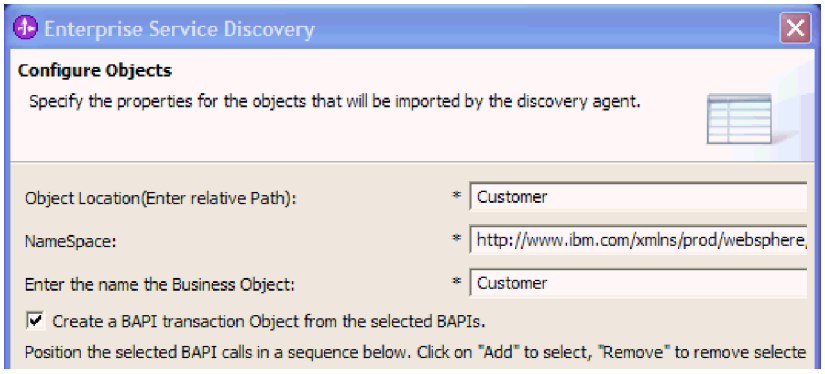


Figure 117. The Configure Objects window

2. Click **Add**.
   Two BAPIs and a COMMIT operation are listed.
3. Select both BAPIs and **COMMIT**, and click **OK**.

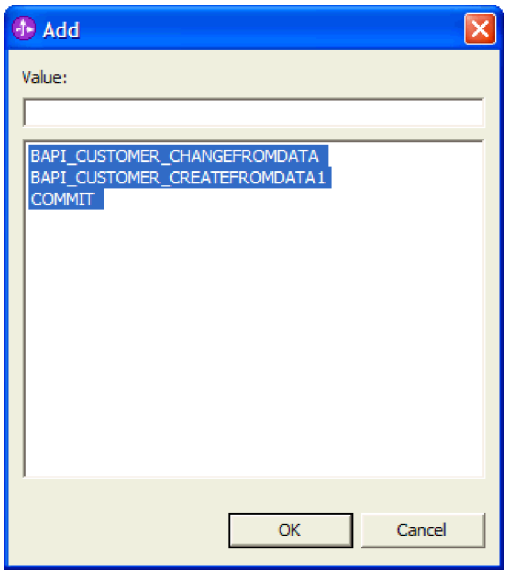


Figure 118. The Add window, with the two BAPIs and COMMIT highlighted

4. Specify the operation for this transaction by clicking **Create**.

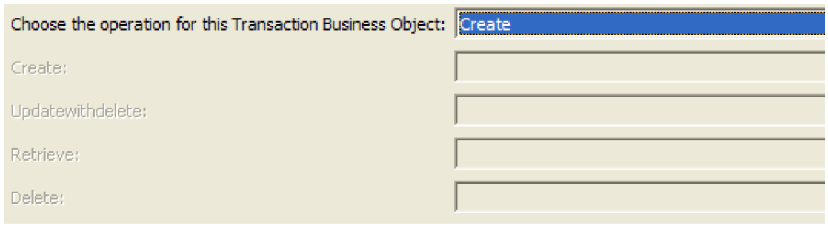


Figure 119. The operations available for selection

5. Click **Next**.

Result

You have associated an operation (Create) with the transaction object and selected a name for the object. The Generate Artifacts window is displayed.

Generating artifacts

To generate the module, which is the artifact that can be exported to an EAR file for deployment, you create the module, include the adapter in the module, and specify an alias used to authenticate the caller to the SAP server.

1. In the Generate Artifacts window, create a new module.
   a. Click **New**.
   b. Click **Create a module project** and click **Next**.
   c. Type `Customer`.

As you type the name of the module, the name is added to the path next to **Directory**.

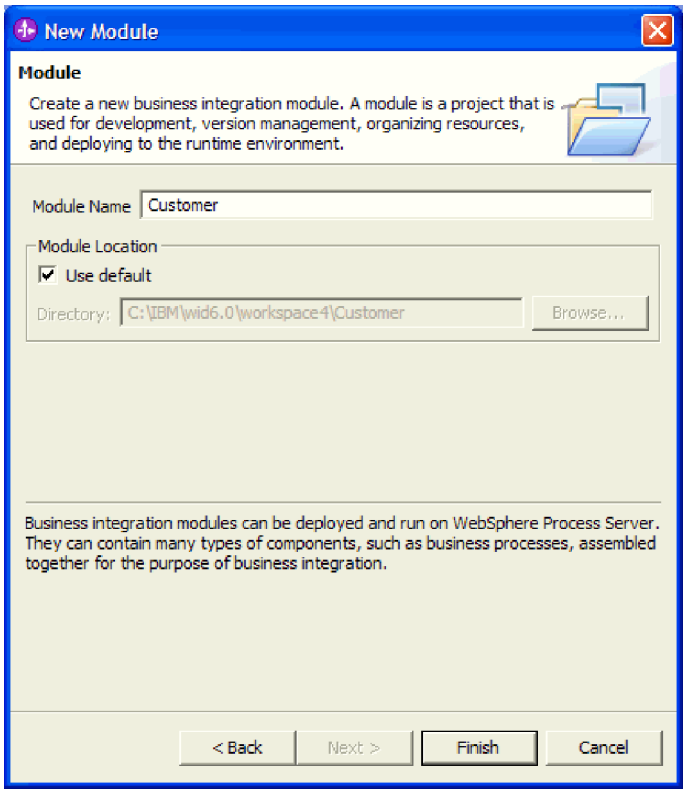


Figure 120. The New Module window

d. Click **Finish**.

2. Accept the default value for **Name** and **Deploy connector with module**.
3. Select **Use discovered connection properties**.

   When you select **Use discovered connection properties**, the entries you made earlier (such as user name and IP address) are displayed at the bottom of the window.

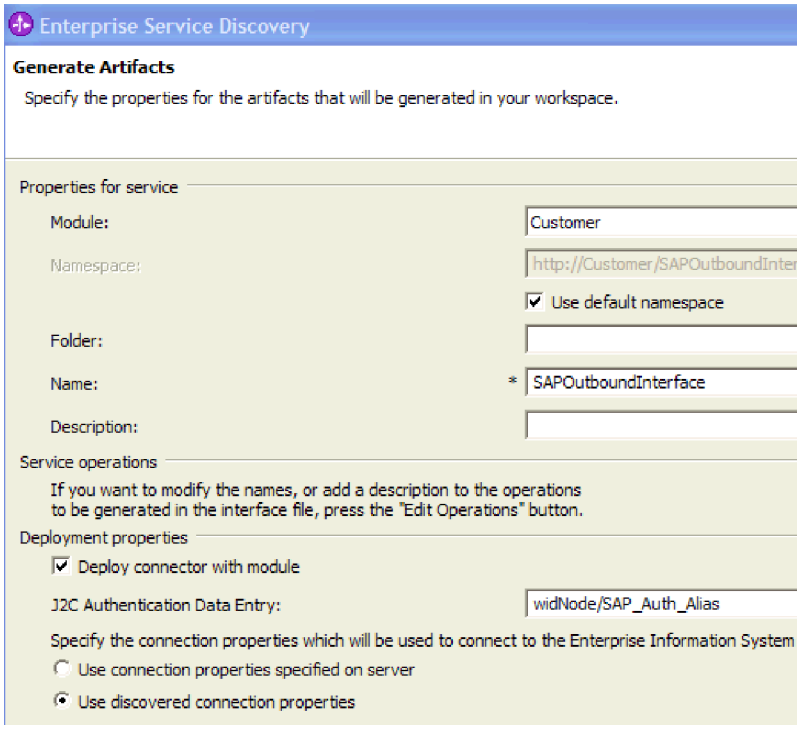


Figure 121. The Generate Artifacts window

4. Indicate the authentication alias to use by typing the alias that you created (in the beginning of the tutorial) in the administrative console. In the example shown earlier, the alias is `widNode/SAP_Auth_Alias`.
5. Click **Finish**.

Result

The new Customer module is added to the Business Integration perspective.

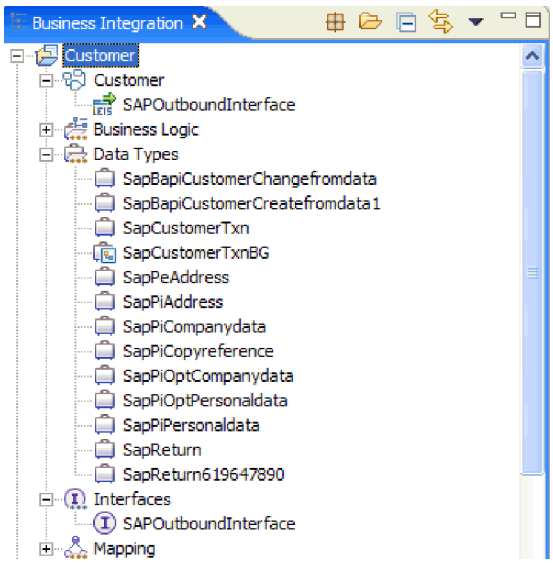


Figure 122. The Customer module displayed in the Business Integration perspective

Deploying the module for testing

To deploy the module to the test environment of WebSphere Process Server, you start the server and add the module (CustomerApp) to it. ″App″ is appended to the name of the module to indicate that the module is a deployable application.

1. Select the test environment server.
   a. Click the **Servers** tab.
   b. Right-click **WebSphere Process Server v6.0**.

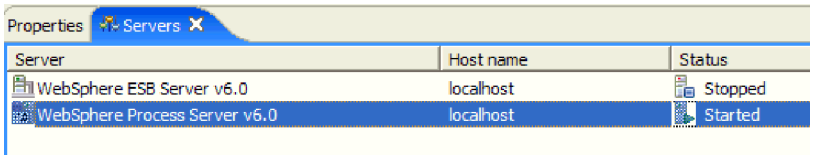


Figure 123. Selecting the WebSphere Process Server test environment from the Servers tab

2. Click **Add and remove projects**.
3. Select **CustomerApp** and click **Add**.
4. Click **Finish**.

Result

You see status messages in the **Console** tab as CustomerApp is deployed to the server.

Testing the module

Test the module to make sure you can perform the transaction on the SAP server. You create and then change customer data and view the results to make sure the transaction was successfully processed.

About this task

To test this tutorial, you use actual values that exist on your SAP server. If you have not already done so, obtain actual values for the following data. If necessary, see your SAP administrator to obtain the data.

- Sales Organization
- Distribution Channel
- Division
- Reference Customer

How to perform this task

1. In the Business Integration perspective, begin the testing procedure by right-clicking **Customer** and clicking **Test** → **Test Module**.
2. In the **verb** row, select **Create** from the list.
3. Enter values for the CUSTOMERCREATEFROMDATA1 BAPI.
   a. In the Initial request parameters window, scroll to **SapPiPersonaldata** under **SapBapiCustomerCreatefromdata1**.

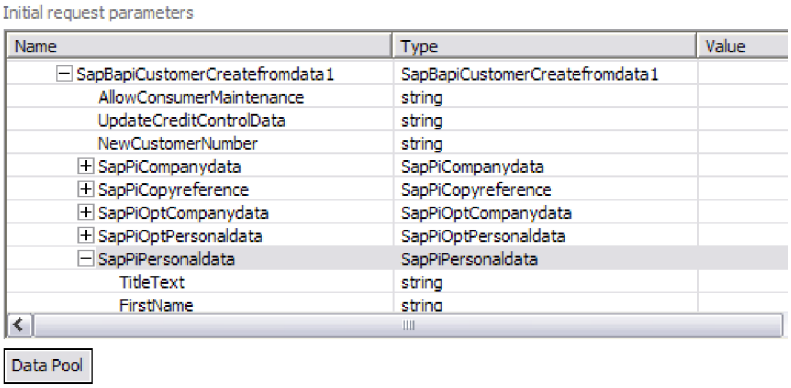


Figure 124. The SapPiPersonaldata property

b. Type values for the following properties. When the **Value** field shows *user data*, enter any value (for example, enter your own name for **FirstName**):

Table 17. Values for SapPiPersonaldata

Property	**Value**
FirstName	*user data*
LastName	*user data*
City	*user data*
CityPostalCode	*user data*
HouseNumber	*user data*
CountryKey	US
RegionStateProvinceCounty	CA
LanguageKey	EN
CurrencyKey	EUR

c. Scroll up to **SapPiCopyreference**, under **SapBapiCustomerCreatefromdata1**.

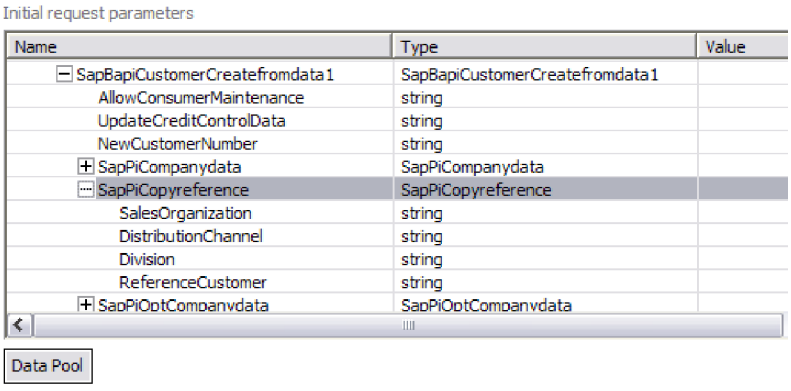


Figure 125. The SapPiCopyreference property

d. Type values for the following properties. Use the values you obtained from your SAP administrator.

Table 18. Values for SapPiCopyreference

Property
SalesOrganization
DistributionChannel
Division
ReferenceCustomer

4. Enter values for the CUSTOMERCREATEFROMDATA1 BAPI.
   a. Scroll up to **SapBapiCustomerChangefromdata**.

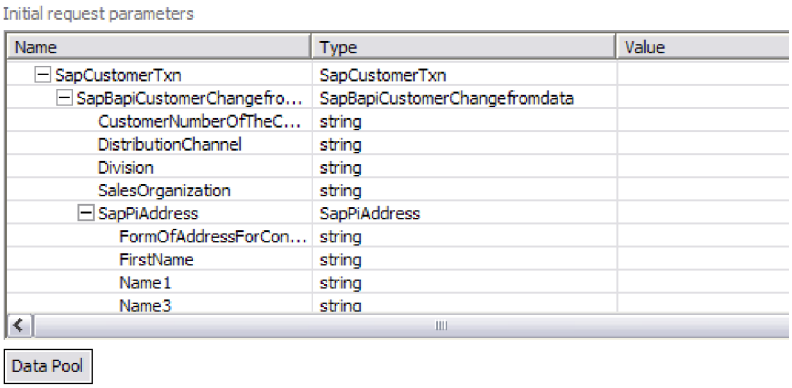


Figure 126. The SapBapiCustomerChangefromdata property

   b. Type values for the following properties. Use the values you obtained from your SAP administrator.

Table 19. Values for SapBapiCustomerChangefromdata

Property
CustomerNumberoftheChangedCustomer
DistributionChannel
Division
SalesOrganization

   c. For **SapPiAddress**, type the following values:

Table 20. Values for SapPiAddress

Property	Value
FirstName	*user data*
Name	*user data*
HouseNumberAndStreet	*user data*
PostalCode	*user data*
City	*user data*
CountryKey	US
RegionStateProvinceCounty	CA
LanguageKey	EN
CurrencyKey	EUR

5. Click **Continue**.
6. Click **Finish**. The integration test client invokes the transaction.
7. Scroll down to the Return parameters window to see the results.

Result

The data on the SAP server has been updated. You can log on to the SAP server to verify that the changes were made.

Tutorial 3: Posting an IDoc to an SAP application

To create a module that invokes an operation to send an IDoc to an SAP application, you create an adapter project, use the enterprise service discovery wizard to generate business objects based on the IDoc, and create a module that contains WebSphere Adapter for SAP Software and the newly generated business objects. You then deploy the module to the test environment of WebSphere Integration Developer.

Creating the authentication alias

To create an authentication alias, display the WebSphere Process Server administrative console and specify the user ID and password you use to access the SAP server. The user ID and password are then associated with the authentication alias.

1. Launch WebSphere Integration Developer by clicking **Start** → **Programs** → **IBM WebSphere** → **Integration Developer V6.0.2** → **WebSphere Integration Developer V6.0.2**.
2. If you are prompted to specify a workspace, accept the default value.

   The workspace is a directory where WebSphere Integration Developer stores your project.
3. When the WebSphere Integration Developer window is displayed, close the Welcome page.
4. Switch to the Business Integration Perspective by clicking **Window** → **Open Perspective**. Then click **Business Integration (default)** and click **OK**.
5. Display the administrative console.
   a. Click the **Servers** tab.
   b. If **WebSphere Process Server v6.0** does not show a status of **Started**, right-click **WebSphere Process Server v6.0** and click **Start**.
   c. Right-click **WebSphere Process Server v6.0** and click **Run administrative console**.

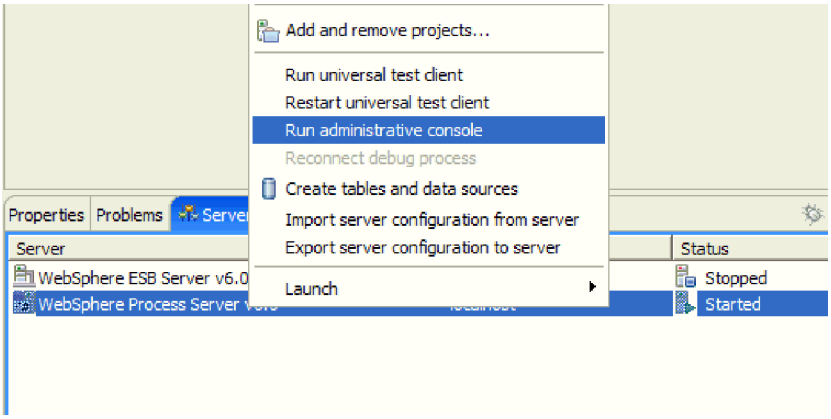


Figure 127. Selecting Run administrative console for the WebSphere Process Server

d. Log in to the administrative console by typing **admin** and clicking **Log in**.

6. In the WebSphere Process Server administrative console, click **Security** → **Global security**.

Figure 128. The Security item on the administrative console

7. Under **Authentication**, click **JAAS Configuration** → **J2C Authentication data**.

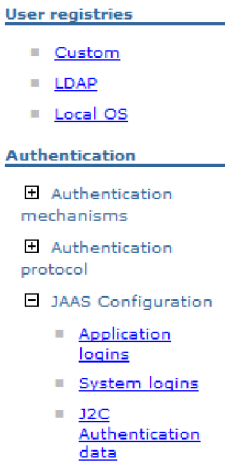


Figure 129. The Authentication section of the administrative console

8. If an alias named **SAP_Auth_Alias** does not already exist, create it now.
   a. Determine from your SAP administrator whether the authentication alias is case-sensitive (for example, whether the alias must be entered in uppercase).
   b. Click **New**.
   c. In the General properties window, type `SAP_Auth_Alias` in the **Alias** field.

      Note: If your SAP server requires that the alias be entered in a specific format (for example, all uppercase), type the alias according to that format.
   d. Type the user ID and password that are required to connect to the SAP server.

      Note: If your SAP server requires that the password be entered in a specific format (for example, all uppercase), type the password according to that format.
   e. Click **OK**.

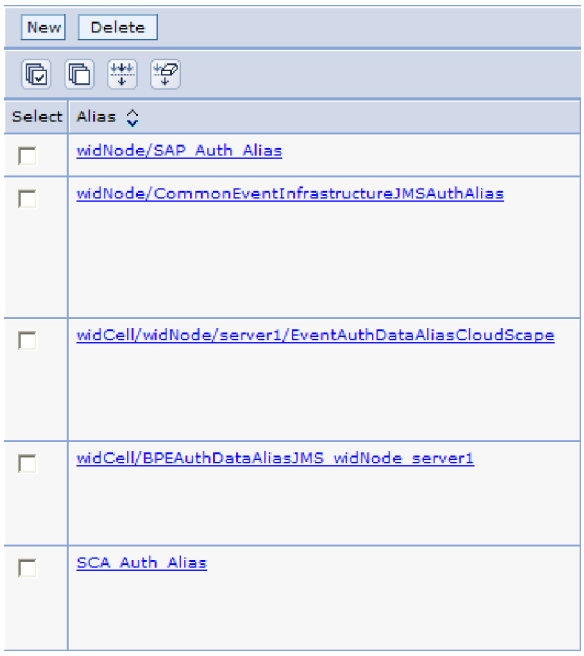


Figure 130. The list of aliases, including the newly created SAP_Auth_Alias

Make note of the name as it appears in the Alias list. In the example, the name is **widNode/SAP_Auth_Alias**. This name is the one you will use in subsequent configuration windows.

f. Click **Save**.

Result

You have created an authentication alias, which you will use when you configure the adapter properties.

Creating the adapter project in WebSphere Integration Developer

To begin the process of creating a module to communicate with an SAP service, you create an adapter project. The adapter project (called a *connector project* in WebSphere Integration Developer) 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.

About this task

You can use the same adapter project for multiple tutorials. If you have already created an adapter project by importing the adapter RAR file, you do not need to create it again, unless you want to have separate adapter projects for each tutorial.

How to perform this task

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**.

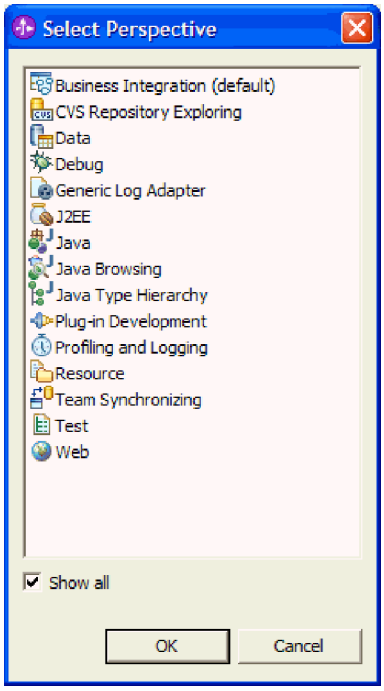


Figure 131. Selecting J2EE from the Select Perspective list

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

d. Click **OK**.

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

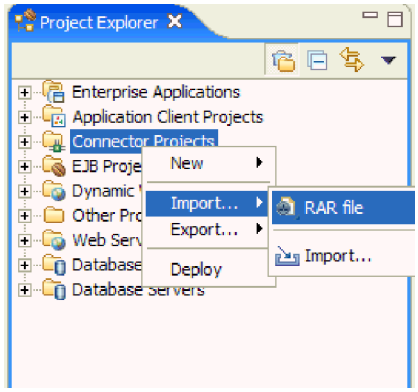


Figure 132. Importing the RAR file

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

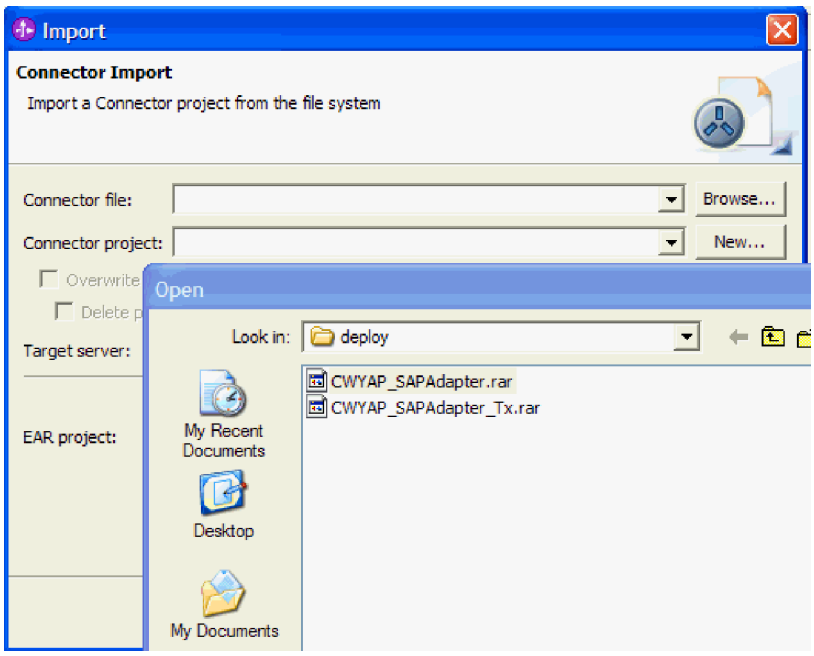


Figure 133. Selecting the RAR file from the installation directory

4. Select the RAR file and click **Open**.
5. Accept the default setting (**CWYAP_SAPAdapter**) for **Connector project**.

   The connector project has the same name as the RAR file.

   If a project named CWYAP_SAPAdapter.rar already exists in this workspace, the name in the **Connector project** field has a number appended to it (for example, CWYAP_SAPAdapter1).
6. 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.
7. Clear the **Add module to an EAR project** check box.

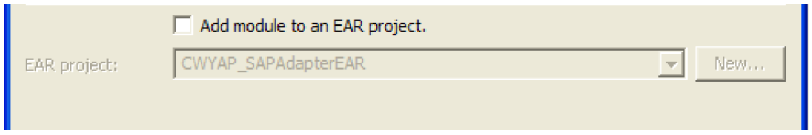


Figure 134. Clearing the Add module to an EAR project check box

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

Result

A new adapter project, named CWYAP_SAPAdapter, is created. To see its contents, expand **CWYAP_SAPAdapter**.

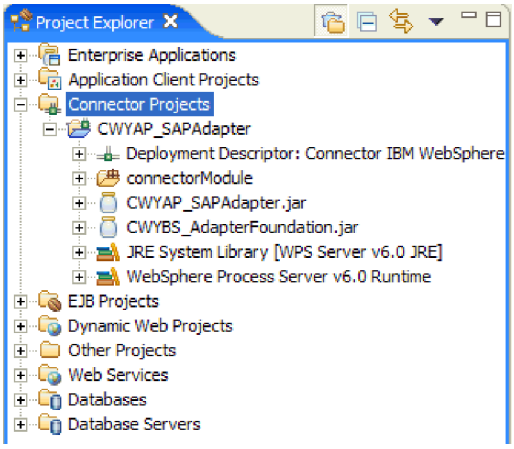


Figure 135. The CWYAP_SAPAdapter project in the Project Explorer window

Adding external dependencies

To add the required external dependency files, you copy the files, including the sapjco.jar file, to directories within the WebSphere Integration Developer directory. You then add the sapjco.jar file to the adapter project you created.

1. If you have not already done so as part of the installation of the adapter or as part of running another tutorial, copy the required files as outlined in the following steps.
   a. Obtain the files for your operating system from your SAP administrator or from the SAP Web site.

Table 21. Files to be installed

Operating system	Files to be installed
Windows	Any *.dll files that come with the SAP Jco download from the SAP Web site
Unix (including Unix System Services on z/OS)	Any .so and .o files that come with the SAP Jco download from the SAP Web site

   b. Copy the files to the following locations in the WebSphere Integration Developer installation directory:
      - \runtimes\bi_v6\java\bin
      - \eclipse\jre\bin

      For z/OS, add the files to the ${*WAS_INSTALL_ROOT*}/lib directory.
   c. For Windows environments only, obtain the msvcp71.dll and msvcr71.dll files from your SAP administrator or the SAP Web site.
   d. For Windows environments only, install the msvcp71.dll and msvcr71.dll files in the Windows system path.
   e. Obtain the sapjco.jar file from your SAP administrator or the SAP Web site.
   f. Copy sapjco.jar to the following location in the WebSphere Integration Developer installation directory: \runtimes\bi_v6\lib

      For z/OS, add ${*WAS_INSTALL_ROOT*}/lib/sapjco.jar to WAS_SERVER_ONLY_server_region_classpath
2. Import the sapjco.jar file into the adapter project.

a. In the J2EE perspective of WebSphere Integration Developer, expand **Connector Projects**.
b. Right-click **CWYAP_SAPAdapter** and click **Properties**.

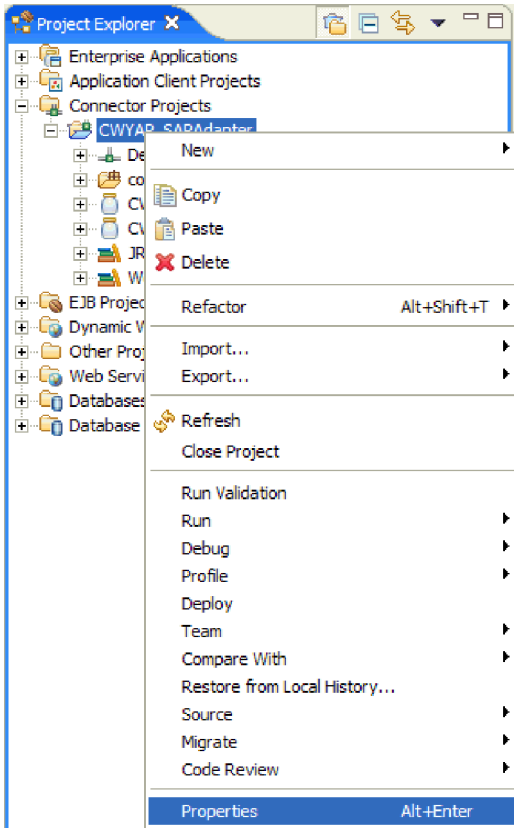


Figure 136. The CWYAP_SAPAdapter project, displayed in the Project Explorer

c. In the left side of the Properties for CWYAP_SAPAdapter window, click **Java Build Path**.

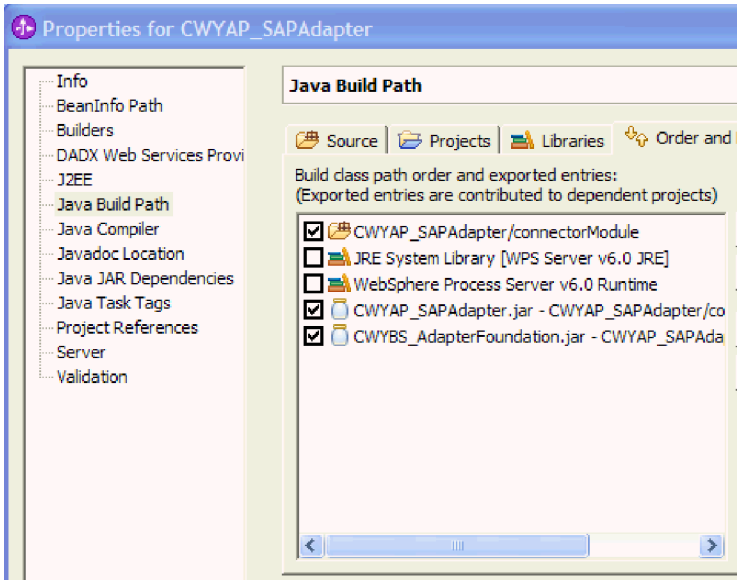


Figure 137. Selecting Java Build Path

d. Click the **Libraries** tab, and click **Add External JARs**.

e. Navigate to the directory on your local file system where the sapjco.jar file is located. Then select **sapjco.jar** and click **Open**.

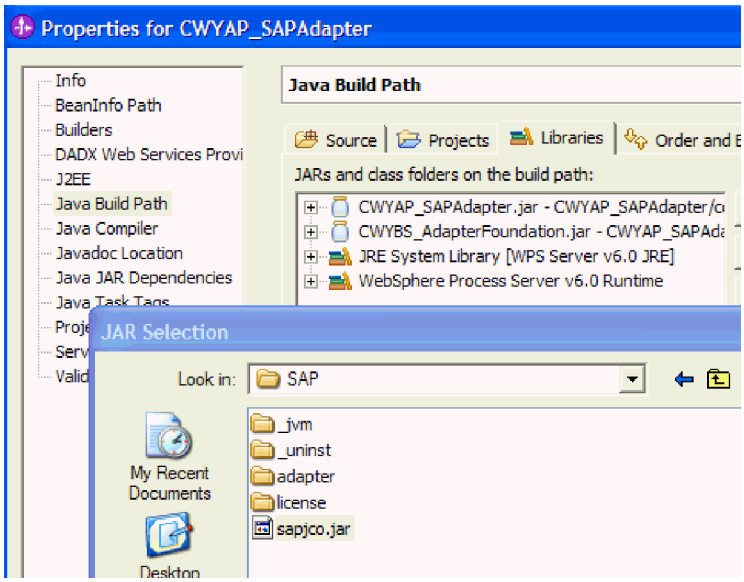


Figure 138. The JAR Selection window, with the sapjco.jar file highlighted for selection

f. Click **OK**.

The file sapjco.jar appears in the list of JARs and class folders in the build path.

Result

The sapjco.jar file is now part of your connector project and appears in the Project Explorer window of WebSphere Integration Developer.

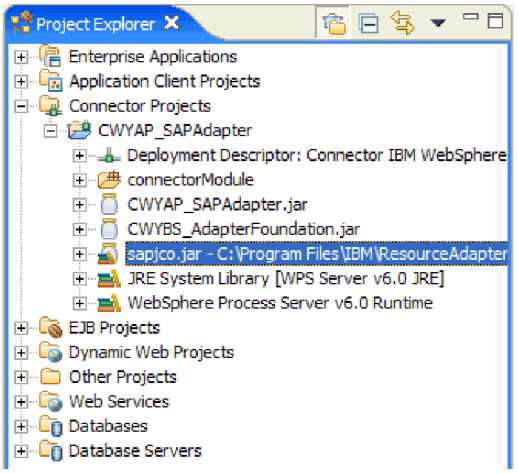


Figure 139. The Project Explorer window of WebSphere Integration Developer

Configuring the adapter for outbound processing

To configure the adapter, set the connection properties for enterprise service discovery. Then use the enterprise service discovery wizard to select and configure the necessary business objects and to generate a deployable module.

Setting connection properties for enterprise service discovery

To set connection properties for the enterprise service discovery wizard so that it can access the SAP server, specify such information as the user name and password you use to access the server as well as the name or IP address of the server.

1. In WebSphere Integration Developer, start the enterprise service discovery wizard.
   a. Switch to the Business Integration Perspective by clicking **Window** → **Open Perspective** → **Other**. Then click **Business Integration (default)** and click **OK**.
   b. Right-click **New** → **Enterprise Service Discovery**.

      If **Enterprise Service Discovery** is not displayed, click **New** → **Other**, expand **Business Integration**, and click **Enterprise Service Discovery**. Then click **Next**.

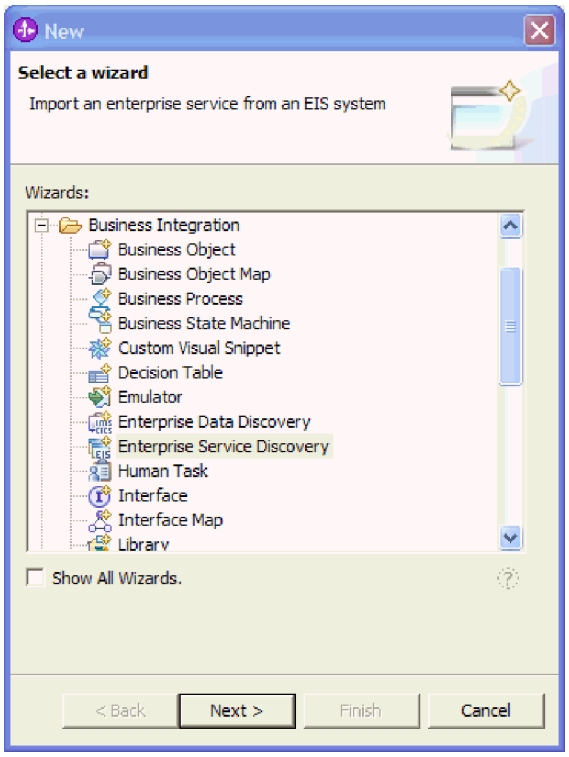


Figure 140. The expanded list of wizards

2. In the Select an Enterprise Service Resource Adapter window, make sure **IBM WebSphere Adapter for SAP Software** is selected, and click **Next**.
3. In the Configure Settings for Discovery Agent window, specify the configuration properties needed to connect to the SAP server.
   a. Type the name and password you use to access the SAP server.

      The password is case-sensitive.
   b. Type your client ID.

      This is typically `100`.
   c. In the **Application Server Host** field, type the name (or IP address) of your SAP server.

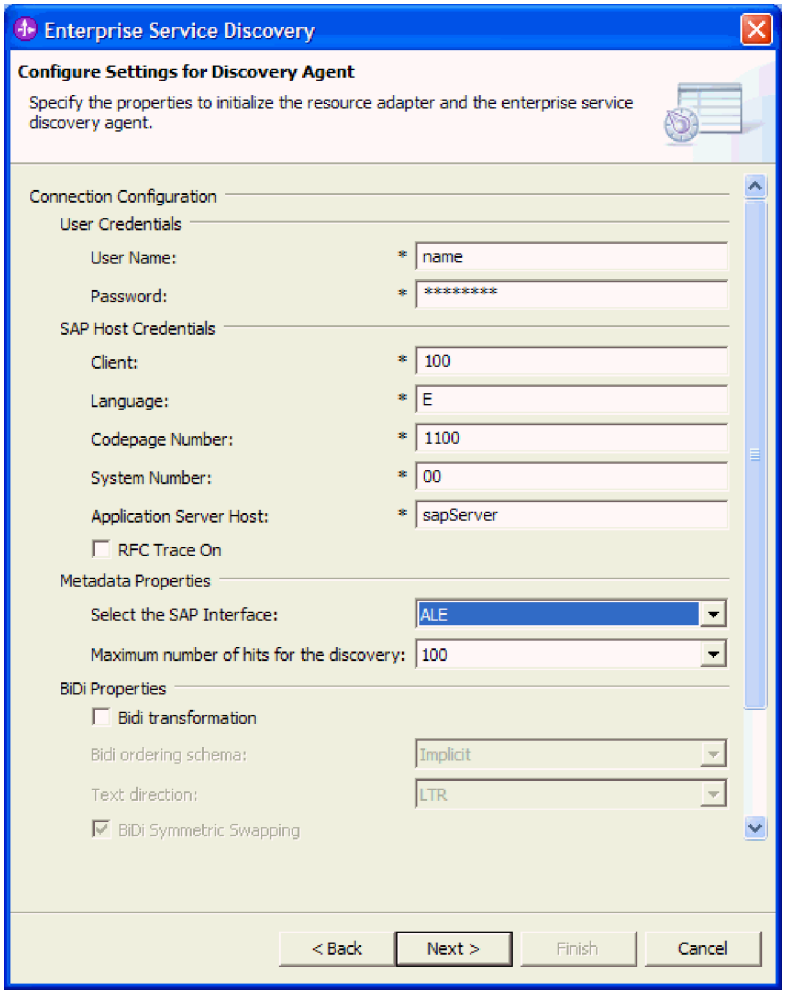


Figure 141. The Configure Settings for Discovery Agent window

4. Make sure **ALE** (the default) is displayed in the **Select the SAP interface** field.
5. Set the logging level so that you can see any errors that might arise during enterprise service discovery.
   a. At the bottom of the Configure Settings for Discovery Agent window, click **Show Advanced**.

      The button changes to **Hide Advanced**.

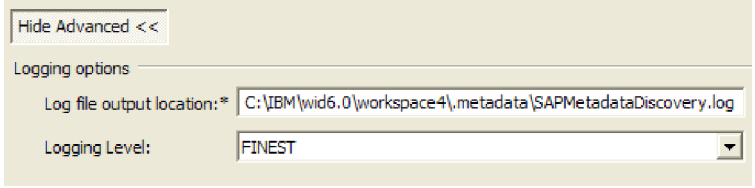


Figure 142. The Logging options displayed when you select Show Advanced

   b. For **Logging Level**, select **FINEST**.
6. Click **Next**.

Result

The enterprise service discovery wizard contacts the SAP server, using the information you provided (such as user name and password) to log in. You see the Find and Discover Enterprise Services window.

Selecting business objects and services

To select the ALE IDoc, specify search criteria (such as the name of the IDoc). The enterprise service discovery wizard uses the search criteria to find the IDoc on the SAP server.

1. From the Find and Discover Enterprise Services window, begin the process of searching for services by clicking **Execute Query**.
2. Expand **ALE**, **Discover IDoc From System**, and **Basic IDocs**, and click **Discover By Name**.

   The **Filter** button is now enabled.

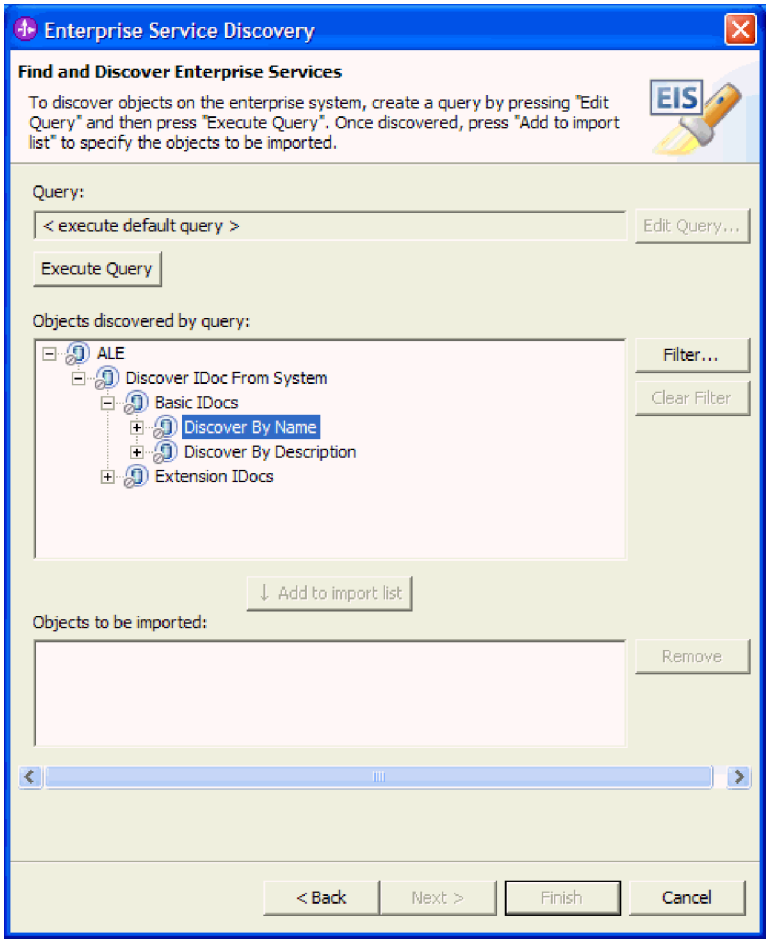


Figure 143. The Find and Discover Enterprise Services window

3. Click **Filter**.
4. In the Filter Properties for Discover By Name window, indicate that you want the ALEREQ1 IDoc by typing `alereq01`.

Figure 144. The Filter Properties for Discover by Name window

5. Click **OK**.
6. Select the IDoc.
   a. Expand **Discover By Name (filtered)**.
   b. Click **ALEREQ01** and click **Add to import list**.

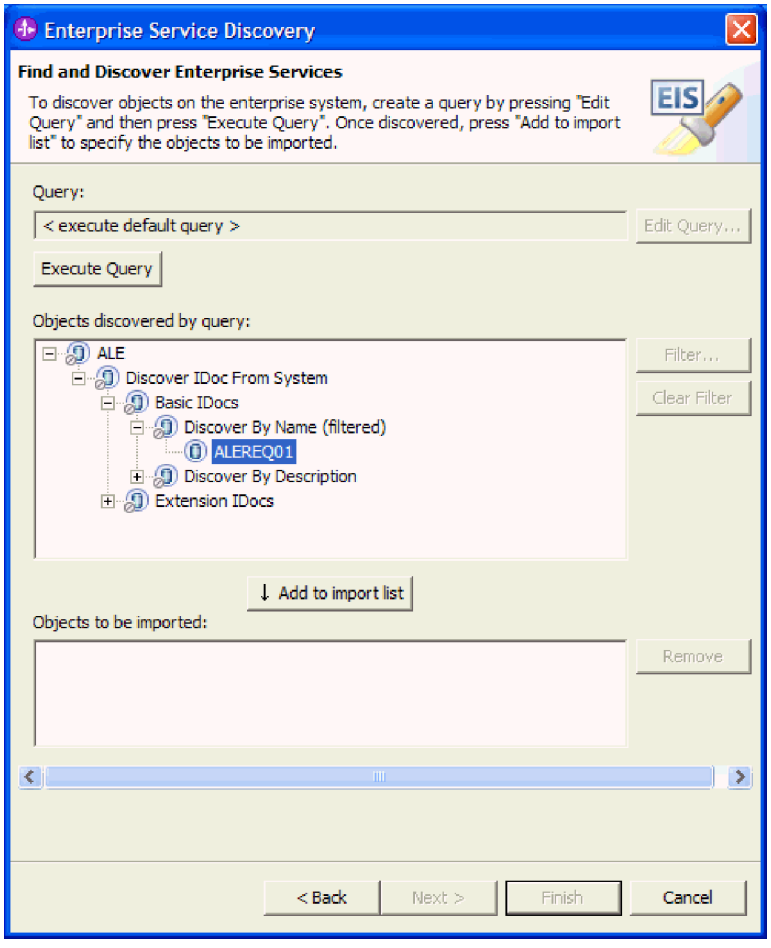


Figure 145. The Find and Discover Enterprise Services window

7. In the Configuration Parameters for ALEREQ01 window, select the **Use SAP Field Name to generate attribute(s)** check box.

8. In the **Enter the Release** field, accept the default SAP release number.
9. Click **OK**.
10. Click **Next**.

Result

The enterprise service discovery wizard has found ALEREQ01, and you are ready to configure it.

Configuring the selected objects

To configure the business object, you specify information about the object (such as the location of the object and the operation associated with the object).

1. In the Configure Objects window, specify the directory where the object should be stored and indicate that the IDoc is used for outbound processing.
   a. In the **Object Location (Enter relative Path)** field, type `bodefs` as the name of the directory.
   b. Select **Outbound** from the **Service Type** field.

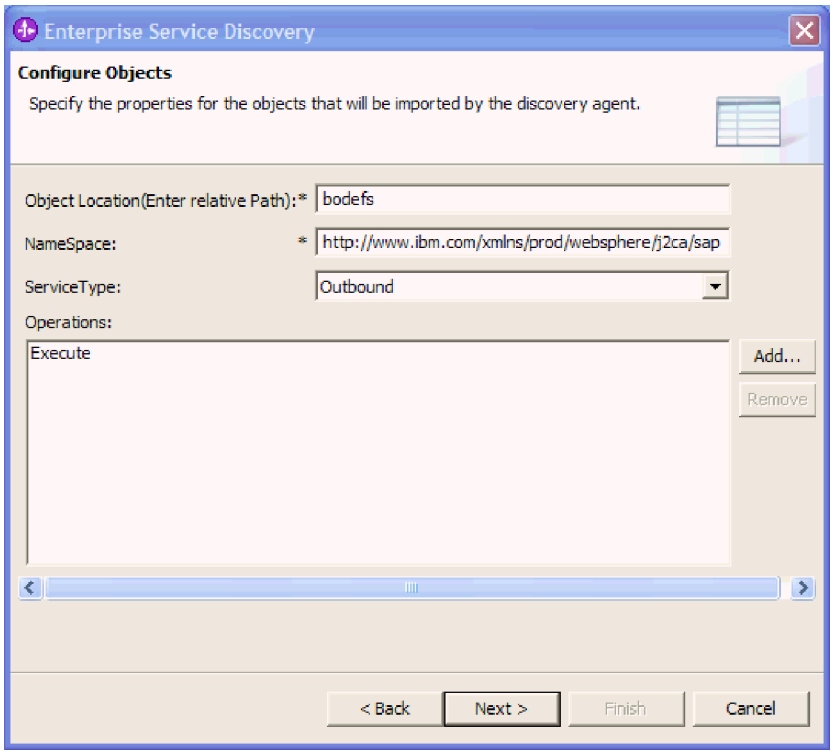


Figure 146. The Configure Objects window, with the sample values entered

When you select **Outbound**, the list of operations changes. Execute is the only operation you can use for outbound processing.

2. Click **Next**.

Result

You have associated an operation (Execute) with the object and selected a location for the object. The Generate Artifacts window is displayed.

Generating artifacts

To generate the module, which is the artifact that can be exported to an EAR file for deployment, you create the module, include the adapter in the module, and specify an alias used to authenticate the caller to the SAP server.

1. In the Generate Artifacts window, create a new module.
   a. Click **New**.
   b. Click **Create a module project** and click **Next**.
   c. Type `alereq01`.

      As you type the name of the module, the name is added to the path next to **Directory**.

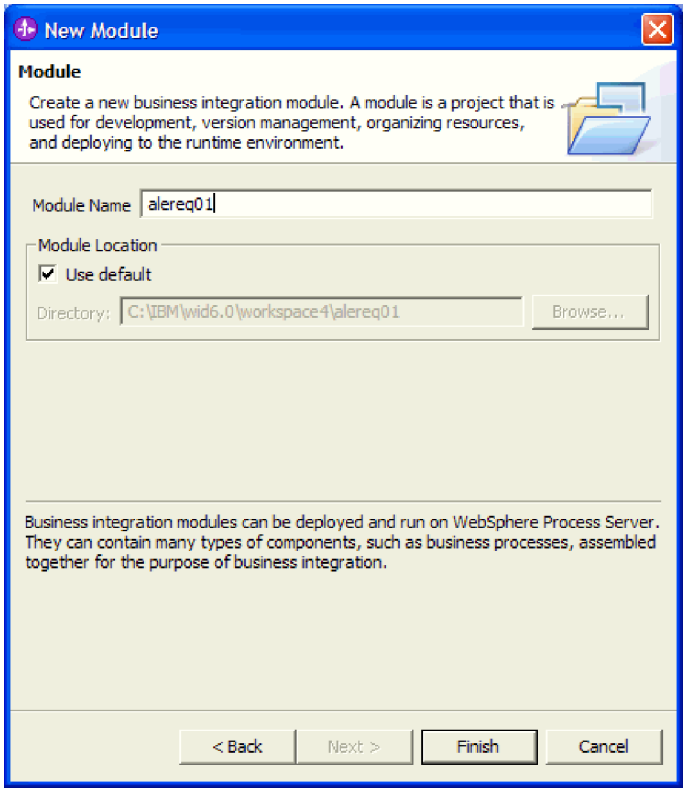


Figure 147. The New Module window

   d. Click **Finish**.
2. Accept the default value for **Name** and **Deploy connector with module**.
3. Indicate the authentication alias to use by typing the alias that you created (in the beginning of the tutorial) in the administrative console. In the example shown earlier, the alias is `widNode/SAP_Auth_Alias`.
4. Select **Use discovered connection properties**.

   When you select **Use discovered connection properties**, the entries you made earlier (such as user name and IP address) are displayed at the bottom of the window.
5. Click **Finish**.

Result

The new module is added to the Business Integration perspective.

Deploying the module for testing

To deploy the module to the test environment of WebSphere Process Server, you start the server and add the module (ALEREQApp) to it. ″App″ is appended to the name of the module to indicate that the module is a deployable application.

1. Select the test environment server.
   a. Click the **Servers** tab.
   b. Right-click **WebSphere Process Server v6.0**.

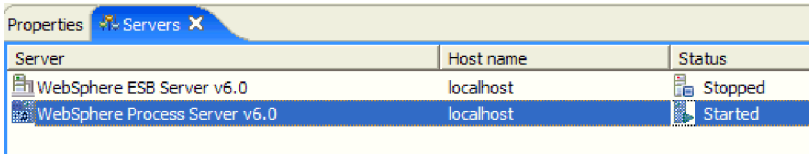


Figure 148. Selecting the WebSphere Process Server test environment from the Servers tab

2. Click **Add and remove projects**.
3. Select **ALEREQApp** and click **Add**.
4. Click **Finish**.

Result

You see status messages in the **Console** tab as ALEREQApp is deployed to the server.

Testing the module

Test the module to make sure you can create the ALEREQ01 IDoc on the SAP server. You enter a verb (Create) and values for the control record and data record of the IDoc. You can then display a list of IDocs on the SAP server to make sure the IDoc was created.

About this task

To test this tutorial, you use data from your SAP server. If you have not already done so, obtain actual values for the following data. If necessary, see your SAP administrator to obtain the data.

- Client
- IdocNumber
- SenderPort
- PartnerNumberofSender
- ReceiverPort
- PartnerNumberofRecipient

How to perform this task

1. In the Business Integration perspective, begin the testing procedure by right-clicking **ALEREQ01** and clicking **Test → Test Module**.
2. In the **verb** row under **executeSapAlereq01Input**, select **Create** from the list.
3. Enter the IDoc Control record data:
   a. Right-click **SapAlereq01IDocBO** and click **Add Element**.

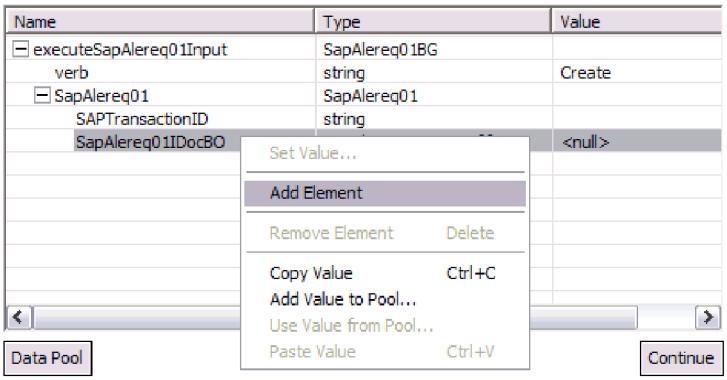


Figure 149. Adding an element to SapAlereq01DocBO

If you are prompted to enter the number of elements to add, select **1** and click **OK**.

b. Type the following values in the associated fields:

Field	**Value**
ReceiverPort	*SAP administrator-supplied value*
PartnerTypeOfSender	LS
NameOfBasicType	ALEREQ01
PartnerNumberOfSender	*SAP administrator-supplied value*
NameOfTableStructure	EDI_DC40
Client	*SAP administrator-supplied value*
LogicalMessageType	ALEREQ
PartnerNumberOfRecipient	*SAP administrator-supplied value*
SenderPort	*SAP administrator-supplied value*
IdocNumber	*SAP administrator-supplied value*
PartnerTypeOfRecipient	LS

4. Set the IDoc Data Record level property values.
   a. Right-click **SapAlereq01DataRecord** and click **Add Element**.
   b. Type the following values in the associated fields:

Field	**Value**
Logicalmessagetype	ALEREQ
Messagetype	ALEREQ

   c. Right-click **SapAlereq01E2aleq1** and click **Add Element**.
   d. Type the following values in the associated fields:

Field	**Value**
IncludingExcludingindicator	E
RelationaloperatorEqNeGtLtGeLe	LT
Lowerlimitforfieldcontents	0
Upperlimitforfieldcontents	100

5. Click **Continue**.
6. In the Select Deployment window, select **WebSphere Process Server v6.0**, and click **Finish**.

Result

The IDoc is created on the SAP server. You can verify that the IDoc was created by starting the SAP server and running the WE02 transaction.

Tutorial 4: Receiving an IDoc packet

To create a module that receives a split IDoc packet, you create an adapter project, use the enterprise service discovery wizard to generate business objects based on the IDoc, and create a module that contains WebSphere Adapter for SAP Software and the newly generated business objects. You then deploy the module to the test environment of WebSphere Integration Developer.

Creating the authentication alias

To create an authentication alias, display the WebSphere Process Server administrative console and specify the user ID and password you use to access the SAP server. The user ID and password are then associated with the authentication alias.

1. Launch WebSphere Integration Developer by clicking **Start → Programs → IBM WebSphere → Integration Developer V6.0.2 → WebSphere Integration Developer V6.0.2**.
2. If you are prompted to specify a workspace, accept the default value.

   The workspace is a directory where WebSphere Integration Developer stores your project.
3. When the WebSphere Integration Developer window is displayed, close the Welcome page.
4. Switch to the Business Integration Perspective by clicking **Window → Open Perspective**. Then click **Business Integration (default)** and click **OK**.
5. Display the administrative console.
   a. Click the **Servers** tab.
   b. If **WebSphere Process Server v6.0** does not show a status of **Started**, right-click **WebSphere Process Server v6.0** and click **Start**.
   c. Right-click **WebSphere Process Server v6.0** and click **Run administrative console**.

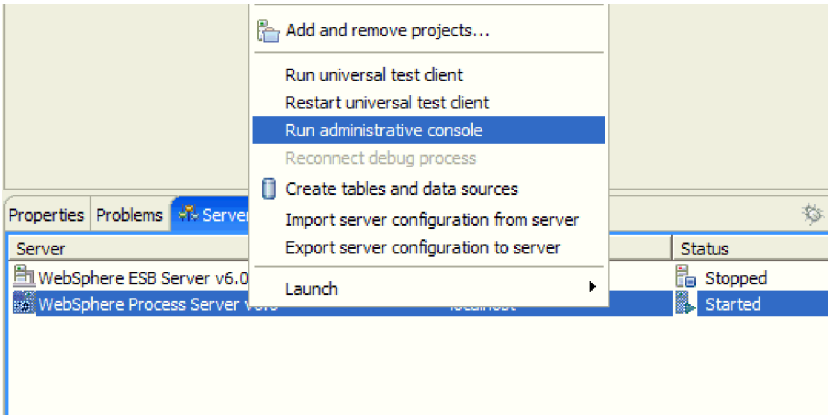


Figure 150. Selecting Run administrative console for the WebSphere Process Server

d. Log in to the administrative console by typing **admin** and clicking **Log in**.

6. In the WebSphere Process Server administrative console, click **Security** → **Global security**.

Figure 151. The Security item on the administrative console

7. Under **Authentication**, click **JAAS Configuration** → **J2C Authentication data**.

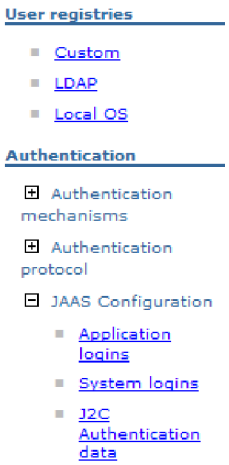


Figure 152. The Authentication section of the administrative console

8. If an alias named **SAP_Auth_Alias** does not already exist, create it now.
   a. Determine from your SAP administrator whether the authentication alias is case-sensitive (for example, whether the alias must be entered in uppercase).
   b. Click **New**.
   c. In the General properties window, type `SAP_Auth_Alias` in the **Alias** field.

      Note: If your SAP server requires that the alias be entered in a specific format (for example, all uppercase), type the alias according to that format.
   d. Type the user ID and password that are required to connect to the SAP server.

      Note: If your SAP server requires that the password be entered in a specific format (for example, all uppercase), type the password according to that format.
   e. Click **OK**.

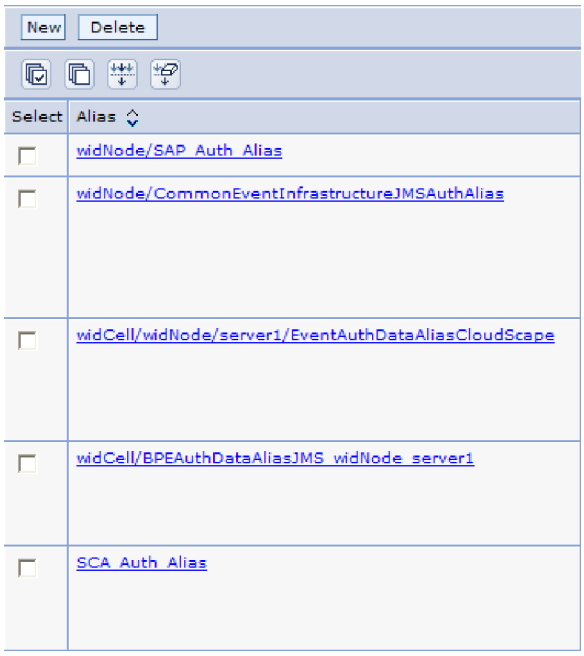


Figure 153. The list of aliases, including the newly created SAP_Auth_Alias

Make note of the name as it appears in the Alias list. In the example, the name is **widNode/SAP_Auth_Alias**. This name is the one you will use in subsequent configuration windows.

f. Click **Save**.

Result

You have created an authentication alias, which you will use when you configure the adapter properties.

Creating the adapter project in WebSphere Integration Developer

To begin the process of creating a module to communicate with an SAP service, you create an adapter project. The adapter project (called a *connector project* in WebSphere Integration Developer) 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.

About this task

You can use the same adapter project for multiple tutorials. If you have already created an adapter project by importing the adapter RAR file, you do not need to create it again, unless you want to have separate adapter projects for each tutorial.

How to perform this task

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**.

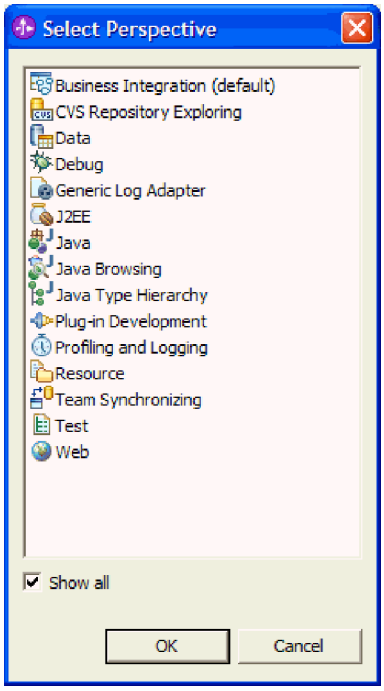


Figure 154. Selecting J2EE from the Select Perspective list

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

d. Click **OK**.

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

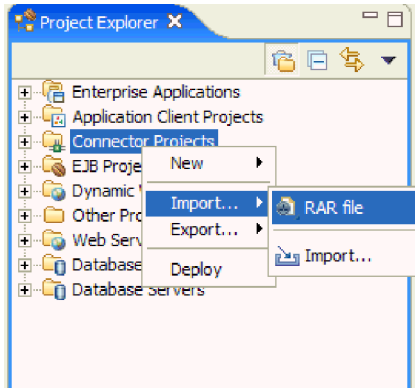


Figure 155. Importing the RAR file

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

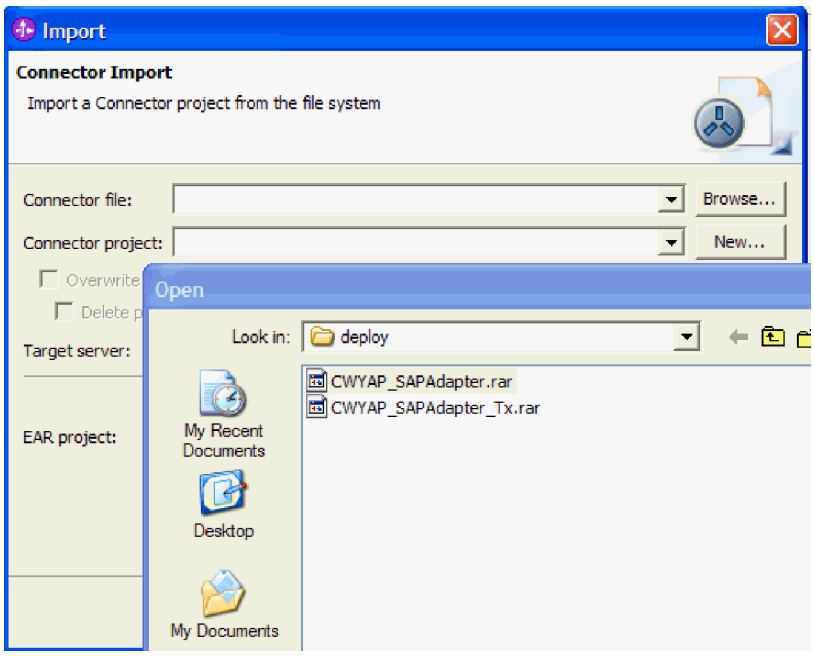


Figure 156. Selecting the RAR file from the installation directory

4. Select the RAR file and click **Open**.
5. Accept the default setting (**CWYAP_SAPAdapter**) for **Connector project**.

   The connector project has the same name as the RAR file.

   If a project named CWYAP_SAPAdapter.rar already exists in this workspace, the name in the **Connector project** field has a number appended to it (for example, CWYAP_SAPAdapter1).
6. 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.
7. Clear the **Add module to an EAR project** check box.

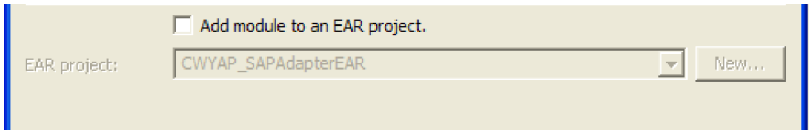


Figure 157. Clearing the Add module to an EAR project check box

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

Result

A new adapter project, named CWYAP_SAPAdapter, is created. To see its contents, expand **CWYAP_SAPAdapter**.

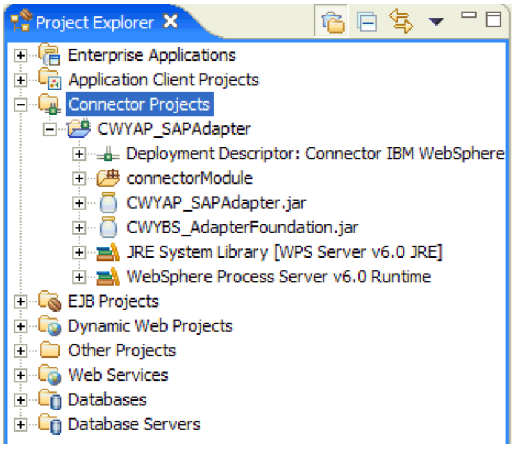


Figure 158. The CWYAP_SAPAdapter project in the Project Explorer window

Adding external dependencies

To add the required external dependency files, you copy the files, including the sapjco.jar file, to directories within the WebSphere Integration Developer directory. You then add the sapjco.jar file to the adapter project you created.

1. If you have not already done so as part of the installation of the adapter or as part of running another tutorial, copy the required files as outlined in the following steps.
   a. Obtain the files for your operating system from your SAP administrator or from the SAP Web site.

Table 22. Files to be installed

Operating system	Files to be installed
Windows	Any *.dll files that come with the SAP Jco download from the SAP Web site
Unix (including Unix System Services on z/OS)	Any .so and .o files that come with the SAP Jco download from the SAP Web site

   b. Copy the files to the following locations in the WebSphere Integration Developer installation directory:
      - \runtimes\bi_v6\java\bin
      - \eclipse\jre\bin

      For z/OS, add the files to the ${*WAS_INSTALL_ROOT*}/lib directory.
   c. For Windows environments only, obtain the msvcp71.dll and msvcr71.dll files from your SAP administrator or the SAP Web site.
   d. For Windows environments only, install the msvcp71.dll and msvcr71.dll files in the Windows system path.
   e. Obtain the sapjco.jar file from your SAP administrator or the SAP Web site.
   f. Copy sapjco.jar to the following location in the WebSphere Integration Developer installation directory: \runtimes\bi_v6\lib

      For z/OS, add ${*WAS_INSTALL_ROOT*}/lib/sapjco.jar to WAS_SERVER_ONLY_server_region_classpath
2. Import the sapjco.jar file into the adapter project.

a. In the J2EE perspective of WebSphere Integration Developer, expand **Connector Projects**.
b. Right-click **CWYAP_SAPAdapter** and click **Properties**.

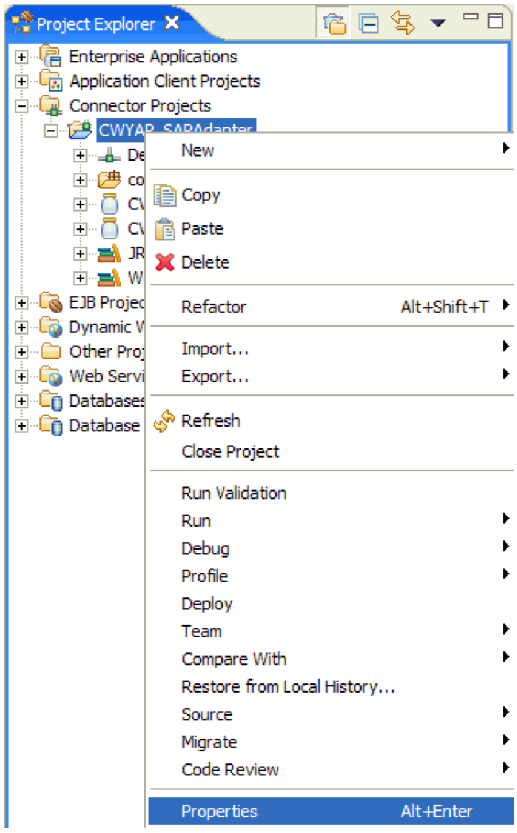


Figure 159. The CWYAP_SAPAdapter project, displayed in the Project Explorer

c. In the left side of the Properties for CWYAP_SAPAdapter window, click **Java Build Path**.

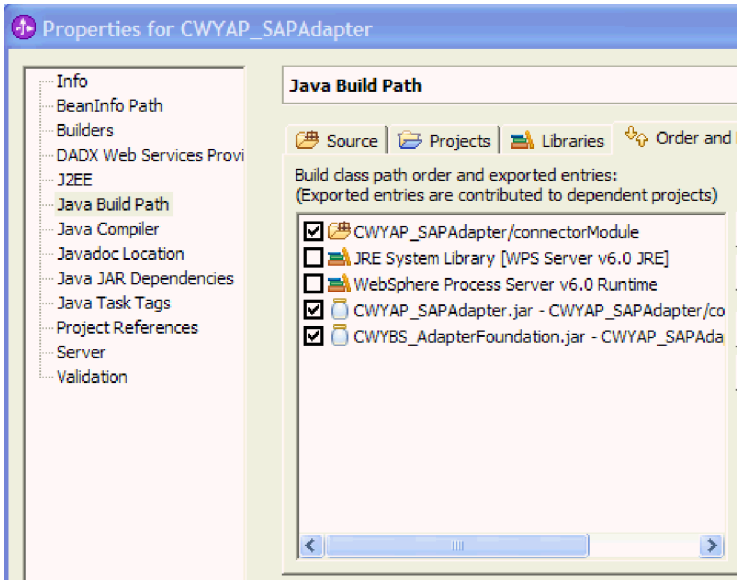


Figure 160. Selecting Java Build Path

d. Click the **Libraries** tab, and click **Add External JARs**.

e. Navigate to the directory on your local file system where the sapjco.jar file is located. Then select **sapjco.jar** and click **Open**.

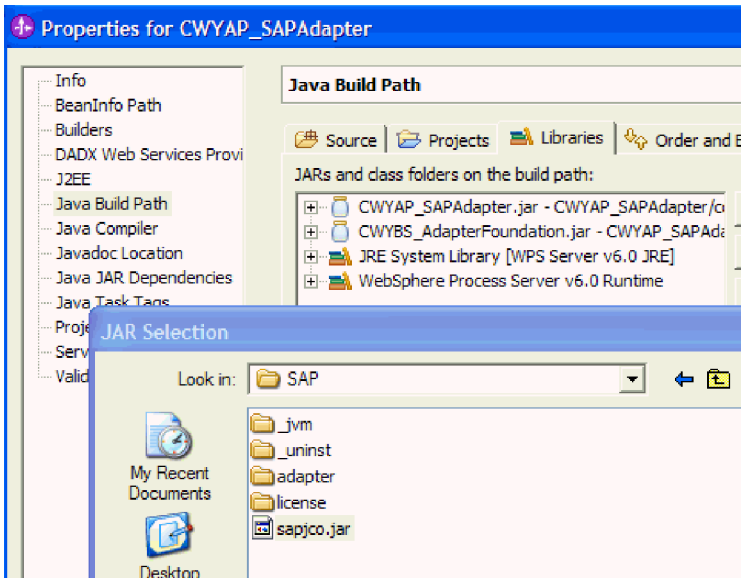


Figure 161. The JAR Selection window, with the sapjco.jar file highlighted for selection

f. Click **OK**.

The file sapjco.jar appears in the list of JARs and class folders in the build path.

Result

The sapjco.jar file is now part of your connector project and appears in the Project Explorer window of WebSphere Integration Developer.

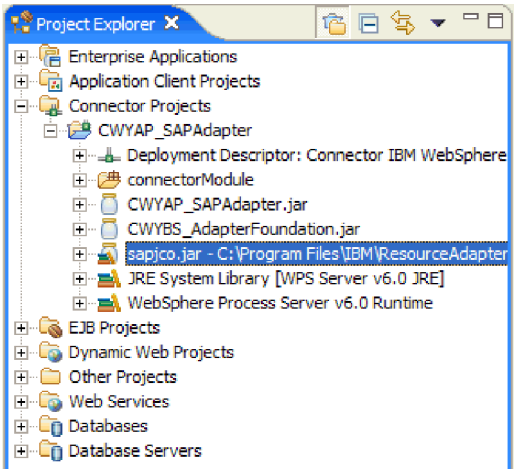


Figure 162. The Project Explorer window of WebSphere Integration Developer

Configuring the data source

To configure the JDBC data source for ALE inbound processing, you use the administrative console. The data source, which is required for inbound processing, is used for event tracking and recovery. This tutorial uses the Cloudscape JDBC provider.

1. Begin the process of creating a data source by displaying the administrative console.
   a. From the J2EE Perspective of WebSphere Integration Developer, click the **Server** tab.
   b. If **WebSphere Process Server v6.0** does not show a status of **Started**, right-click **WebSphere Process Server v6.0** and click **Start**.
   c. Right-click **WebSphere Process Server v6.0** and click **Run administrative console**.

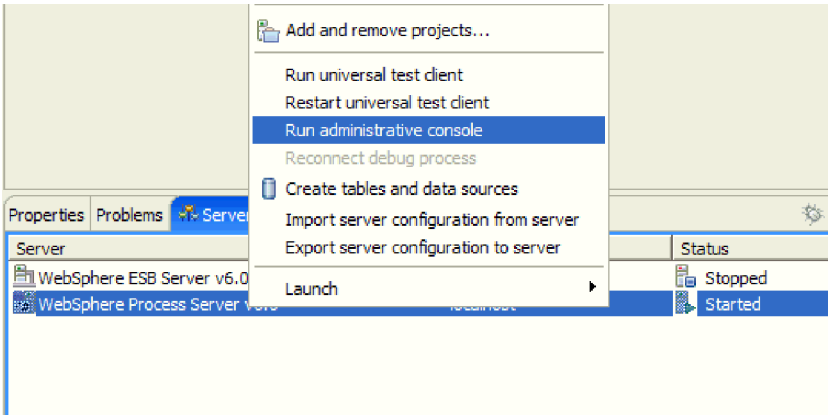


Figure 163. Selecting Run administrative console for the WebSphere Process Server

   d. Log in to the administrative console by typing **admin** and clicking **Log in**.
2. Select a JDBC provider.
   a. Click **Resources** → **JDBC Providers**.
   b. Click **Cloudscape JDBC Provider**.

3. Select **Data sources**.

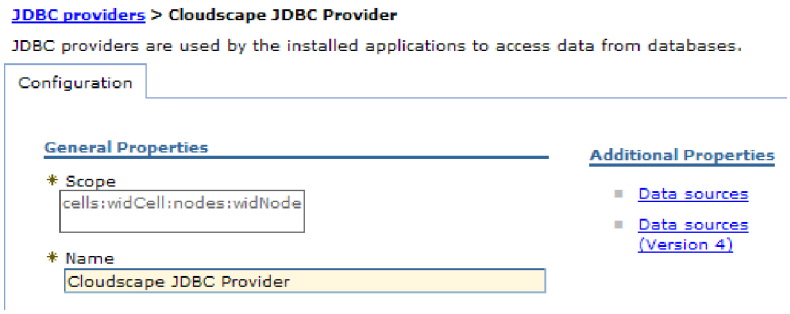


Figure 164. The Cloudscape JDBC provider Configuration tab

4. Create a new data source by clicking **New**.
5. Type the following values in the specified field. Leave the default values for the rest of the configurations.

Option	Description
Name	ALEEventRecoveryDS
JNDI Name	jdbc/ALEEventRecovery
Description	ALEEventRecoveryDS
DatabaseName	ALEEventRecoveryDB

6. Click **Apply**.
   After the changes are applied, **Custom properties** becomes active.
7. Click **Custom properties**.
8. Scroll down and click **createDatabase**.

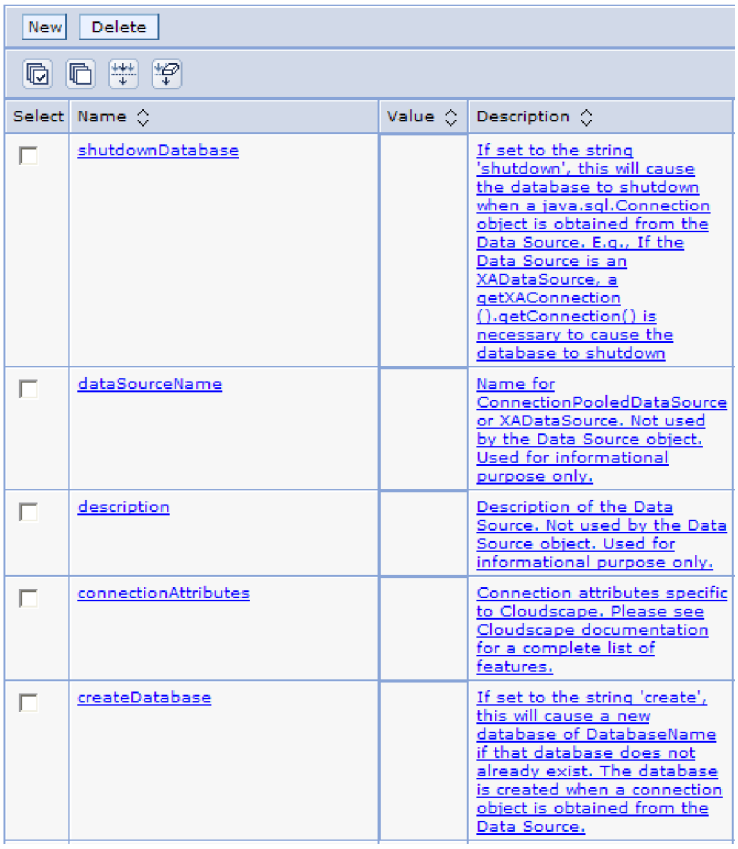


Figure 165. Selecting the createDatabase entry

9. Type `create` in the **value** field, and click **Apply**.
10. Save your configurations.

Result

The new data source, ALEEventRecoveryDS, is displayed in the list of data sources.

Configuring the adapter for inbound processing

To configure the adapter, set the connection properties for enterprise service discovery. Then use the enterprise service discovery wizard to select and configure the necessary business objects and to generate a deployable module.

Setting connection properties for enterprise service discovery

To set connection properties for the enterprise service discovery wizard so that it can access the SAP server, specify such information as the user name and password you use to access the server as well as the name or IP address of the server.

1. In WebSphere Integration Developer, start the enterprise service discovery wizard.
   a. Switch to the Business Integration Perspective by clicking **Window** → **Open Perspective** → **Other**. Then click **Business Integration (default)** and click **OK**.
   b. Right-click **New** → **Enterprise Service Discovery**.

If **Enterprise Service Discovery** is not displayed, click **New** → **Other**, expand **Business Integration**, and click **Enterprise Service Discovery**. Then click **Next**.

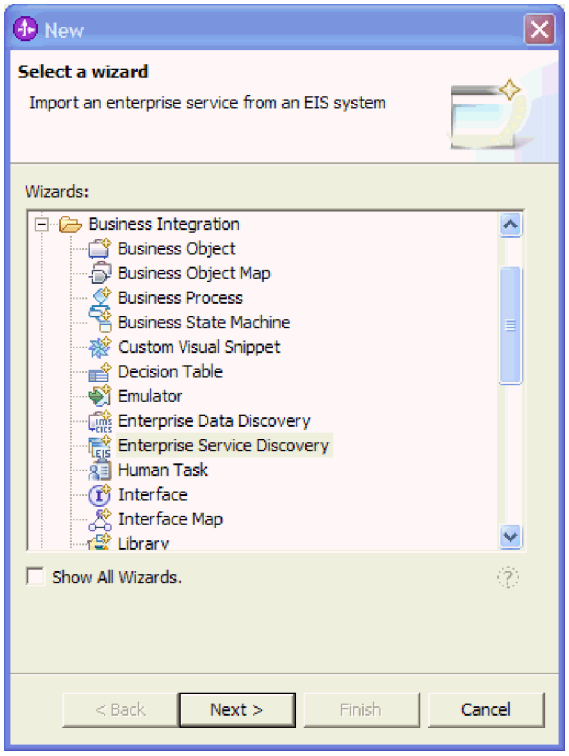


Figure 166. The expanded list of wizards

2. In the Select an Enterprise Service Resource Adapter window, make sure **IBM WebSphere Adapter for SAP Software** is selected, and click **Next**.
3. In the Configure Settings for Discovery Agent window, specify the configuration properties needed to connect to the SAP server.
   a. Type the name and password you use to access the SAP server.

      The password is case-sensitive.
   b. Type your client ID.

      This is typically `100`.
   c. In the **Application Server Host** field, type the name (or IP address) of your SAP server.

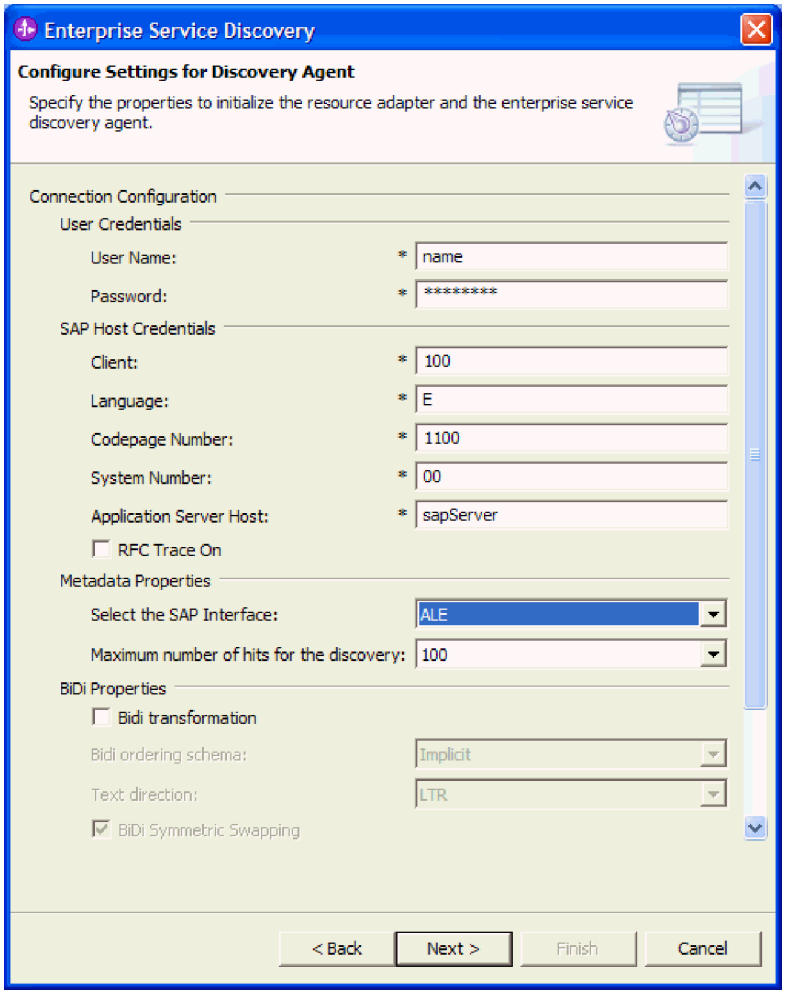


Figure 167. The Configure Settings for Discovery Agent window

4. Make sure **ALE** (the default) is displayed in the **Select the SAP interface** field.
5. Set the logging level so that you can see any errors that might arise during enterprise service discovery.
   a. At the bottom of the Configure Settings for Discovery Agent window, click **Show Advanced**.

      The button changes to **Hide Advanced**.

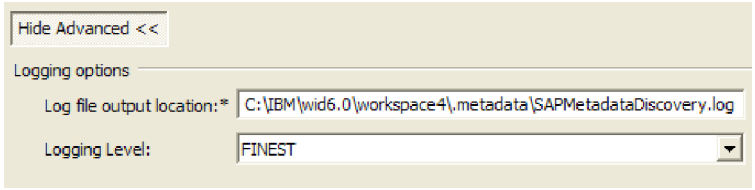


Figure 168. The Logging options displayed when you select Show Advanced

   b. For **Logging Level**, select **FINEST**.
6. Click **Next**.

Result

The enterprise service discovery wizard contacts the SAP server, using the information you provided (such as user name and password) to log in. You see the Find and Discover Enterprise Services window.

Selecting business objects and services

To select the ALE IDoc, specify search criteria (such as the name of the IDoc). The enterprise service discovery wizard uses the search criteria to find the IDoc on the SAP server.

1. From the Find and Discover Enterprise Services window, begin the process of searching for services by clicking **Execute Query**.
2. Expand **ALE**, **Discover IDoc From System**, and **Basic IDocs**, and click **Discover By Name**.

   The **Filter** button is now enabled.

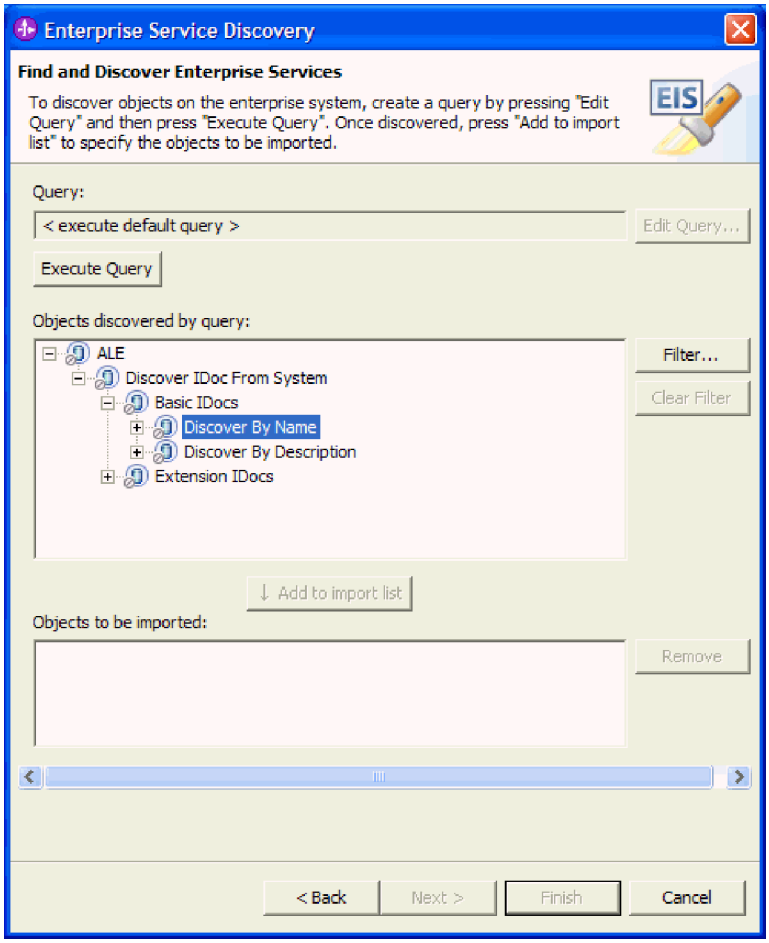


Figure 169. The Find and Discover Enterprise Services window

3. Click **Filter**.
4. In the Filter Properties for Discover By Name window, indicate that you want the ALEREQ1 IDoc by typing `alereq01`.

Figure 170. The Filter Properties for Discover By Name window

5. Click **OK**.
6. Select the IDoc.
   a. Expand **Discover By Name (filtered)**.
   b. Click **ALEREQ01** and click **Add to import list**.

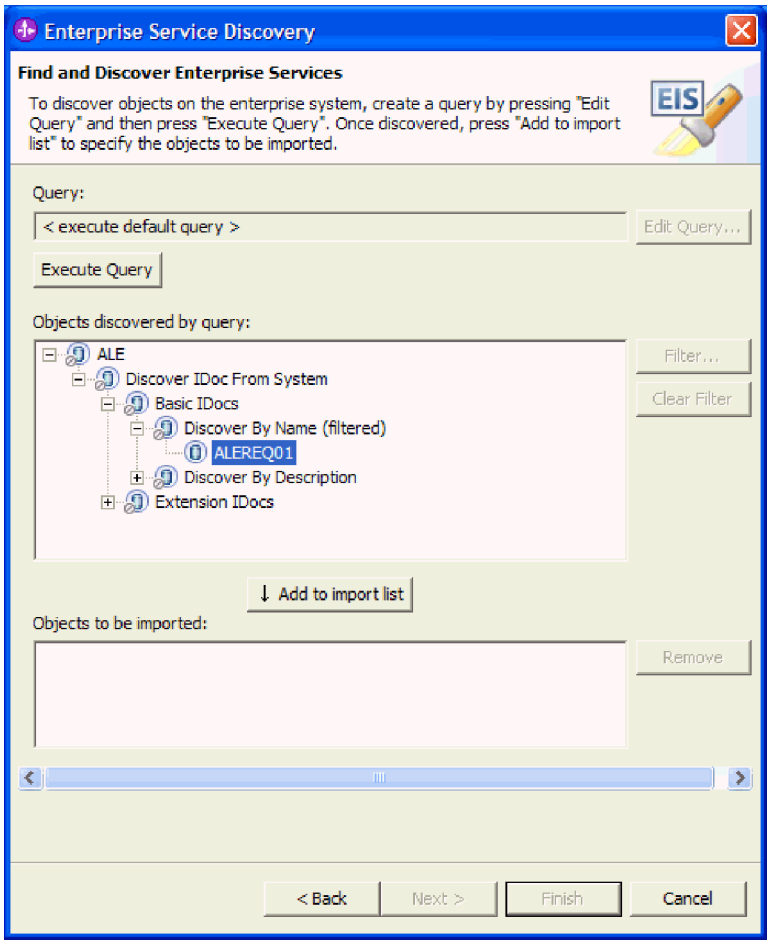


Figure 171. The Find and Discover Enterprise Services window

7. In the Configuration Parameters for ALEREQ01 window, accept the default values for ALEREQ01 by clicking **OK**.

ALEREQ01 is now displayed under **Objects to be imported**.

8. Click **Next**.

Result

The enterprise service discovery wizard has found ALEREQ01, and you are ready to configure it.

Configuring the selected objects

To configure the business object, you specify information about the object (such as the location of the object and the operation associated with the object).

1. In the **Object Location (Enter relative Path)** field, type `bodefs` as the name of the directory.

   Note: Inbound is the default setting for Service Type. Leave the setting as is.

2. For **Operations**, select **Create** and click **Add**.
3. Click **Next**.

Result

You have associated an operation (Create) with the object and selected a location for the object. The Generate Artifacts window is displayed.

Generating artifacts

To generate the module, which is the artifact that can be exported to an EAR file for deployment, you create the module, include the adapter in the module, and specify an alias used to authenticate the caller to the SAP server.

1. In the Generate Artifacts window, create a new module.
   a. Click **New**.
   b. Click **Create a module project** and click **Next**.
   c. Type `split`.

      As you type the name of the module, the name is added to the path next to **Directory**.

   d. Click **Finish**.
2. Accept the default value for **Name** and **Deploy connector with module**.
3. Select **Use discovered connection properties**.

   When you select **Use discovered connection properties**, the entries you made earlier (such as user name and IP address) are displayed at the bottom of the window.

4. Indicate the authentication alias to use by typing the alias that you created (in the beginning of the tutorial) in the administrative console. In the example shown earlier, the alias is `widNode/SAP_Auth_Alias`.
5. Use the information in the following table to set the required inbound connection properties. If a property is already filled in (for example, Client), leave the value that was generated in an earlier task. If you need information about the values to enter, see your SAP administrator.

Option	Description
Gateway Host	Specify the SAP gateway host where the gateway service is running.
Gateway Service	Specify the gateway server identifier. This value is often `sapgw00`.

Option	Description
RFC Program ID	Specify the program identifier under which the RFC server program registers.
Auto Create Event Table	Select this check box.
Event Recovery Table Name	Type the name you specified when you created the data source (`ALEEventRecoveryDS`).
Event Recovery DataSource (JNDI) Name	Type the name you specified when you created the data source (`jdbc/ALEEventRecovery`).
Username to connect to Event Datasource	Type the name you use to access the data source.
Password to connect to Event Datasource	Type the password you use to access the data source.

6. Click **Finish**.
7. Set the MsgType property for the Create operation in the SapAlereq01 business object.
   a. In the Business Integration perspective, expand **Data Types** (under the split module), and double-click **SapAlereq01** to open it.

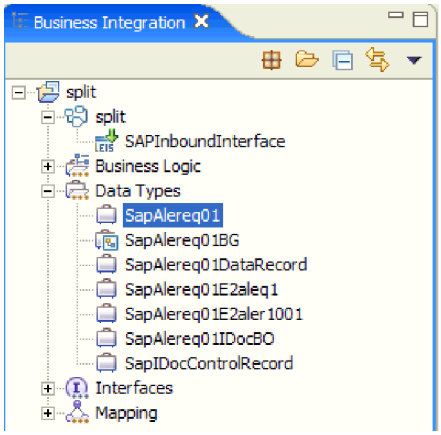


Figure 172. Data Types of the split module

   b. Click the SapAlereq01 business object, and in the Properties tab, click **Application Info**.

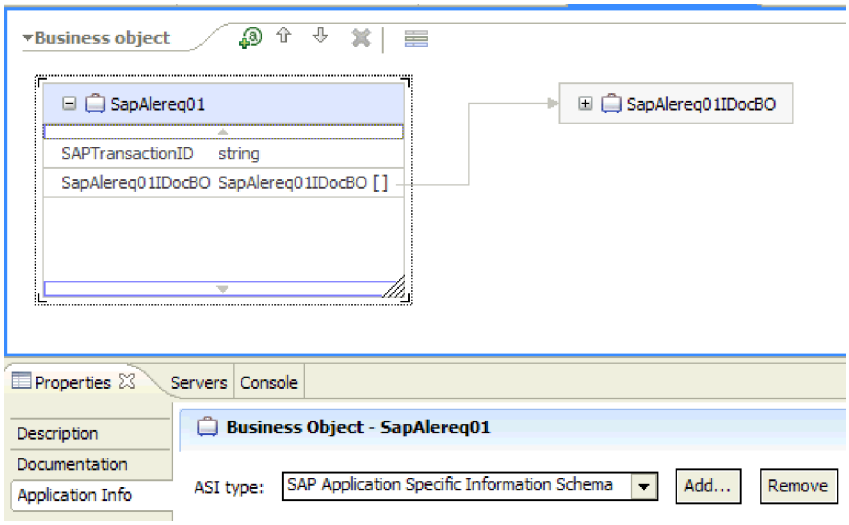


Figure 173. The Properties tab associated with SapAlereq01

c. Expand **sapasi:Operation**, and type `ALEREQ01` in the **sapasi:MsgType** field.

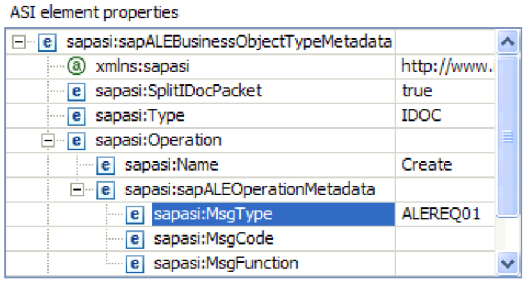


Figure 174. The ASI element properties

d. Close the SapAlereq01 tab.

e. In the Save Resource window, click **Yes**.

Result

The new split module is added to the Business Integration Perspective.

Generating reference bindings

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

How to perform this task

1. In the Business Integration Perspective of WebSphere Integration Developer, right-click the split module, and select **Open With → Assembly Editor**.
2. In the Assembly Diagram window, create a new component by clicking the top icon in the left pane and then clicking the top icon in the resulting menu, which has hover help that reads **Component (with no implementation type)**.

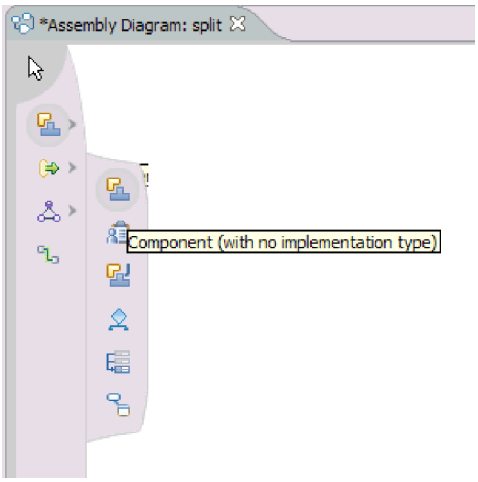


Figure 175. Selecting the new component icon

The cursor changes to the placement icon.

3. Click the palette to add the new component to the Assembly Diagram window.
4. Click and drag the Export component to the new component.

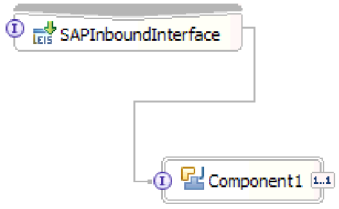


Figure 176. Wiring the components

5. In the Add Wire window, click **OK**.
6. Create a Java component to act as an endpoint by right-clicking the new component and selecting **Generate Implementation → Java**.
7. In the Generate Implementation window, select the package in which the Java code will be created, and click **OK**.
8. In the Java file editor, make any desired changes to the Java file. For example, you might wish to write code to print trace and log messages.
9. Save the Java file.

Result

You have generated a component that acts as an endpoint so that you can test your module.

Deploying the module for testing

To deploy the module to the test environment of WebSphere Process Server, you start the server and add the module (splitApp) to it. "App" is appended to the name of the module to indicate that the module is a deployable application.

1. Select the test environment server.

a. Click the **Servers** tab.
b. Right-click **WebSphere Process Server v6.0**.

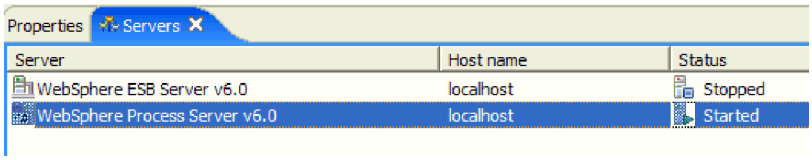


Figure 177. Selecting the WebSphere Process Server test environment from the Servers tab

2. Click **Add and remove projects**.
3. Select **splitApp** and click **Add**.
4. Click **Finish**.

Result

You see status messages in the **Console** tab as splitApp is deployed to the server.

Testing the module

Test the module using the WebSphere Integration Developer test client to make sure you can receive an IDoc from the SAP server.

About this task

To test this tutorial, you use actual values that exist on your SAP server. If you have not already done so, obtain actual values for the following data. If necessary, see your SAP administrator to obtain the data.

- RFC Program ID
- Client
- IdocNumber
- SenderPort
- PartnerNumberofSender
- ReceiverPort
- PartnerNumberofRecipient

How to perform this task

1. In the Business Integration perspective, begin the testing procedure by right-clicking **split** and clicking **Test** → **Attach**.
2. Examine the Configurations window and confirm that a monitor exists for the export.
3. Return to the Events window, and click **Continue**.
4. Select **WebSphere Process Server v6.0** and click **Finish**.
5. Enter data in the SAP server to trigger an inbound event.

   You should have the ALE Outbound configuration complete for the DEBMAS message type before proceeding with the event triggering using the following steps. Refer to SAP documentation for help with configuring ALE. (help.sap.com).

   a. Use the WE19 transaction in the SAP client user interface to send an ALE IDoc from the SAP instance.
   b. Click **Existing IDoc**.

c. Select an existing IDoc that you want to send out.
d. Select **IDocCreate** from the menu.
e. Click **Standard Outbound Processing**.
f. Click **Continue**.

   This creates an event for the ALE inbound application.

6. Check the values in the object published to the WebSphere Integration Developer test client.

Result

After the adapter has successfully processed the event, the Request parameters window is populated with the data object returned by the adapter.

Tutorial 5: Receiving a non-split IDoc packet

To create a module that receives a non-split IDoc packet, you create an adapter project, use the enterprise service discovery wizard to generate business objects based on the IDoc, and create a module that contains WebSphere Adapter for SAP Software and the newly generated business objects. You then deploy the module to the test environment of WebSphere Integration Developer.

Creating the authentication alias

To create an authentication alias, display the WebSphere Process Server administrative console and specify the user ID and password you use to access the SAP server. The user ID and password are then associated with the authentication alias.

1. Launch WebSphere Integration Developer by clicking **Start → Programs → IBM WebSphere → Integration Developer V6.0.2 → WebSphere Integration Developer V6.0.2**.
2. If you are prompted to specify a workspace, accept the default value.

   The workspace is a directory where WebSphere Integration Developer stores your project.
3. When the WebSphere Integration Developer window is displayed, close the Welcome page.
4. Switch to the Business Integration Perspective by clicking **Window → Open Perspective**. Then click **Business Integration (default)** and click **OK**.
5. Display the administrative console.
   a. Click the **Servers** tab.
   b. If **WebSphere Process Server v6.0** does not show a status of **Started**, right-click **WebSphere Process Server v6.0** and click **Start**.
   c. Right-click **WebSphere Process Server v6.0** and click **Run administrative console**.

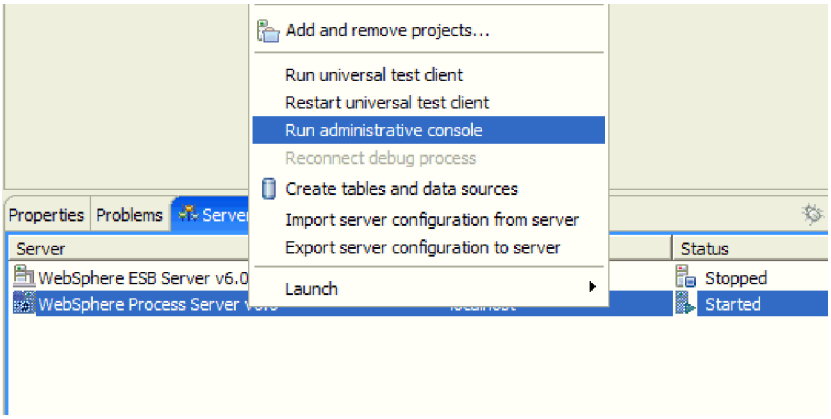


Figure 178. Selecting Run administrative console for the WebSphere Process Server

d. Log in to the administrative console by typing **admin** and clicking **Log in**.

6. In the WebSphere Process Server administrative console, click **Security** → **Global security**.

Figure 179. The Security item on the administrative console

7. Under **Authentication**, click **JAAS Configuration** → **J2C Authentication data**.

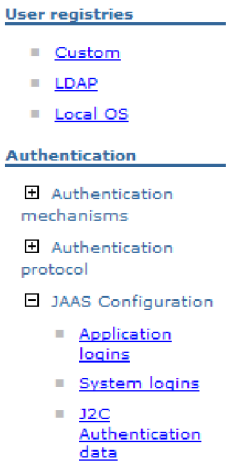


Figure 180. The Authentication section of the administrative console

8. If an alias named **SAP_Auth_Alias** does not already exist, create it now.
   a. Determine from your SAP administrator whether the authentication alias is case-sensitive (for example, whether the alias must be entered in uppercase).
   b. Click **New**.
   c. In the General properties window, type `SAP_Auth_Alias` in the **Alias** field.

      Note: If your SAP server requires that the alias be entered in a specific format (for example, all uppercase), type the alias according to that format.
   d. Type the user ID and password that are required to connect to the SAP server.

      Note: If your SAP server requires that the password be entered in a specific format (for example, all uppercase), type the password according to that format.
   e. Click **OK**.

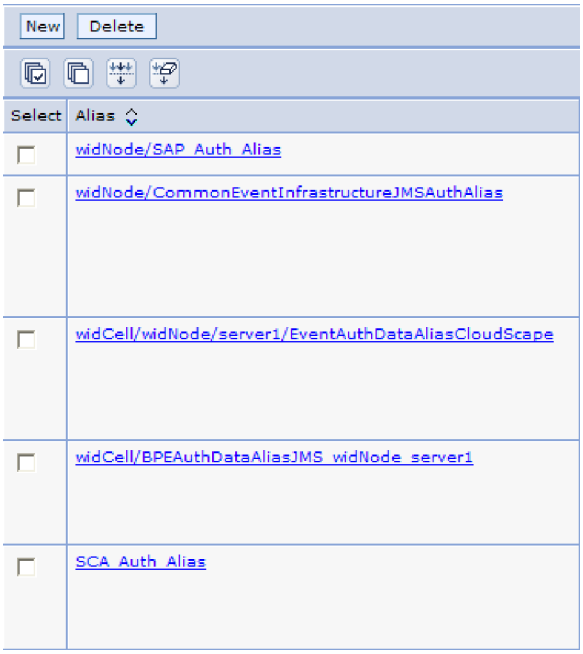


Figure 181. The list of aliases, including the newly created SAP_Auth_Alias

Make note of the name as it appears in the Alias list. In the example, the name is **widNode/SAP_Auth_Alias**. This name is the one you will use in subsequent configuration windows.

f. Click **Save**.

Result

You have created an authentication alias, which you will use when you configure the adapter properties.

Creating the adapter project in WebSphere Integration Developer

To begin the process of creating a module to communicate with an SAP service, you create an adapter project. The adapter project (called a *connector project* in WebSphere Integration Developer) 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.

About this task

You can use the same adapter project for multiple tutorials. If you have already created an adapter project by importing the adapter RAR file, you do not need to create it again, unless you want to have separate adapter projects for each tutorial.

How to perform this task

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**.

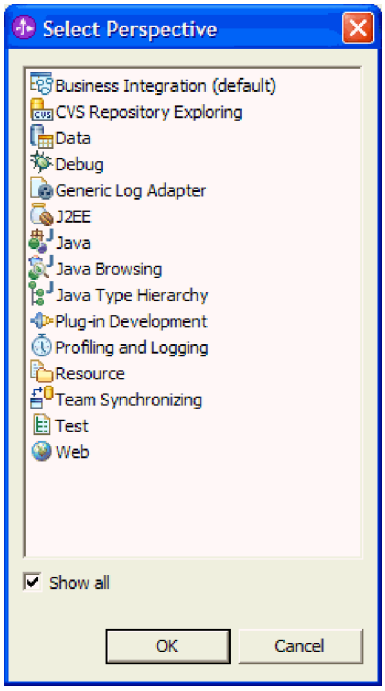


Figure 182. Selecting J2EE from the Select Perspective list

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

d. Click **OK**.

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

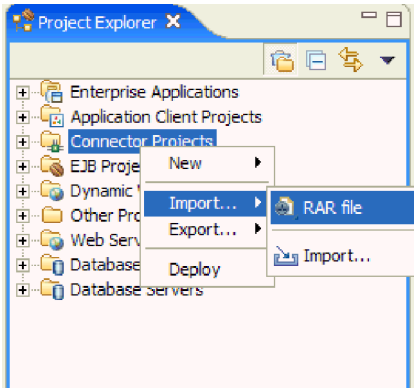


Figure 183. Importing the RAR file

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

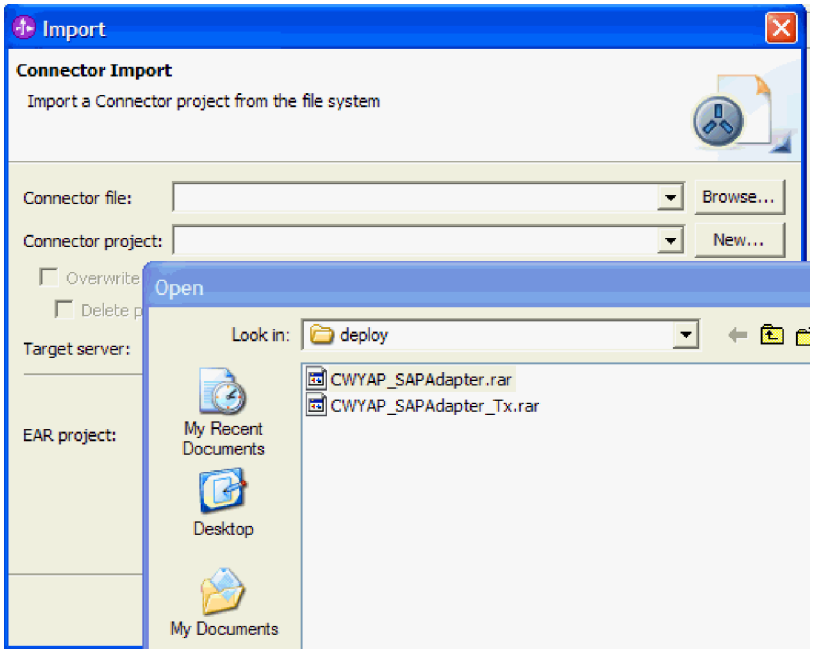


Figure 184. Selecting the RAR file from the installation directory

4. Select the RAR file and click **Open**.
5. Accept the default setting (**CWYAP_SAPAdapter**) for **Connector project**.

   The connector project has the same name as the RAR file.

   If a project named CWYAP_SAPAdapter.rar already exists in this workspace, the name in the **Connector project** field has a number appended to it (for example, CWYAP_SAPAdapter1).
6. 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.
7. Clear the **Add module to an EAR project** check box.

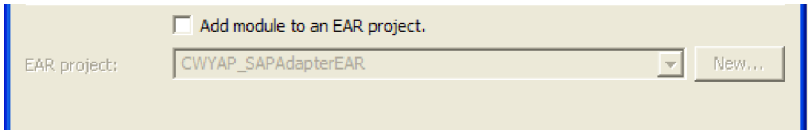


Figure 185. Clearing the Add module to an EAR project check box

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

Result

A new adapter project, named CWYAP_SAPAdapter, is created. To see its contents, expand **CWYAP_SAPAdapter**.

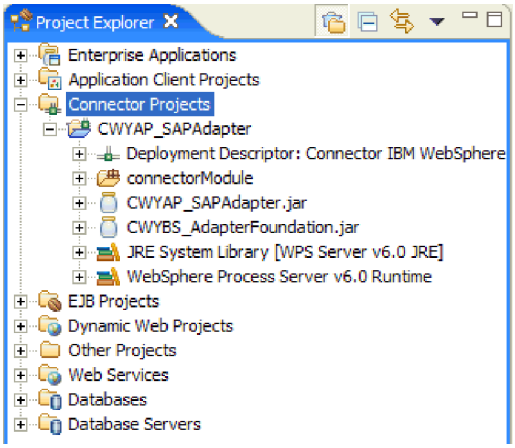


Figure 186. The CWYAP_SAPAdapter project in the Project Explorer window

Adding external dependencies

To add the required external dependency files, you copy the files, including the sapjco.jar file, to directories within the WebSphere Integration Developer directory. You then add the sapjco.jar file to the adapter project you created.

1. If you have not already done so as part of the installation of the adapter or as part of running another tutorial, copy the required files as outlined in the following steps.
   a. Obtain the files for your operating system from your SAP administrator or from the SAP Web site.

Table 23. Files to be installed

Operating system	Files to be installed
Windows	Any *.dll files that come with the SAP Jco download from the SAP Web site
Unix (including Unix System Services on z/OS)	Any .so and .o files that come with the SAP Jco download from the SAP Web site

   b. Copy the files to the following locations in the WebSphere Integration Developer installation directory:
      - \runtimes\bi_v6\java\bin
      - \eclipse\jre\bin

      For z/OS, add the files to the ${*WAS_INSTALL_ROOT*}/lib directory.
   c. For Windows environments only, obtain the msvcp71.dll and msvcr71.dll files from your SAP administrator or the SAP Web site.
   d. For Windows environments only, install the msvcp71.dll and msvcr71.dll files in the Windows system path.
   e. Obtain the sapjco.jar file from your SAP administrator or the SAP Web site.
   f. Copy sapjco.jar to the following location in the WebSphere Integration Developer installation directory: \runtimes\bi_v6\lib

      For z/OS, add ${*WAS_INSTALL_ROOT*}/lib/sapjco.jar to WAS_SERVER_ONLY_server_region_classpath
2. Import the sapjco.jar file into the adapter project.

a. In the J2EE perspective of WebSphere Integration Developer, expand **Connector Projects**.
b. Right-click **CWYAP_SAPAdapter** and click **Properties**.

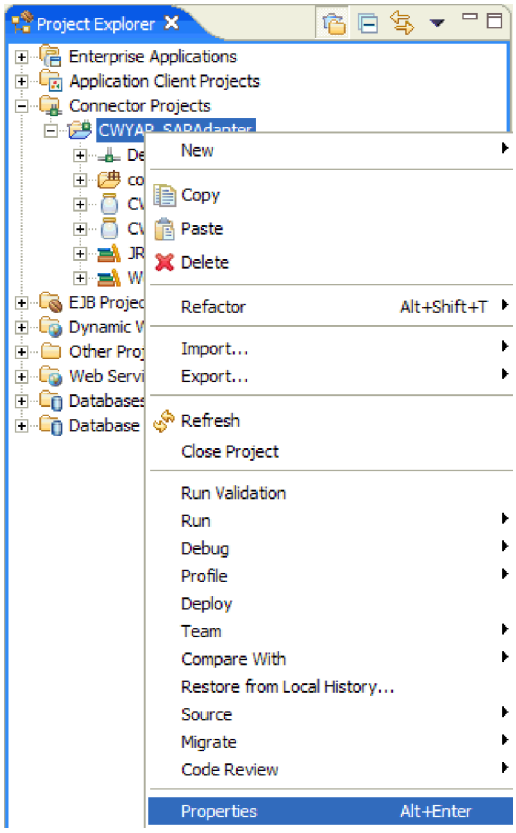


Figure 187. The CWYAP_SAPAdapter project, displayed in the Project Explorer

c. In the left side of the Properties for CWYAP_SAPAdapter window, click **Java Build Path**.

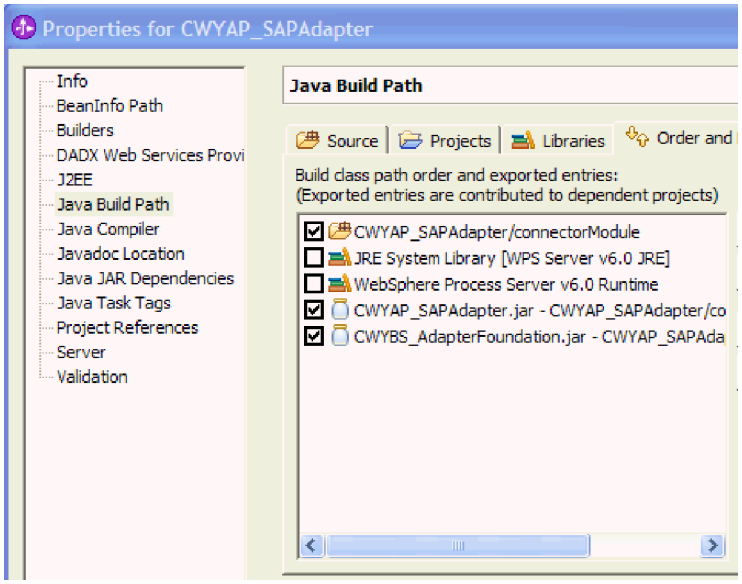


Figure 188. Selecting Java Build Path

d. Click the **Libraries** tab, and click **Add External JARs**.

e. Navigate to the directory on your local file system where the sapjco.jar file is located. Then select **sapjco.jar** and click **Open**.

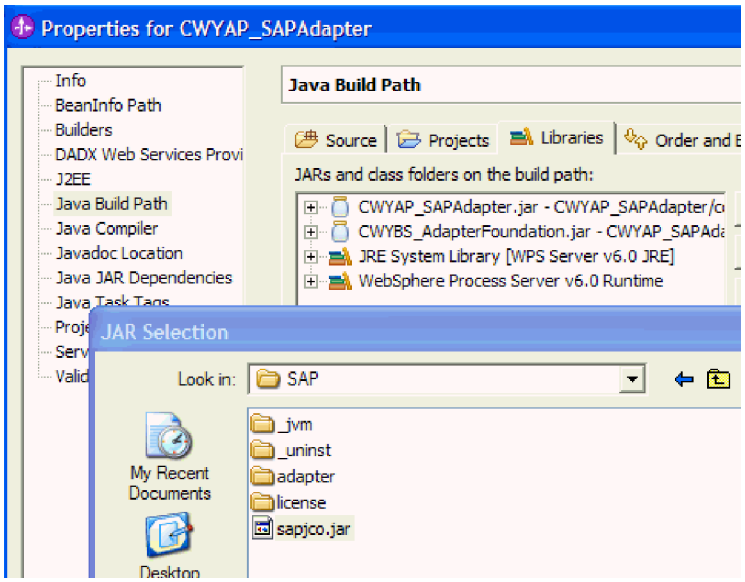


Figure 189. The JAR Selection window, with the sapjco.jar file highlighted for selection

f. Click **OK**.

The file sapjco.jar appears in the list of JARs and class folders in the build path.

Result

The sapjco.jar file is now part of your connector project and appears in the Project Explorer window of WebSphere Integration Developer.

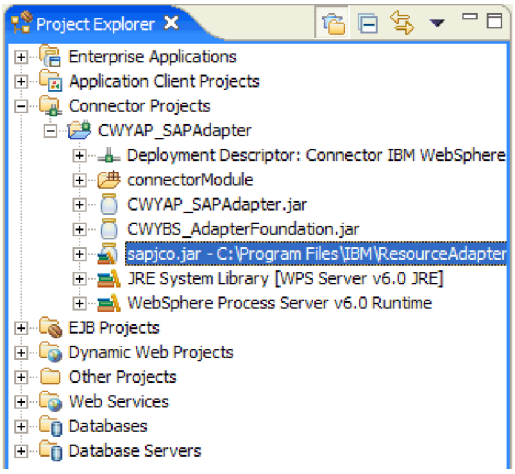


Figure 190. The Project Explorer window of WebSphere Integration Developer

Configuring the data source

To configure the JDBC data source for ALE inbound processing, you use the administrative console. The data source, which is required for inbound processing, is used for event tracking and recovery. This tutorial uses the Cloudscape JDBC provider.

1. Begin the process of creating a data source by displaying the administrative console.
   a. From the J2EE Perspective of WebSphere Integration Developer, click the **Server** tab.
   b. If **WebSphere Process Server v6.0** does not show a status of **Started**, right-click **WebSphere Process Server v6.0** and click **Start**.
   c. Right-click **WebSphere Process Server v6.0** and click **Run administrative console**.

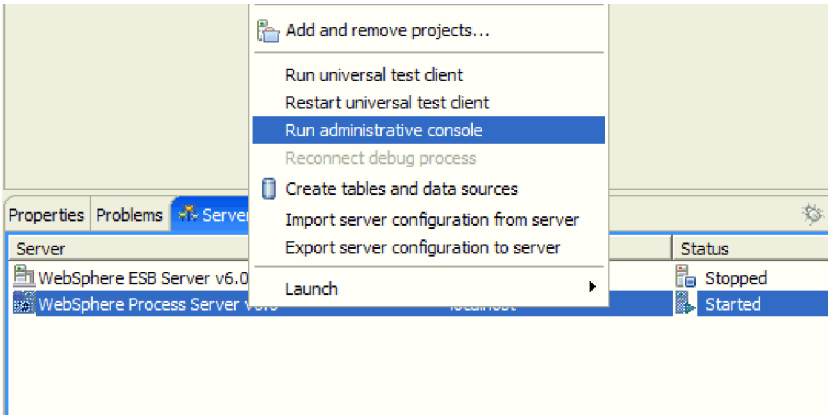


Figure 191. Selecting Run administrative console for the WebSphere Process Server

   d. Log in to the administrative console by typing **admin** and clicking **Log in**.
2. Select a JDBC provider.
   a. Click **Resources** → **JDBC Providers**.
   b. Click **Cloudscape JDBC Provider**.

3. Select **Data sources**.

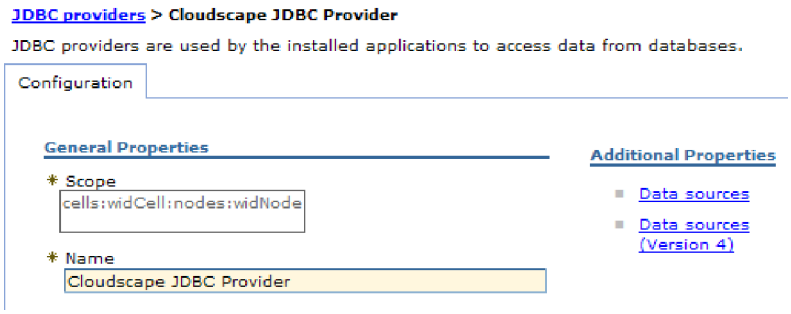


Figure 192. The Cloudscape JDBC provider Configuration tab

4. Create a new data source by clicking **New**.
5. Type the following values in the specified field. Leave the default values for the rest of the configurations.

Option	Description
Name	ALEEventRecoveryDS
JNDI Name	jdbc/ALEEventRecovery
Description	ALEEventRecoveryDS
DatabaseName	ALEEventRecoveryDB

6. Click **Apply**.
   After the changes are applied, **Custom properties** becomes active.
7. Click **Custom properties**.
8. Scroll down and click **createDatabase**.

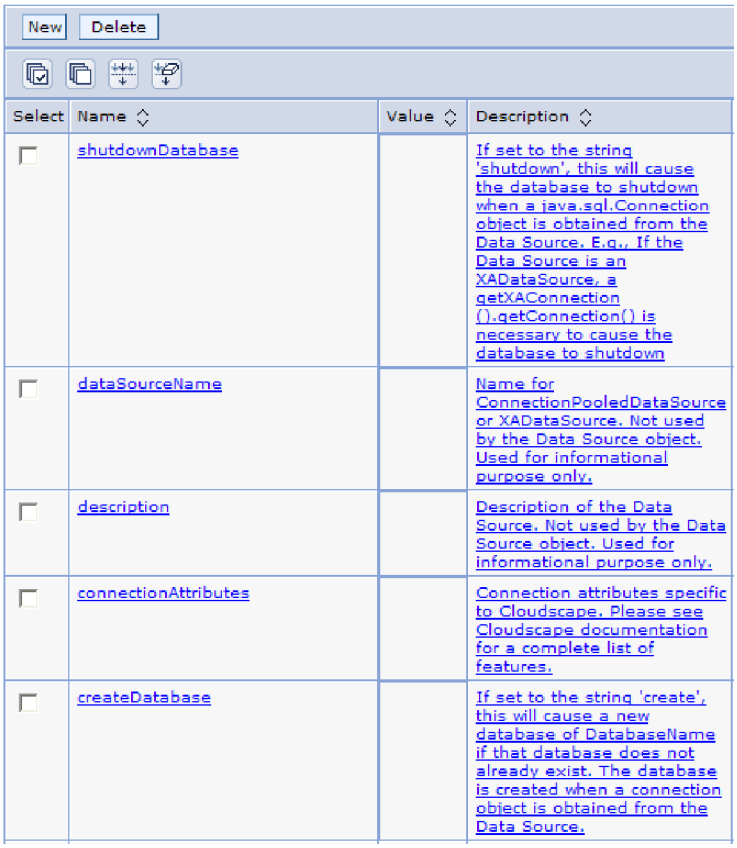


Figure 193. Selecting the createDatabase entry

9. Type `create` in the **value** field, and click **Apply**.
10. Save your configurations.

Result

The new data source, ALEEventRecoveryDS, is displayed in the list of data sources.

Configuring the adapter for inbound processing

To configure the adapter, set the connection properties for enterprise service discovery. Then use the enterprise service discovery wizard to select and configure the necessary business objects and to generate a deployable module.

Setting connection properties for enterprise service discovery

To set connection properties for the enterprise service discovery wizard so that it can access the SAP server, specify such information as the user name and password you use to access the server as well as the name or IP address of the server.

1. In WebSphere Integration Developer, start the enterprise service discovery wizard.
   a. Switch to the Business Integration Perspective by clicking **Window** → **Open Perspective** → **Other**. Then click **Business Integration (default)** and click **OK**.
   b. Right-click **New** → **Enterprise Service Discovery**.

If **Enterprise Service Discovery** is not displayed, click **New** → **Other**, expand **Business Integration**, and click **Enterprise Service Discovery**. Then click **Next**.

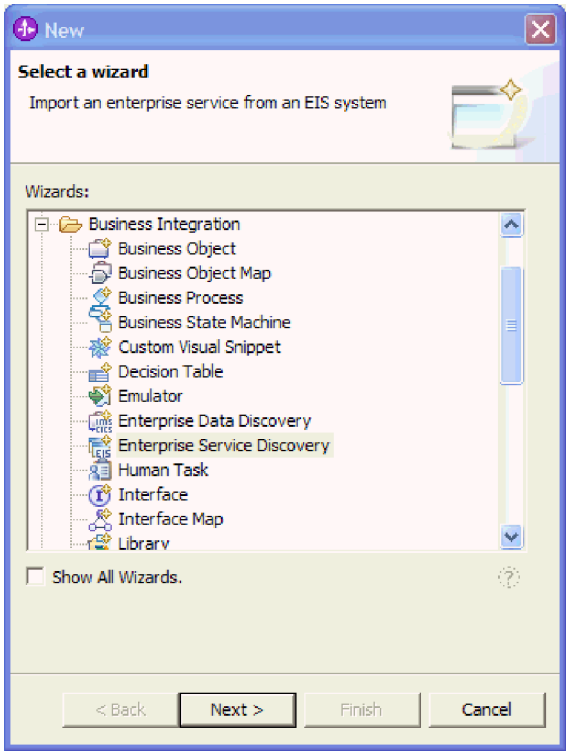


Figure 194. The expanded list of wizards

2. In the Select an Enterprise Service Resource Adapter window, make sure **IBM WebSphere Adapter for SAP Software** is selected, and click **Next**.
3. In the Configure Settings for Discovery Agent window, specify the configuration properties needed to connect to the SAP server.
   a. Type the name and password you use to access the SAP server.

      The password is case-sensitive.
   b. Type your client ID.

      This is typically `100`.
   c. In the **Application Server Host** field, type the name (or IP address) of your SAP server.

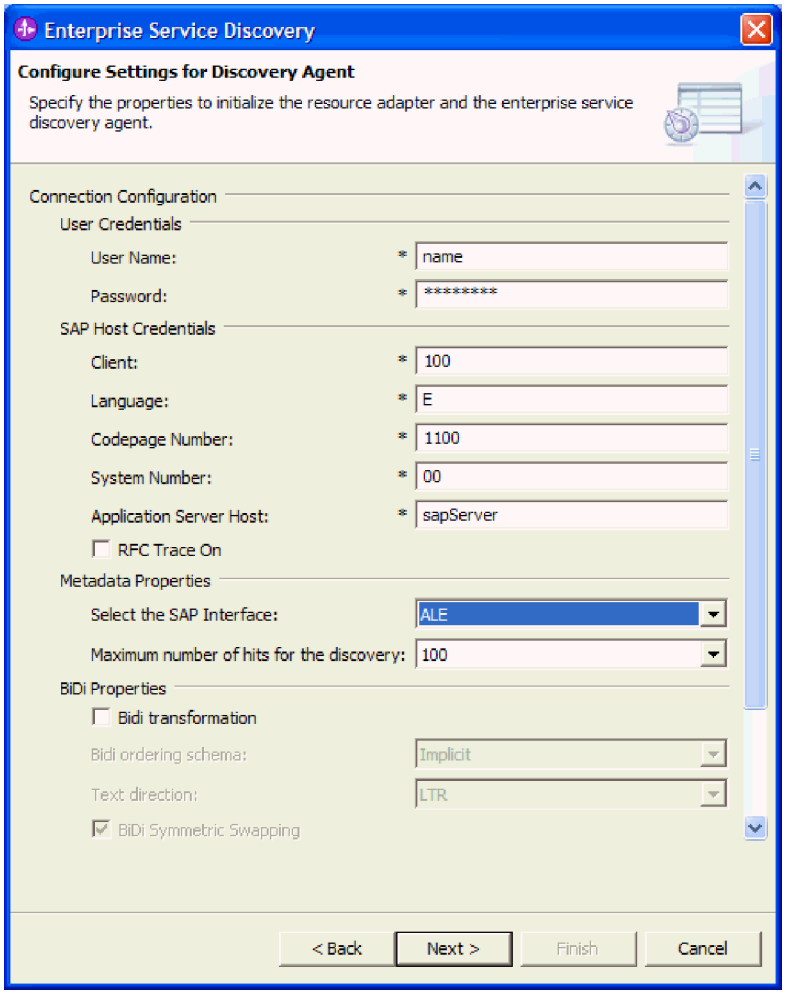


Figure 195. The Configure Settings for Discovery Agent window

4. Make sure **ALE** (the default) is displayed in the **Select the SAP interface** field.
5. Set the logging level so that you can see any errors that might arise during enterprise service discovery.
   a. At the bottom of the Configure Settings for Discovery Agent window, click **Show Advanced**.

      The button changes to **Hide Advanced**.

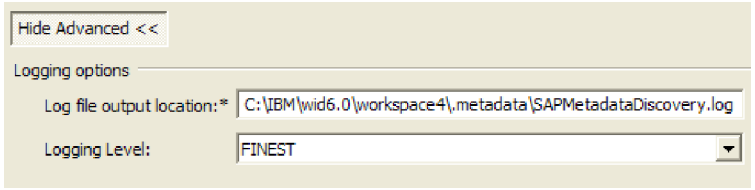


Figure 196. The Logging options displayed when you select Show Advanced

   b. For **Logging Level**, select **FINEST**.
6. Click **Next**.

Result

The enterprise service discovery wizard contacts the SAP server, using the information you provided (such as user name and password) to log in. You see the Find and Discover Enterprise Services window.

Selecting business objects and services

To select the ALE IDoc, specify search criteria (such as the name of the IDoc). The enterprise service discovery wizard uses the search criteria to find the IDoc on the SAP server.

1. From the Find and Discover Enterprise Services window, begin the process of searching for services by clicking **Execute Query**.
2. Expand **ALE**, **Discover IDoc From System**, and **Basic IDocs**, and click **Discover By Name**.

   The **Filter** button is now enabled.

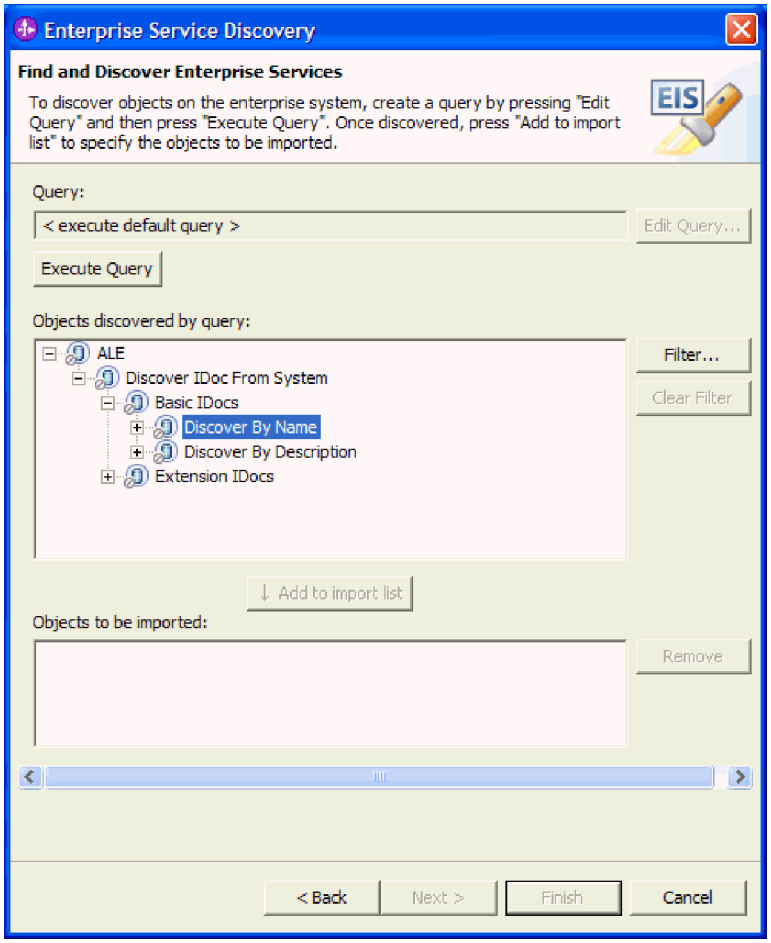


Figure 197. The Find and Discover Enterprise Services window

3. Click **Filter**.
4. In the Filter Properties for Discover By Name window, indicate that you want the ALEREQ1 IDoc by typing `alereq01`.

Figure 198. The Filter Properties for Discover By Name window

5. Click **OK**.
6. Select the IDoc.
   a. Expand **Discover By Name (filtered)**.
   b. Click **ALEREQ01** and click **Add to import list**.

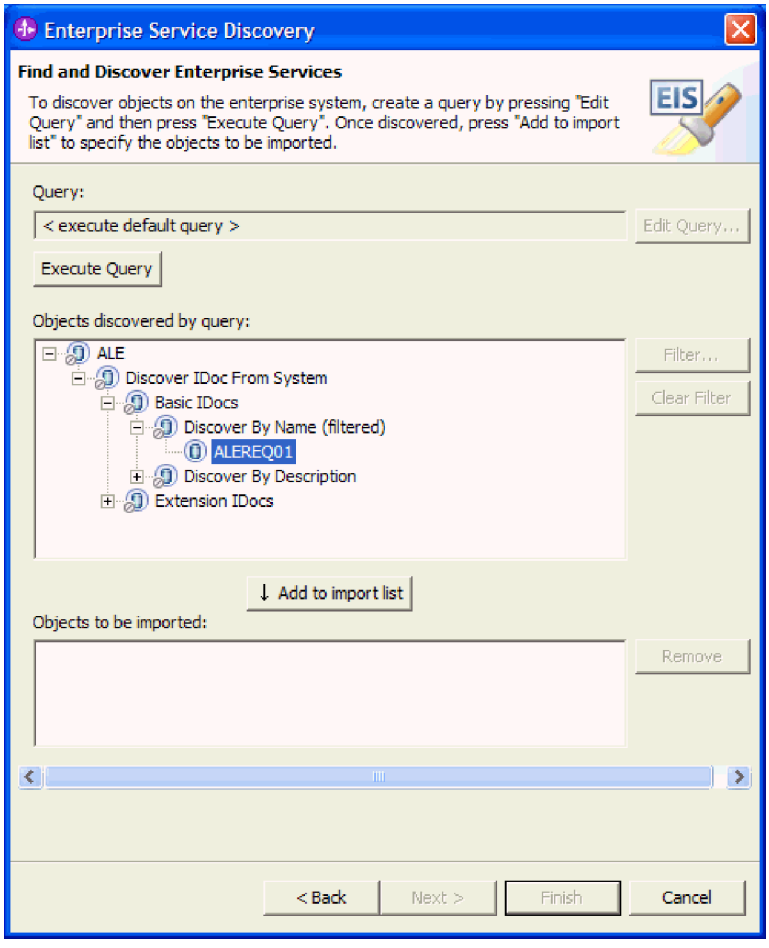


Figure 199. The Find and Discover Enterprise Services window

7. In the Configuration Parameters for ALEREQ01 window, select the **Send an IDoc Packet as one Business Object** and **Send IDoc with Unparsed Data** check boxes, and click **OK**.

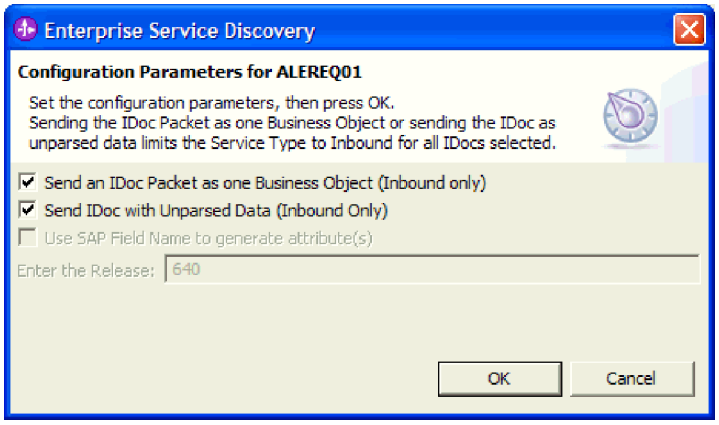


Figure 200. The Configuration Parameters window

ALEREQ01 is now displayed under **Objects to be imported**.

8. Click **Next**.

Result

The enterprise service discovery wizard has found ALEREQ01, and you are ready to configure it.

Configuring the selected objects

To configure the business object, you specify information about the object (such as the location of the object and the operation associated with the object).

1. In the **Object Location (Enter relative Path)** field, type `bodefs` as the name of the directory.

   Note: Inbound is the default setting for Service Type. Leave the setting as is.
2. For **Operations**, select **Create** and click **Add**.
3. Click **Next**.

Result

You have associated an operation (Create) with the object and selected a location for the object. The Generate Artifacts window is displayed.

Generating artifacts

To generate the module, which is the artifact that can be exported to an EAR file for deployment, you create the module, include the adapter in the module, and specify an alias used to authenticate the caller to the SAP server.

1. In the Generate Artifacts window, create a new module.
   a. Click **New**.
   b. Click **Create a module project** and click **Next**.
   c. Type `ALENonSplit`.

      As you type the name of the module, the name is added to the path next to **Directory**.

d. Click **Finish**.

2. Accept the default value for **Name** and **Deploy connector with module**.
3. Select **Use discovered connection properties**.

   When you select **Use discovered connection properties**, the entries you made earlier (such as user name and IP address) are displayed at the bottom of the window.
4. Indicate the authentication alias to use by typing the alias that you created (in the beginning of the tutorial) in the administrative console. In the example shown earlier, the alias is `widNode/SAP_Auth_Alias`.
5. Use the information in the following table to set the required inbound connection properties. If a property is already filled in (for example, Client), leave the value that was generated in an earlier task. If you need information about the values to enter, see your SAP administrator.

Option	Description
Gateway Host	Specify the SAP gateway host where the gateway service is running.
Gateway Service	Specify the gateway server identifier. This value is often `sapgw00`.
RFC Program ID	Specify the program identifier under which the RFC server program registers.
Auto Create Event Table	Select this check box.
Event Recovery Table Name	Type the name you specified when you created the data source (`ALEEventRecoveryDS`).
Event Recovery DataSource (JNDI) Name	Type the name you specified when you created the data source (`jdbc/ALEEventRecovery`).
Username to connect to Event Datasource	Type the name you use to access the data source.
Password to connect to Event Datasource	Type the password you use to access the data source.

6. Click **Finish**.

Result

The new ALENonSplit module is added to the Business Integration perspective.

Generating reference bindings

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

How to perform this task

1. In the Business Integration Perspective of WebSphere Integration Developer, right-click the split module, and select **Open With → Assembly Editor**.

2. In the Assembly Diagram window, create a new component by clicking the top icon in the left pane and then clicking the top icon in the resulting menu, which has hover help that reads **Component (with no implementation type)**.

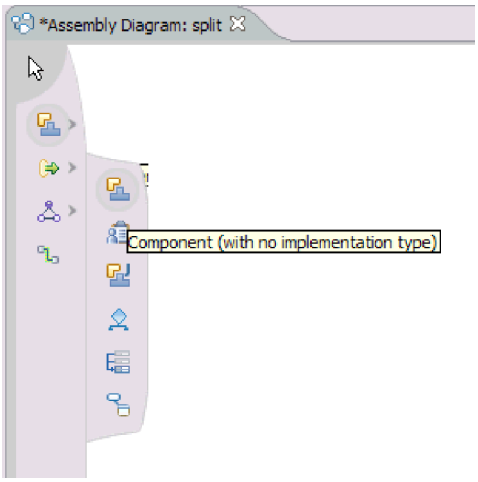


Figure 201. Selecting the new component icon

The cursor changes to the placement icon.

3. Click the palette to add the new component to the Assembly Diagram window.
4. Click and drag the Export component to the new component.

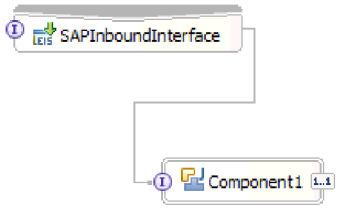


Figure 202. Wiring the components

5. In the Add Wire window, click **OK**.
6. Create a Java component to act as an endpoint by right-clicking the new component and selecting **Generate Implementation → Java**.
7. In the Generate Implementation window, select the package in which the Java code will be created, and click **OK**.
8. In the Java file editor, make any desired changes to the Java file. For example, you might wish to write code to print trace and log messages.
9. Save the Java file.

Result

You have generated a component that acts as an endpoint so that you can test your module.

Deploying the module for testing

To deploy the module to the test environment of WebSphere Process Server, you start the server and add the module (ALENonSplitApp) to it. "App" is appended to the name of the module to indicate that the module is a deployable application.

1. Select the test environment server.
   a. Click the **Servers** tab.
   b. Right-click **WebSphere Process Server v6.0**.

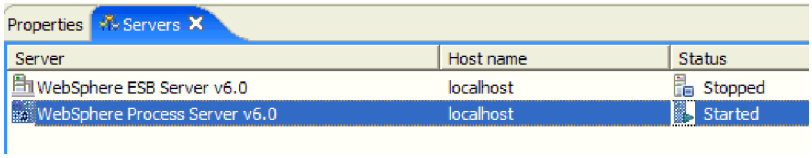


Figure 203. Selecting the WebSphere Process Server test environment from the Servers tab

2. Click **Add and remove projects**.
3. Select **ALENonSplitApp** and click **Add**.
4. Click **Finish**.

Result

You see status messages in the **Console** tab as ALENonSplitApp is deployed to the server.

Testing the module

Test the module using the WebSphere Integration Developer test client to make sure you can receive an IDoc from the SAP server.

1. In the Business Integration perspective, begin the testing procedure by right-clicking **ALENonSplit** and clicking **Test → Attach**.
2. Examine the Configurations window and confirm that a monitor exists for the export.
3. Return to the Events window, and click **Continue**.
4. Select **WebSphere Process Server v6.0** and click **Finish**.
5. Enter data in the SAP server to trigger an inbound event.

   You should have the ALE Outbound configuration complete for the DEBMAS message type before proceeding with the event triggering using the following steps. Refer to SAP documentation for help with configuring ALE. (help.sap.com).
   a. Use the WE19 transaction in the SAP client user interface to send an ALE IDoc from the SAP instance.
   b. Click **Existing IDoc**.
   c. Select an existing IDoc that you want to send out.
   d. Select **IDocCreate** from the menu.
   e. Click **Standard Outbound Processing**.
   f. Click **Continue**.

   This creates an event for the ALE inbound application.
6. Check the values in the object published to the WebSphere Integration Developer integration test client.

Result

After the adapter has successfully processed the event, the Request parameters window is populated with the data object returned by the adapter.

Tutorial 6: Querying data in an SAP table

To create a module that queries data in tables on the SAP server, you create an adapter project, use the enterprise service discovery wizard to generate business objects based on the data, and create a module that contains WebSphere Adapter for SAP Software and the newly generated business objects. You then deploy the module to the test environment of WebSphere Integration Developer.

Creating the authentication alias

To create an authentication alias, display the WebSphere Process Server administrative console and specify the user ID and password you use to access the SAP server. The user ID and password are then associated with the authentication alias.

1. Launch WebSphere Integration Developer by clicking **Start** → **Programs** → **IBM WebSphere** → **Integration Developer V6.0.2** → **WebSphere Integration Developer V6.0.2**.
2. If you are prompted to specify a workspace, accept the default value.

   The workspace is a directory where WebSphere Integration Developer stores your project.
3. When the WebSphere Integration Developer window is displayed, close the Welcome page.
4. Switch to the Business Integration Perspective by clicking **Window** → **Open Perspective**. Then click **Business Integration (default)** and click **OK**.
5. Display the administrative console.
   a. Click the **Servers** tab.
   b. If **WebSphere Process Server v6.0** does not show a status of **Started**, right-click **WebSphere Process Server v6.0** and click **Start**.
   c. Right-click **WebSphere Process Server v6.0** and click **Run administrative console**.

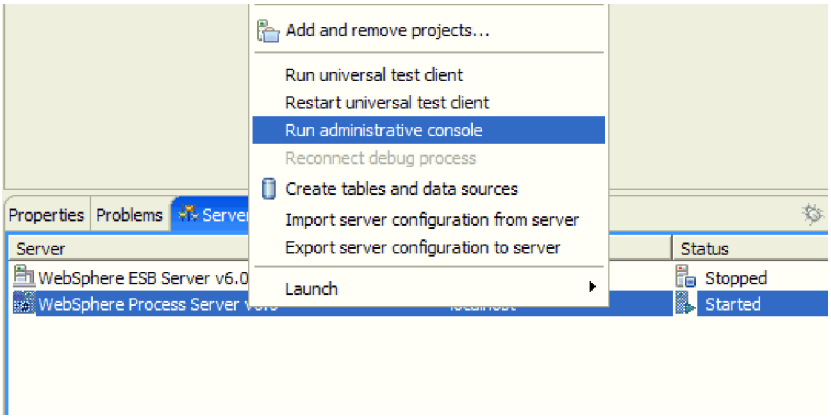


Figure 204. Selecting Run administrative console for the WebSphere Process Server

   d. Log in to the administrative console by typing **admin** and clicking **Log in**.

6. In the WebSphere Process Server administrative console, click **Security** → **Global security**.

Figure 205. The Security item on the administrative console

7. Under **Authentication**, click **JAAS Configuration** → **J2C Authentication data**.

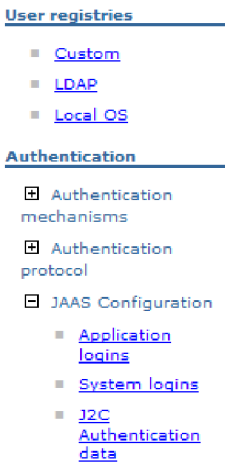


Figure 206. The Authentication section of the administrative console

8. If an alias named **SAP_Auth_Alias** does not already exist, create it now.
   a. Determine from your SAP administrator whether the authentication alias is case-sensitive (for example, whether the alias must be entered in uppercase).
   b. Click **New**.
   c. In the General properties window, type `SAP_Auth_Alias` in the **Alias** field.

Note: If your SAP server requires that the alias be entered in a specific format (for example, all uppercase), type the alias according to that format.

d. Type the user ID and password that are required to connect to the SAP server.

Note: If your SAP server requires that the password be entered in a specific format (for example, all uppercase), type the password according to that format.

e. Click **OK**.

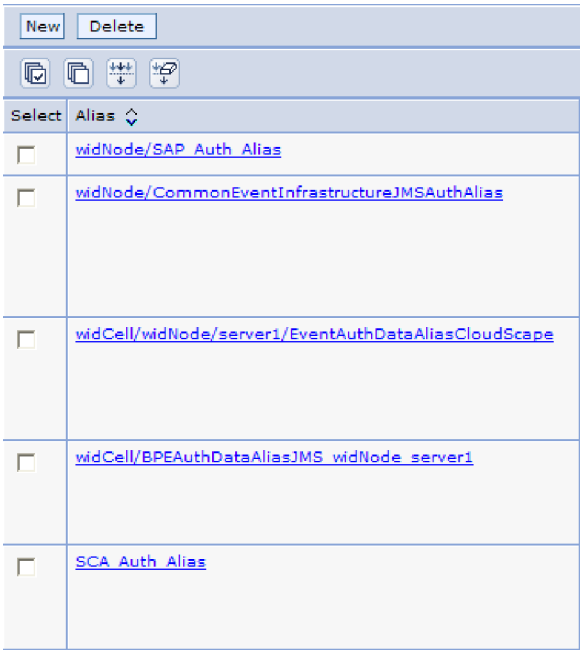


Figure 207. The list of aliases, including the newly created SAP_Auth_Alias

Make note of the name as it appears in the Alias list. In the example, the name is **widNode/SAP_Auth_Alias**. This name is the one you will use in subsequent configuration windows.

f. Click **Save**.

Result

You have created an authentication alias, which you will use when you configure the adapter properties.

Creating the adapter project in WebSphere Integration Developer

To begin the process of creating a module to communicate with an SAP service, you create an adapter project. The adapter project (called a *connector project* in WebSphere Integration Developer) 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.

About this task

You can use the same adapter project for multiple tutorials. If you have already created an adapter project by importing the adapter RAR file, you do not need to create it again, unless you want to have separate adapter projects for each tutorial.

How to perform this task

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**.

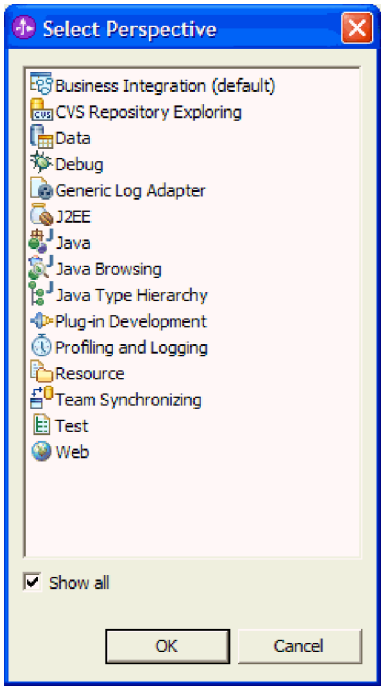


Figure 208. Selecting J2EE from the Select Perspective list

   c. If you see the Confirm Enablement window, select **Always enable capabilities and don't ask me again**.
   d. Click **OK**.
2. Import the RAR file by right-clicking **Connector Projects** and clicking **Import** → **RAR file**.

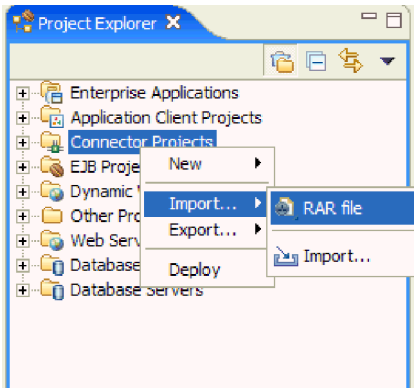


Figure 209. Importing the RAR file

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

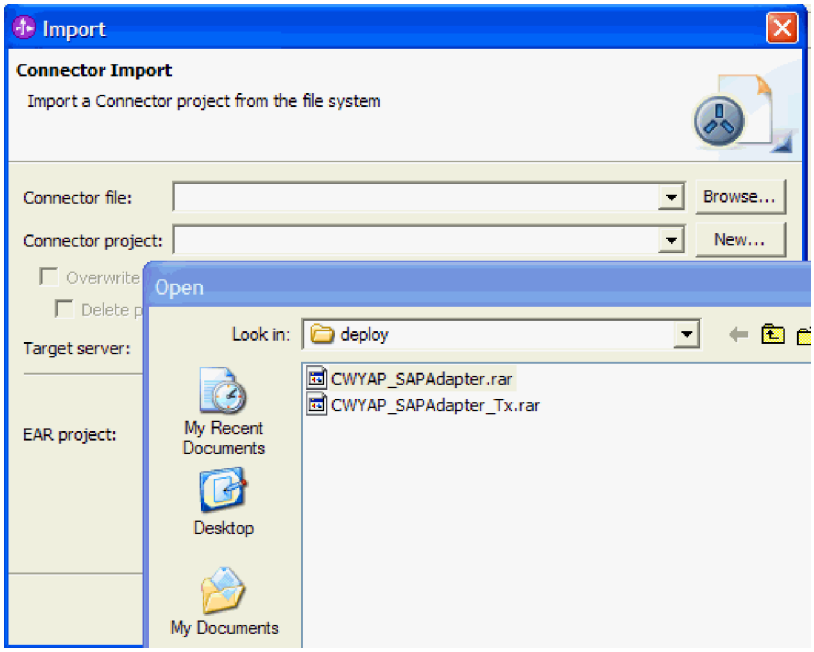


Figure 210. Selecting the RAR file from the installation directory

4. Select the RAR file and click **Open**.
5. Accept the default setting (**CWYAP_SAPAdapter**) for **Connector project**.

   The connector project has the same name as the RAR file.

   If a project named CWYAP_SAPAdapter.rar already exists in this workspace, the name in the **Connector project** field has a number appended to it (for example, CWYAP_SAPAdapter1).
6. 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.
7. Clear the **Add module to an EAR project** check box.

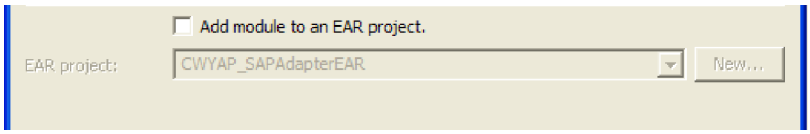


Figure 211. Clearing the Add module to an EAR project check box

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

8. Click **Finish**.

Result

A new adapter project, named CWYAP_SAPAdapter, is created. To see its contents, expand **CWYAP_SAPAdapter**.

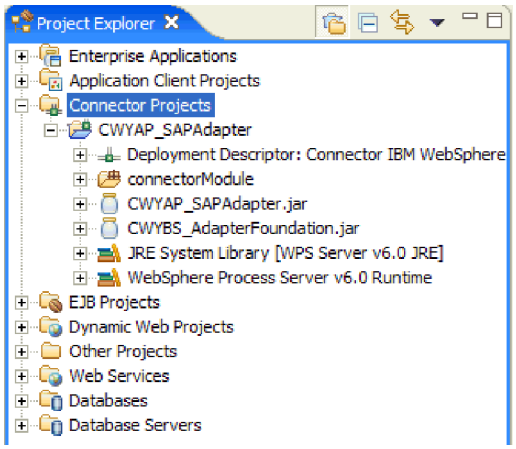


Figure 212. The CWYAP_SAPAdapter project in the Project Explorer window

Adding external dependencies

To add the required external dependency files, you copy the files, including the sapjco.jar file, to directories within the WebSphere Integration Developer directory. You then add the sapjco.jar file to the adapter project you created.

1. If you have not already done so as part of the installation of the adapter or as part of running another tutorial, copy the required files as outlined in the following steps.
   a. Obtain the files for your operating system from your SAP administrator or from the SAP Web site.

Table 24. Files to be installed

Operating system	Files to be installed
Windows	Any *.dll files that come with the SAP Jco download from the SAP Web site
Unix (including Unix System Services on z/OS)	Any .so and .o files that come with the SAP Jco download from the SAP Web site

   b. Copy the files to the following locations in the WebSphere Integration Developer installation directory:
      - \runtimes\bi_v6\java\bin

- \eclipse\jre\bin

  For z/OS, add the files to the ${*WAS_INSTALL_ROOT*}/lib directory.

c. For Windows environments only, obtain the msvcp71.dll and msvcr71.dll files from your SAP administrator or the SAP Web site.

d. For Windows environments only, install the msvcp71.dll and msvcr71.dll files in the Windows system path.

e. Obtain the sapjco.jar file from your SAP administrator or the SAP Web site.

f. Copy sapjco.jar to the following location in the WebSphere Integration Developer installation directory: \runtimes\bi_v6\lib

   For z/OS, add ${*WAS_INSTALL_ROOT*}/lib/sapjco.jar to WAS_SERVER_ONLY_server_region_classpath

2. Import the sapjco.jar file into the adapter project.

   a. In the J2EE perspective of WebSphere Integration Developer, expand **Connector Projects**.

   b. Right-click **CWYAP_SAPAdapter** and click **Properties**.

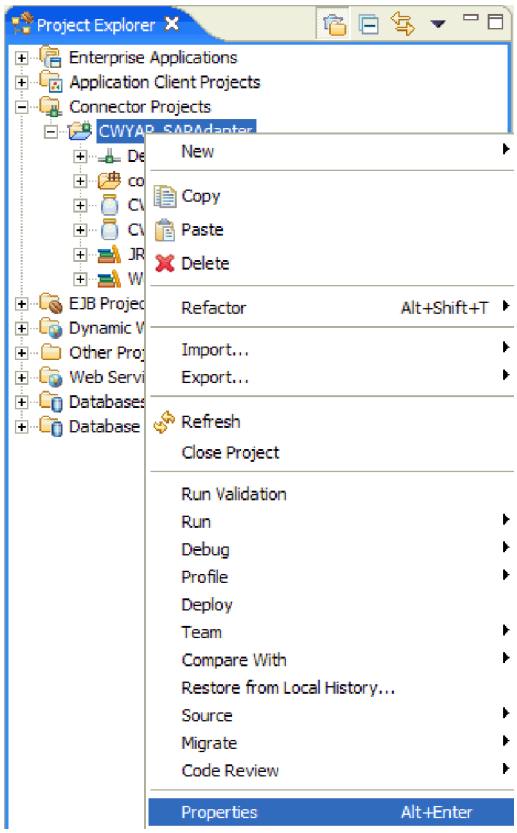


Figure 213. The CWYAP_SAPAdapter project, displayed in the Project Explorer

   c. In the left side of the Properties for CWYAP_SAPAdapter window, click **Java Build Path**.

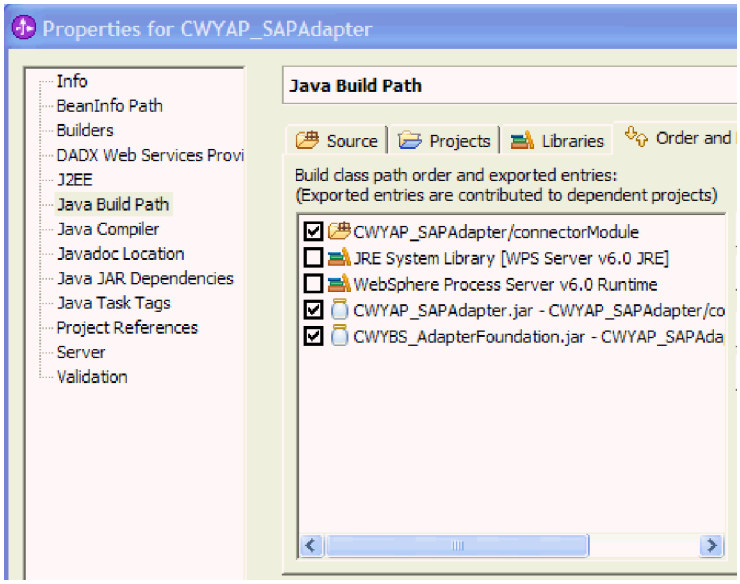


Figure 214. Selecting Java Build Path

d. Click the **Libraries** tab, and click **Add External JARs**.

e. Navigate to the directory on your local file system where the sapjco.jar file is located. Then select **sapjco.jar** and click **Open**.

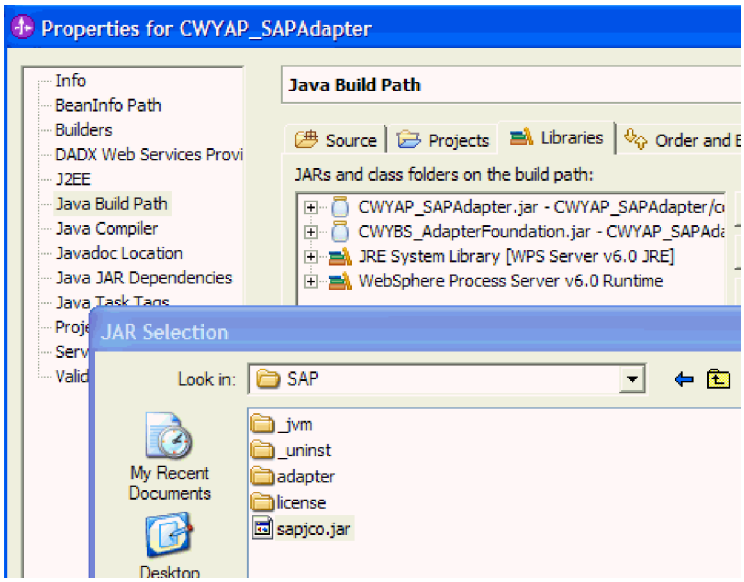


Figure 215. The JAR Selection window, with the sapjco.jar file highlighted for selection

f. Click **OK**.

The file sapjco.jar appears in the list of JARs and class folders in the build path.

Result

The sapjco.jar file is now part of your connector project and appears in the Project Explorer window of WebSphere Integration Developer.

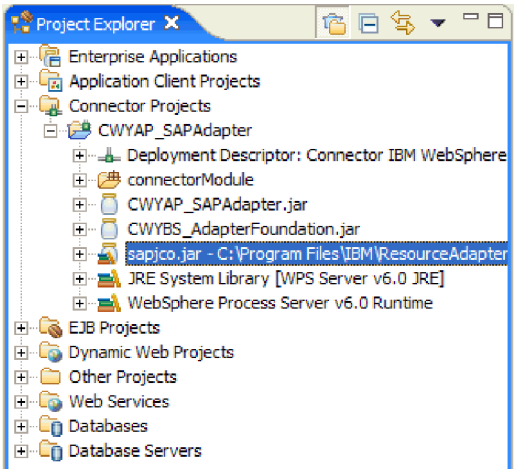


Figure 216. The Project Explorer window of WebSphere Integration Developer

Configuring the adapter for outbound processing

To configure the adapter, set the connection properties for enterprise service discovery. Then use the enterprise service discovery wizard to select and configure the necessary business objects and to generate a deployable module.

Setting connection properties for enterprise service discovery

To set connection properties for the enterprise service discovery wizard so that it can access the SAP server, specify such information as the user name and password you use to access the server as well as the name or IP address of the server.

1. In WebSphere Integration Developer, start the enterprise service discovery wizard.
   a. Switch to the Business Integration Perspective by clicking **Window** → **Open Perspective** → **Other**. Then click **Business Integration (default)** and click **OK**.
   b. Right-click **New** → **Enterprise Service Discovery**.

      If **Enterprise Service Discovery** is not displayed, click **New** → **Other**, expand **Business Integration**, and click **Enterprise Service Discovery**. Then click **Next**.

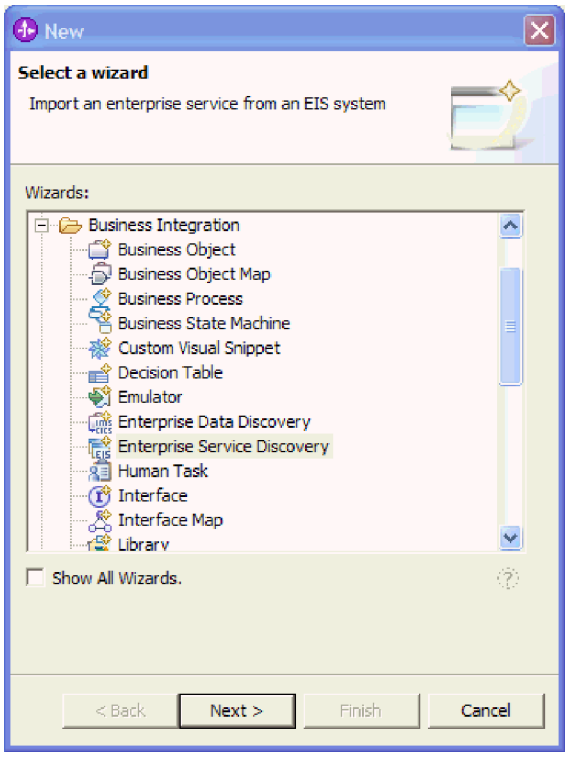


Figure 217. The expanded list of wizards

2. In the Select an Enterprise Service Resource Adapter window, make sure **IBM WebSphere Adapter for SAP Software** is selected, and click **Next**.
3. In the Configure Settings for Discovery Agent window, specify the configuration properties needed to connect to the SAP server.
   a. Type the name and password you use to access the SAP server.

      The password is case-sensitive.
   b. Type your client ID.

      This is typically `100`.
   c. In the **Application Server Host** field, type the name (or IP address) of your SAP server.

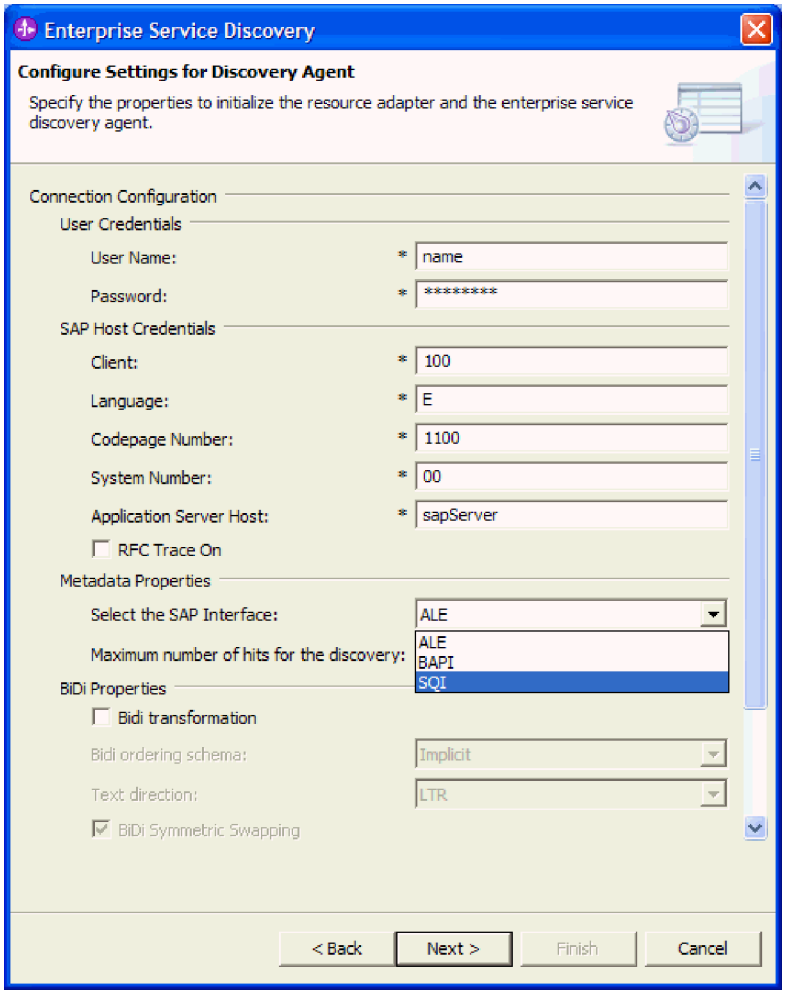


Figure 218. The Configure Settings for Discovery Agent window

4. Indicate which SAP interface you want to work with by selecting **SQI** from the **Select the SAP interface** list.
5. Set the logging level so that you can see any errors that might arise during enterprise service discovery.
   a. At the bottom of the Configure Settings for Discovery Agent window, click **Show Advanced**.

      The button changes to **Hide Advanced**.

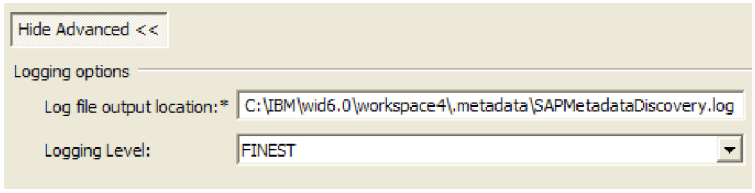


Figure 219. The Logging options displayed when you select Show Advanced

   b. For **Logging Level**, select **FINEST**.
6. Click **Next**.

Result

The enterprise service discovery wizard contacts the SAP server, using the information you provided (for user name and password) to log in. You see the Find and Discover Enterprise Services window.

Selecting business objects and services

To query data on the SAP server, specify search criteria that the enterprise service discovery wizard uses to find the data on the server.

1. From the Find and Discover Enterprise Services window, begin the process of searching for services by clicking **Execute Query**.
2. Expand **SQI** and click **Discover By Name**.

   The **Filter** button is now enabled.
3. Click **Filter**.
4. Select the KNA1 table.
   a. In the Filter Properties for Discover By Name window, type `KNA1`. Then click **OK**.
   b. Expand **Discover By Name (filtered)**.

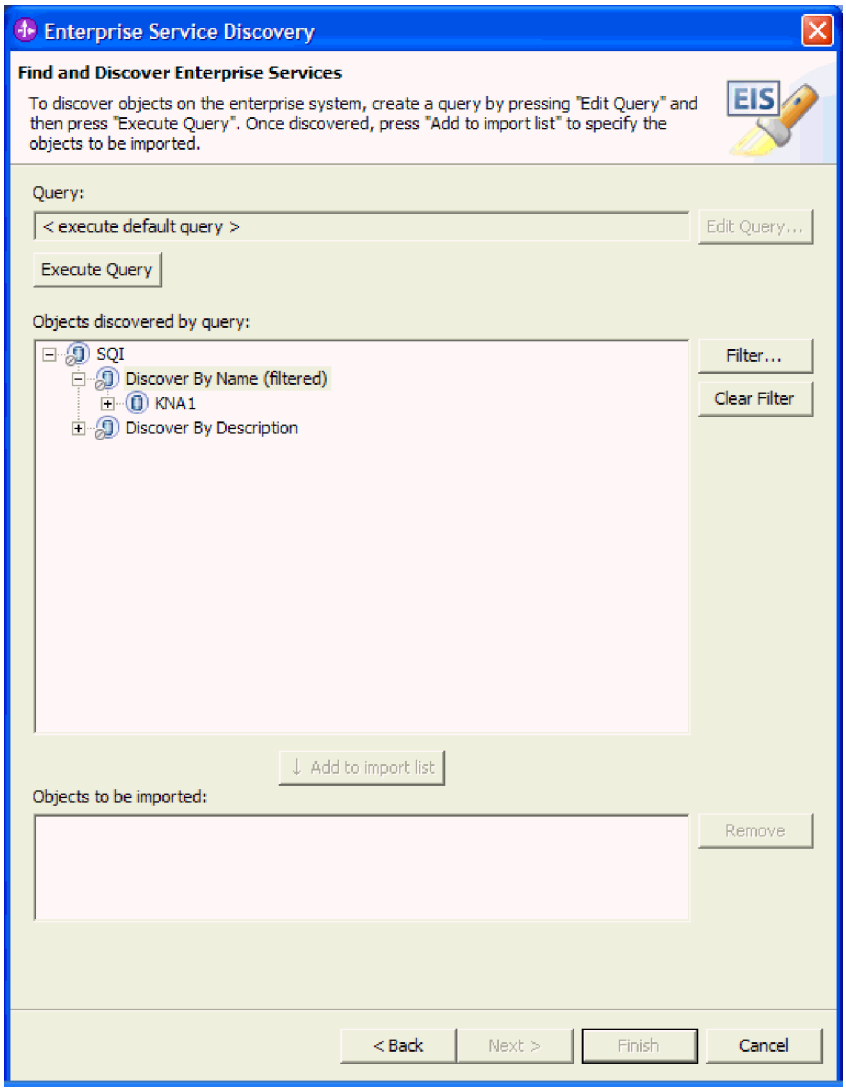


Figure 220. The Find and Discover Enterprise Services window

c. Select **KNA1** and click **Add to import list**.
d. In the Configuration Parameters for KNA1 window, accept the defaults by clicking **OK**.

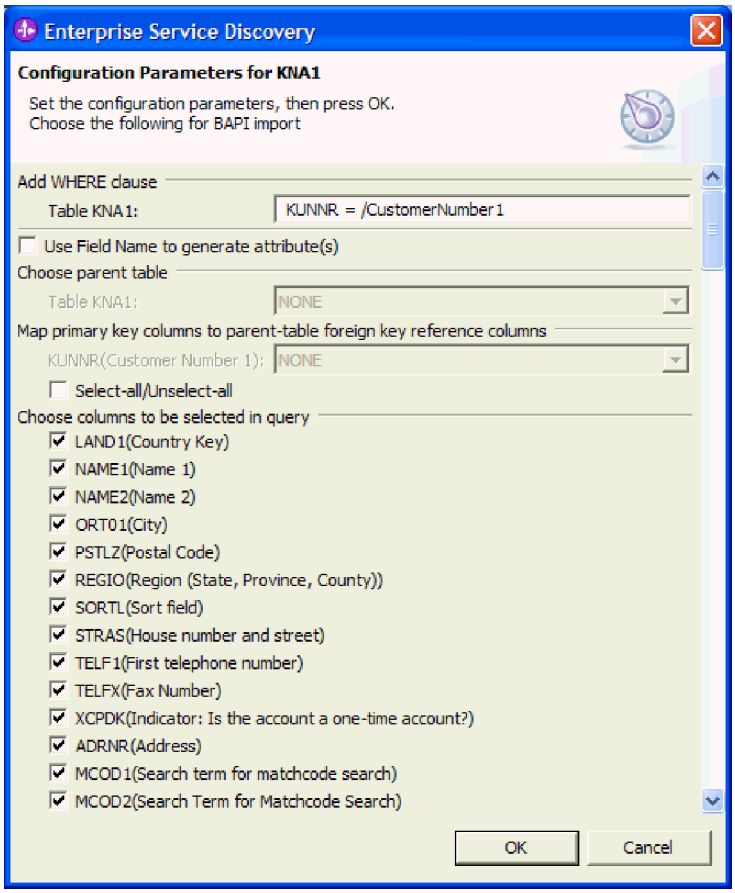


Figure 221. The Configuration Parameters window, which you leave blank to accept the defaults

5. Click **Discover By Name (filtered)** and then click **Filter**.
6. Select the ADRC table.
   a. Type `ADRC` and click **OK**.
   b. Select **ADRC** and click **Add to import list**.
   c. Under **Choose Parent Table**, select **KNA1**.
   d. Under **Map primary key columns to parent-table foreign key reference columns**, select **ADRNR** for **ADDRNUMBER**. Leave the default setting of **NONE** for the other columns.

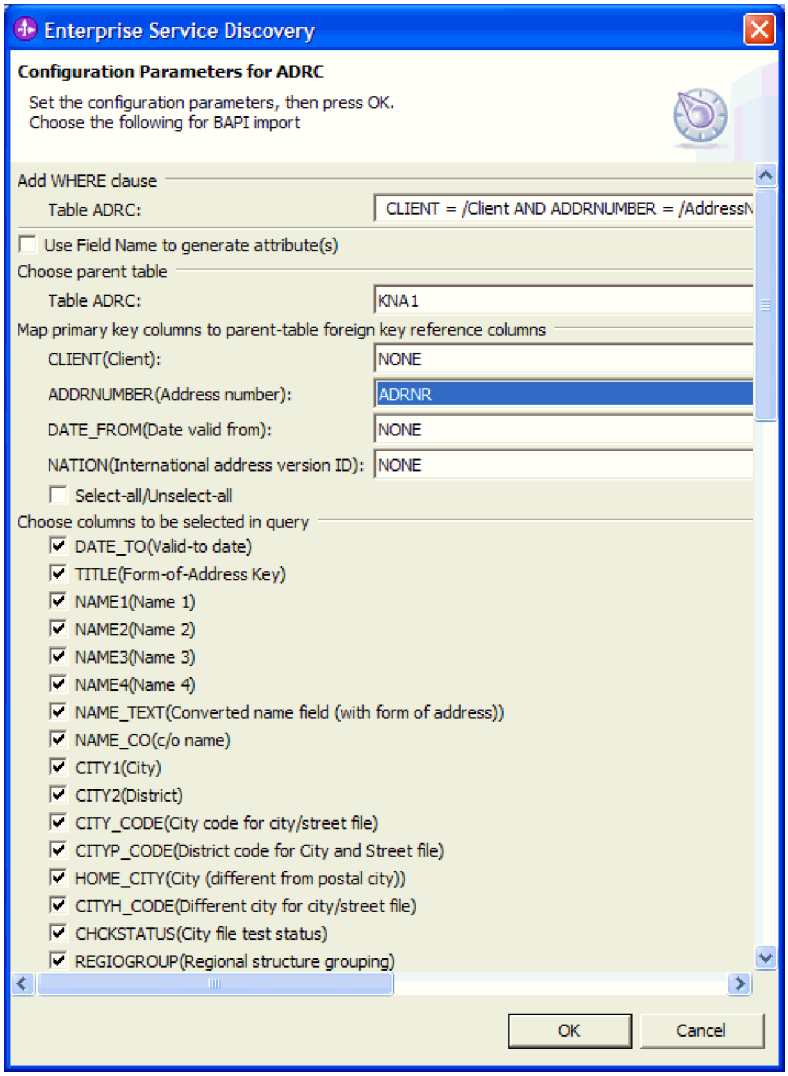


Figure 222. The Configuration Parameters window

e. Select the default settings for **Columns to be selected in query** by clicking **OK**.

7. Click **Next**.

Result

The enterprise service discovery wizard has found the tables you specified, and you have selected the data you want to query.

Configuring the selected objects

To configure the business object, you specify information about the object (such as the location of the object).

1. In the Configure Objects window, type `SQICustomer` for **Object Location**.
2. Click **Next**.

Result

You have selected the location where the object is stored. The Generate Artifacts window is displayed.

Generating artifacts

To generate the module, which is the artifact that can be exported to an EAR file for deployment, you create the module, include the adapter in the module, and specify an alias used to authenticate the caller to the SAP server.

1. In the Generate Artifacts window, create a new module.
   a. Click **New**.
   b. Click **Create a module project** and click **Next**.
   c. Type `SQICustomer`.

      As you type the name of the module, the name is added to the path next to **Directory**.
   d. Click **Finish**.
2. Accept the default value for **Name** and **Deploy connector with module**.
3. Indicate the authentication alias to use by typing the alias that you created (in the beginning of the tutorial) in the administrative console. In the example shown earlier, the alias is `widNode/SAP_Auth_Alias`.
4. Select **Use discovered connection properties**.

   When you select **Use discovered connection properties**, the entries you made earlier (such as user name and IP address) are displayed at the bottom of the window.
5. Click **Finish**.

Result

The new SQICustomer module is added to the Business Integration perspective.

Deploying the module for testing

To deploy the module to the test environment of WebSphere Process Server, you start the server and add the module (SQICustomerApp) to it. ″App″ is appended to the name of the module to indicate that the module is a deployable application.

1. Select the test environment server.
   a. Click the **Servers** tab.
   b. Right-click **WebSphere Process Server v6.0**.

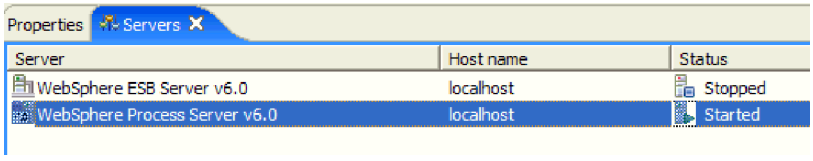


Figure 223. Selecting the WebSphere Process Server test environment from the Servers tab

2. Click **Add and remove projects**.
3. Select **SQICustomerApp** and click **Add**.
4. Click **Finish**.

Result

You see status messages in the **Console** tab as SQICustomerApp is deployed to the server.

Testing the module

Test the module to make sure you can retrieve data from the SAP server. You enter a customer number, and the data associated with that customer is returned.

About this task

To test this tutorial, you use an actual customer number. If you have not already done so, obtain a valid customer number. If necessary, see your SAP administrator to obtain the data.

How to perform this task

1. In the Business Integration perspective, begin the testing procedure by right-clicking **SQICustomer** and clicking **Test → Test Module**.

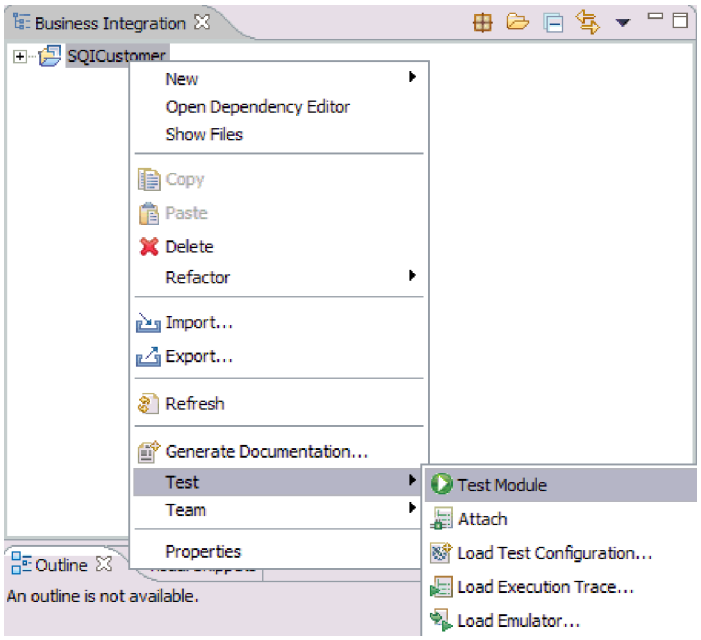


Figure 224. Specifying that you want to test SQICustomerApp

2. Set the parameters listed in the following steps and shown in the following figure:
   a. For **Operation**, select **retrieveallSapKna1**.
   b. In the Value column for **verb**, select **Retrieveall**.
   c. In the Value column for **CustomerNumber1**, type the number of an existing customer.

      The customer number is a ten-digit value. Type leading zeroes, if needed, to ensure that the number is ten digits. For example, if the customer number is 1, you would type `0000000001`.

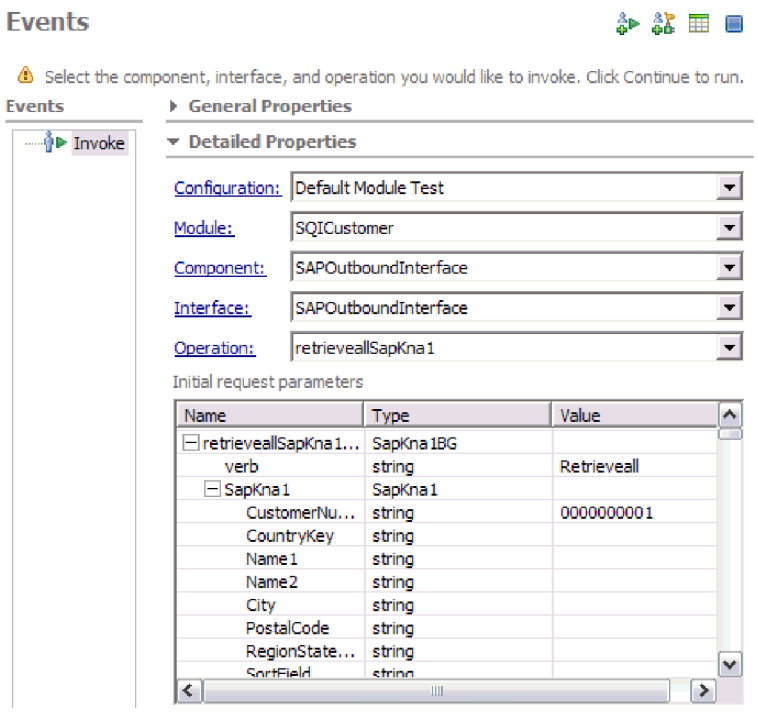


Figure 225. The Initial request parameters section of the test environment, with the sample values filled in

You can retrieve data for all customers by selecting **Retrieveall** for the verb and leaving the other fields in the table array blank. For example, if you wanted to retrieve data for all customers from the SapKna1 table, the Initial request parameters section would look like this:

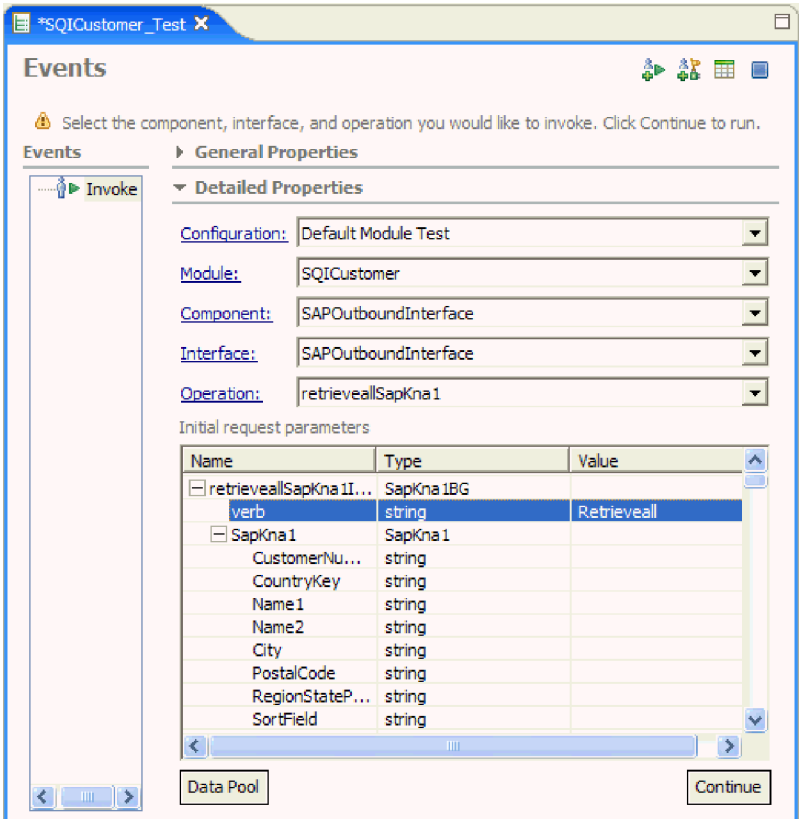


Figure 226. The Initial request parameters section for retrieving all customer data

3. Click **Continue**.
4. Click **Finish**. The integration test client invokes SQICustomerApp.

Result

Data associated with the customer number you entered is displayed under **Return parameters**.

Troubleshooting the tutorial

If you have problems while running the tutorial, check to see whether the sapjco.jar file is in the correct place. If you are unable to deploy the module using WebSphere Integration Developer, use the administrative console of WebSphere Process Server.

Checking the placement of sapjco.jar

If you see the error message `com.sap.mw.jco class not found` when you attempt to connect to the SAP server during enterprise service discovery, check that the sapjco.jar file was installed into the correct directory and imported into the adapter project.

1. Make sure the sapjco.jar file is in the \runtimes\bi_v6\lib directory of the WebSphere Integration Developer installation directory. If it is not, move it there now.
2. Check to see whether the sapjco.jar file has been imported into the adapter project.

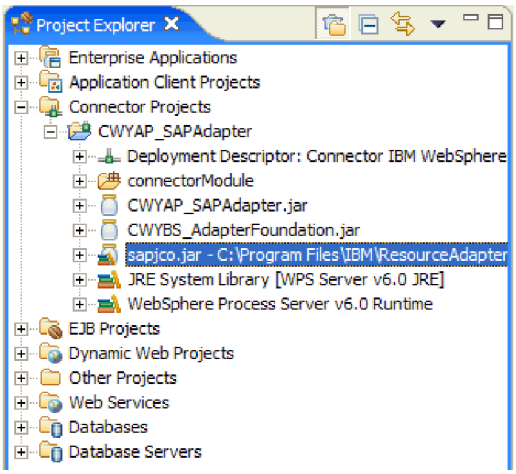


Figure 227. The Project Explorer window of WebSphere Integration Developer

3. If the sapjco.jar file does not appear in the adapter project, use the following procedure.
   a. Right-click **CWYAP_SAPAdapter** and click **Properties**.
   b. In the left side of the Properties for CWYAP_SAPAdapter window, click **Java Build Path**.
   c. Click the **Libraries** tab, and click **Add External JARs**.
   d. Navigate to the directory on your local file system where the sapjco.jar file is located. Then select **sapjco.jar** and click **Open**.
   e. Click **OK**.

Deploying the project through the administrative console

If you are unable to test the tutorial through the WebSphere Integration Developer test environment, export the adapter project as an EAR file and then deploy the EAR file, using the administrative console.

- Export the adapter project as an EAR file.
  1. From the Business Integration perspective, right-click the name of the adapter project, and select **Export**.
  2. Select **EAR file** and click **Next**.
  3. Specify which module to use by clicking the down arrow in the **EAR project** field and selecting the name of the adapter project.

     Notice that **App** is appended to the name, indicating that the module is a deployable application.
  4. Indicate where the file should be saved by clicking **Browse** and selecting the location.
  5. Click **Finish**.
- Deploy the EAR file to the administrative console.
  1. Click **Servers** in WebSphere Integration Developer.
  2. If it is not already started, right-click the WebSphere Process Server instance and start the instance.
  3. Confirm that the status of the server is **Started**.
  4. Right-click the server and click **Run administrative console**.
  5. Click **Log in** in the administrative console window.
  6. Select **Applications** → **Enterprise Applications**.

7. Select **Install**.
8. In the Path to the new application window, browse to the system where the EAR file was saved.
9. Select the EAR file, and click **Open**.
10. Click **Next**.
11. Click **Next**.
12. In the Install new application window, click the **Summary** step.
13. Click **Finish**. The project is installed.
14. Click **Save to Master Configuration**.
15. Click **Save**.

Chapter 12. Viewing the sample adapter artifacts

To view the sample artifacts for each tutorial, import into IBM WebSphere Integration Developer the quick start reference files included with the adapter. Note that these artifacts are for reference only. They probably will not execute in your enterprise information system environment. If you have not stepped through the tutorials, you can still use the reference files to view examples of correctly-generated artifacts before you create your own.

Before you begin

Locate the quick start reference files in the referencefiles subdirectory of the samples directory. There is a project interchange zip file for each quick start tutorial. For instance, Tutorial1.zip is for quick start tutorial 1.

Important: Do not modify or use the artifacts provided in the quick start reference files. They are provided exclusively for viewing.
Reference files do not include third-party libraries. When imported into IBM WebSphere Integration Developer, the reference files might generate compilation error messages because dependent libraries are missing. The artifacts in the reference files may not be compatible with the enterprise information system (EIS) you are using. They vary based on EIS version and configuration.

The artifacts were generated with version ECC 6.0 of SAP Software.

About this task

Import the quick start reference files into WebSphere Integration Developer to view sample artifacts associated with each quick start tutorial.

How to perform this task

1. In the Business Integration perspective of WebSphere Integration Developer, click **File → Import**.
2. In the Import window, select **Project Interchange** and click **Next**.
3. Select the project interchange file containing the tutorial artifacts you want to view.
4. Import all the projects in the project interchange file by clicking **Select All**.
5. Click **Finish**.

Result

A business integration module is created with the following artifacts:

- Service import and export definitions
- Business objects (service data objects)
- Interfaces

Chapter 13. Reference information

Detailed information about business objects, external software dependences, adapter properties (enterprise service discovery properties, resource adapter properties, managed (J2C) connection factory properties, and activation specification properties), messages, and related product information is provided for your reference.

Business objects

A business object contains application-specific information (metadata) about processing the business object as well as the operation to be performed on the business object.

Metadata of business objects

The enterprise service discovery wizard automatically generates an XSD file that contains application-specific information (metadata) for business objects. You can view and, if necessary, modify the values of the metadata.

Metadata of BAPI business objects

The metadata generated by enterprise service discovery provides the adapter with instructions on how to process BAPI business objects. Metadata is generated at the business-object level, the operation level, and the property level.

You can view (and modify) the metadata values associated with the business object. Use the Properties tab within WebSphere Integration Developer to modify the values.

Note: Do not change the metadata element name.

Business-object-level metadata

Business-object-level metadata for BAPI business objects defines the top-level wrapper.

The following table describes the business-object metadata elements of a simple BAPI business object.

Table 25. Metadata for wrapper of a simple BAPI business object

Metadata element	Description
Type	The business object type. For a simple BAPI, the value is set to `BAPI`.
Operation	The valid operations include Create, UpdateWithDelete, and Delete. The specified operation metadata is defined in the sapBAPIOperationTypeMetadata tag and contains the following: • `Name`: Name of the operation. • `MethodName`: Name of the BAPI associated with the operation.

The following table describes the business-object metadata elements of a BAPI transaction.

Table 26. Metadata for a BAPI transaction business object

Metadata element	Description
Type	The business object type. For a BAPI transaction business object, this value is always `BAPITXN`.
Operation	The valid operations include Create, UpdateWithDelete, and Delete. The specified operation metadata is defined in the sapBAPIOperationTypeMetadata tag and contains the following parameters: • `Name`: Name of the Operation. • `MethodName`: Name of the BAPI associated with the operation.

Property-level metadata

Property-level metadata can represent child objects or an array of child objects.

The following table describes the metadata elements of a complex property (child) or structure/table property (array of child objects).

Table 27. Property-level metadata

Metadata element	Description
FieldName	The BAPI field name as represented in SAP.
FieldType	The type of the property as it exists in SAP.
PrimaryKey	A boolean that indicates whether this property is a primary key.
ParameterType	Identifies the direction of the mapping. • If the value is `IN`, the property is mapped from the business object to the BAPI. • If the value is `OUT`, the property is mapped from the BAPI in the SAP system to the business object. • If the value is `INOUT`, the property is mapped both ways (BAPI to business object and business object to BAPI).

Operation-level metadata

The metadata for an operation specifies the method name of the BAPI in the SAP system. This name is used by the adapter to execute the BAPI.

The following table describes the operation-level metadata elements of a BAPI business object.

Table 28. Operation-level metadata

Metadata element	Description
MethodName	The name of the BAPI call (method) in the SAP system.
Name	The name of the business object operation associated with the MethodName.

Metadata of ALE business objects

The metadata generated by enterprise service discovery provides the adapter with instructions on how to process ALE business objects. The type of metadata generated depends on whether the business object represents an individual IDoc or an IDoc packet.

You can view (and modify) the metadata values associated with the business object. Use the Properties tab within WebSphere Integration Developer to modify the values.

Note: Do not change the metadata element name.

Metadata is specified at the following levels:

- IDoc business object-level (for individual IDocs)
- IDoc wrapper business object-level (for IDoc packets)
- Operation-level for individual IDoc business objects
- Property-level

Note: There is no metadata at the IDoc Data Record or IDoc Control Record child business object level.

Business-object-level metadata

Business object-level metadata for ALE business objects defines the top-level wrapper of an IDoc.

The following table describes the business-object metadata elements of an ALE business object.

Table 29. Business-object level metadata

Metadata element	Description
Type	The business object type. Possible values are `IDOC` or `UNPARSEDIDOC`.
Operation	Each *outbound* operation contains the following parameters: **Name** Name of the operation, which for outbound processing is always Execute. Each *inbound* operation contains the following parameters: **Name** Name of the operation (Create, Updatewithdelete, or Delete). **MsgType** The message type configured for the IDoc. **MsgCode** The message code configured for the IDoc. **MsgFunction** The message function configured for the IDoc.
SplitIDocPacket	For inbound operations, an indication of whether the IDoc packet needs to be split into individual IDocs. The possible values are `true` or `false`.

Property-level metadata

The following table describes the property-level metadata elements of an ALE business object.

Table 30. Property-level metadata

Metadata element	Description
FieldName	The actual IDoc field name in SAP.
SegmentHierarchy	The hierarchy of the segment in the IDoc.
Offset	The offset value of the current property in the IDoc.
PrimaryKey	An indication of whether this property is a primary key.
ForeignBOKeyRef	Used for the DummyKey property to hold the xpath to the primary key on the control or data record business object property, which you set using the business object editor in the enterprise service discovery wizard.

Operation-level metadata

The operation-level metadata for an ALE business object specifies the operation that posts the IDoc object to the SAP application.

The following table describes the application-specific metadata elements of an ALE business object operation.

Note: Outbound objects use only the Name metadata element. The MsgType, MsgCode, and MsgFunction elements are used for inbound objects only.

Table 31. Operation-level metadata

Metadata element	Description
Name	The name of the operation.
MsgType	The message type configured for the IDoc (for inbound objects only).
MsgCode	The message code configured for the IDoc (for inbound objects only).
MsgFunction	The message function configured for the IDoc (for inbound objects only).

Metadata of SQI business objects

The metadata generated by enterprise service discovery provides the adapter with instructions on how to process SQI business objects. Metadata is generated at the business-object level, the operation level, and the property level.

Business-object-level metadata

You can view (and modify) the metadata values associated with the business object. Use the Properties tab within WebSphere Integration Developer to modify the values.

Note: Do not change the metadata element name.

The following table describes the business-object metadata elements of an SQI business object.

Table 32. Business-object level metadata

Metadata element	Description
TableName	The name of the table that this business object represents.

Table 32. Business-object level metadata (continued)

Metadata element	Description
DataDelimiter	The character used as the delimiter in parsing returned data. The default value is `|`. The enterprise service discovery generates the default value.
Type	The interface type the business object is supporting, which for the SAP query interface is `SQI`.

Property-level metadata

The following table describes the property-level metadata elements of an SQI business object.

Table 33. Property-level metadata

Metadata element	Description
ColumnName	The name of the business-object parameter, which is the actual column name in the SAP table.
PrimaryKey	An indication of whether this property is a primary key.
ForeignKey	Provides the foreign key relationship (if IsKey is true), which is the reference to the parent table key parameter.

Operation-level metadata

The operation-level metadata for an SQI business object specifies which data should be returned from the SAP table.

The following table describes the operation-level metadata elements of an SQI business object.

Table 34. Operation-level metadata

Metadata element	Description
sapWhereClause	Retrieves information from SAP tables. The default value is populated by enterprise service discovery.
maxRows	The maximum number of rows to be returned. The default value is `100`.
rowsSkip	The number of rows to skip before retrieving data. The default value is `0`.

Operations and verbs

An operation reflects the action to be performed on the business object by the adapter. The verb of the business object reflects its state and is defined at the business graph level for after-image objects only. BAPI business objects have operations and verbs associated with them. ALE business objects also have operations and verbs associated with them, although the verb for an outbound ALE business object is not used in outbound processing. SQI business objects have only operations associated with them.

Operations and verbs of BAPI business objects

BAPI business objects support operations and verbs. The operation of a BAPI business object specifies the BAPI call to execute in the SAP system for that object.

The BAPI method determines the operation associated with it. The verb of a BAPI business object specifies the state of the object.

Supported operations

Operations of a business object are invoked by the client component that makes calls to SAP through the adapter. The SAP JCo APIs are used to make the call to the SAP system.

The following table defines operations that the adapter supports.

Note: The definitions listed in the table are the *expected* uses for the operations. The action taken in the SAP application is based on the meaning of the BAPI itself.

Table 35. Supported operations: BAPI business objects

Operation	Definition
After-Image Create	Creates a new entity in SAP that matches the data and structure of the input business object. The business object returned by this operation reflects the newly created entity in SAP.
After-Image Update with Delete	A special form of the UpdateWithDelete operation that provides better performance. It always requires a ChangeSummary, which is expected to include information about business object-level creations and deletions. This enables the adapter to perform operations without the overhead of retrieving the existing entities from SAP and doing comparisons, because the ChangeSummary indicates what needs to be done. If the ChangeSummary is empty, the adapter does not take any action on the request.
Retrieve	Rebuilds the complete business-object hierarchy. The adapter ensures that the returned hierarchical business object exactly matches the database state of the application entity. Non-key values can be used as criteria. Accepts either an after-image or business object. The comparison in either case is by equality only. The request business object might contain any of the following: • A top-level business object but no child objects, even though the business object definition includes children • A business object that contains the top-level business object and some of its defined children • A complete hierarchical business object containing all child business objects Retrieve is intended to return a single, unique business object that meets user-defined criteria. The requirement for performing the retrieve operation is totally dependent on the BAPI. Whatever the BAPI deems as “required” is what allows the retrieve to be successful.
After-Image Delete	Removes an existing entity from SAP and any contained child entities. Note that SAP has the concept of a logical delete, whereby the record is marked as deleted but the entity still exists. This is done in some cases to maintain database integrity because the entity “deleted” is referenced in other entities. Therefore, After-Image Delete behavior depends on the BAPI call.

For an operation that is not supported or does not match the verb in the business graph, the adapter logs the appropriate error and produces a ResourceException.

Supported verbs

The following table lists the verbs that the adapter supports for BAPI business objects.

Table 36. Supported verbs: BAPI business objects

Verb	Definition
Create	The top-level business object and all contained children have been created, if this is an inbound event, or should be created in SAP, if this is an outbound request.
UpdateWithDelete	The top-level business object has been or should be modified; this may include addition or deletion of children as well.
Delete	The top-level business object and any contained children have been or should be deleted.

For verbs that are not supported, the adapter produces a ResourceException error.

Operations and verbs of ALE business objects

ALE IDoc business objects support operations and verbs. The operations that are supported vary, depending on whether the business object is an outbound or inbound object. Verbs are used in inbound business object processing only; they are ignored in outbound business objects.

Supported operations

The operation of an ALE outbound business object is invoked by the client application that makes calls to SAP through the adapter. The client must be designed so that the calls made by its InteractionSpec implementation invoke the operations. The adapter supports the following outbound operation.

Table 37. Supported operation: ALE outbound business objects

Operation	Definition
Execute	Posts the IDoc business object to the SAP application. This is a one-way, asynchronous operation. • If you are using the CWYAP_SAPAdapter.rar version of the adapter, no response is sent back. • If you are using the CWYAP_SAPAdapter_TX.rar version of the adapter, the transaction ID is returned.

For all other operations, the adapter logs the appropriate error and raises a ResourceException.

For ALE inbound business objects, the application-specific information of an operation is used to set the verb of the business graph. The application-specific information of an operation contains the message type, message code, and message function for an IDoc type. The business graph verb corresponds to the operation that has application-specific information matching the Control record fields. The adapter supports the following inbound operations.

Table 38. Supported operations: ALE inbound business objects

Operation	Definition
AfterImageCreate	The top-level business object and all contained children are created.
AfterImageUpdateWithDelete	The top-level business object is or should be modified. This operation can include adding and deleting child objects. Any deleted child objects are known and reflected.
AfterImageDelete	The top-level business object and any contained children are or should be deleted.

Supported verbs

ALE inbound business objects support the following verbs:

Table 39. Supported verbs: ALE inbound business objects

Verb	Definition
Create	The top-level business object and all contained children are created.
UpdateWithDelete	The top-level business object is or should be modified. This can include adding and deleting child objects. Any deleted child objects are known and reflected.
Delete	The top-level business object and any contained children are or should be deleted.

The adapter sets the verb property of the business graph for the ALE business object before sending the business object to the endpoint. The adapter determines the verb by comparing the metadata defined in the application-specific information of the business object operations with the following IDoc Control Record fields:

- Logical_message_type (MESTYP)
- Logical_message_code (MESCOD)
- Logical_message_function (MESFCT)

ALE outbound business objects provide no verb support. The adapter ignores the value in the verb property of the business object graph.

Operations of SQI business objects

The SAP Query interface supports the RetrieveAll operation, with which you can have the results of an SAP table returned to you, and the Exists operation, which you use to determine whether data can be found in the SAP table.

The supported operations for the SAP Query interface are listed in the following table.

Table 40. Supported operations: SQI business objects

Operation	Description
RetrieveAll	Returns a result set in the form of a container of SAPQueryBGs, which represent the data for each row retrieved from the table.

Table 40. Supported operations: SQI business objects (continued)

Operation	Description
Exists	Provides a means to check for the existence of any records in SAP for a defined search criteria. Exists does not return any data; it indicates whether the data exists in SAP. If no data is found, the adapter generates an exception.

Naming conventions

When the enterprise service discovery wizard generates business objects, it names them according to a convention. A prefix of `Sap` is automatically added to the beginning of the business object name, for example. The naming convention depends on whether the name is for a BAPI business object, an ALE business object, or an SQI business object.

Naming conventions for BAPI business objects

The enterprise service discovery wizard provides names for the BAPI business graph, top-level business object, the business object itself, and any child objects.

The following table describes the naming convention that the enterprise service discovery wizard uses for the business object of a simple BAPI.

Table 41. Naming conventions for BAPI business objects

Element	Naming convention
Name of the BusinessGraph	Sap + *Name of the wrapper object you specify in the enterprise service discovery wizard* + Wrapper + BG For example: `SapSalesOrderWrapperBG`
Name of the top-level business object	Sap + *Name of the wrapper object you specify in the enterprise service discovery wizard* + Wrapper For example: `SapSalesOrderWrapper`
Name of the BAPI business object	Sap + *Name of the BAPI interface* For example: `SapBapiSalesOrderCreateFromDat1` **Note:** The top-level object can contain more than one individual BAPI object.
Name of the child object	Sap + *Name of the Structure/Table* For example: `SapReturn` Note that in the case of structures having the same name in different BAPIs or the same structures within a BAPI (for example, one at the export level and one at the table level), the enterprise service discovery wizard generates a unique numeric code and increments the name of the structure that occurs more than once. For example: `SapReturn619647890`, where 619647890 is the unique identifier suffix appended to the name SapReturn.

Naming conventions for ALE business objects

The enterprise service discovery wizard provides names for the ALE business graph, top-level business object, and the business object itself.

The following table shows the naming convention for an ALE business graph, a top-level wrapper object, and an IDoc object. Note that *[Name of Extension type*

IDoc] in the Naming convention column represents an optional entry. It is included in the name only if the selected IDoc is an Extension Type IDoc.

Table 42. Naming conventions for ALE business objects

Element	Naming convention
Name of the BusinessGraph	Sap + *NameofIDoc* + *[Name of Extension type IDoc]* + BG For example: SapAlereq01BG
Name of the top-level wrapper object	Sap + *NameofIDoc* + *[Name of Extension type IDoc]* For example: SapAlereq01
Name of the IDoc business object	Sap + *NameofIDoc* + *[Name of Extension type IDoc]* + IDocBO For example: SapAlereq01IDocBo In the case of an IDoc duplicate name, the enterprise service discovery wizard adds a unique suffix to distinguish the business object. If an IDoc packet has two segments with the same name (for example segOrder), the first business object is assigned the name SapSegOrder and the second business object is assigned a name such as SapSegOrder619647890, where 619647890 is the unique identifier suffix appended to the name by the enterprise service discovery wizard.

Naming conventions for SQI business objects

The enterprise service discovery wizard provides names for the SQI business graph, top-level business object, table object, and query object.

The following table describes the naming convention that the enterprise service discovery wizard uses for the business object of an SQI.

Table 43. Naming conventions for SQI business objects

Element	Naming convention
Name of the BusinessGraph	Sap + *Name of the wrapper object you specify in the enterprise service discovery wizard* + BG For example: SapCustomerBG
Name of the top-level wrapper object	Sap + *Name of the object you specify in the enterprise service discovery wizard* For example: SapCustomer
Name of the table object	Sap + *Name of the SAP table*
Name of the query object	Sap + *Name of the SAP table* + Querybo

External software dependencies

WebSphere Adapter for SAP Software requires the sapjco.jar file to communicate with the SAP system. In addition, the adapter has requirements on external dependencies libraries.

To satisfy the external software dependencies of the Adapter for SAP Software, you must copy the following files to the specified locations during adapter configuration.

Table 44. External dependencies for the WebSphere Adapter for SAP Software

File name	Target directory
Windows - *.dll files that come with the SAP Jco download from the SAP Web site Unix (including Unix System Services on z/OS) - *.so and *.o files (if they exist) that come with the SAP Jco download from the SAP Web site	bin directory of WebSphere Process Server or WebSphere Enterprise Service Bus For z/OS, add the files to the ${*WAS_INSTALL_ROOT*}/lib directory.
msvcp71.dll and msvcr71.dll	Windows/system
sapjco.jar	lib directory of WebSphere Process Server or WebSphere Enterprise Service Bus For z/OS, add ${*WAS_INSTALL_ROOT*}/lib/sapjco.jar to WAS_SERVER_ONLY_server_region_classpath

Adapter configuration properties

WebSphere Adapter for SAP Software has several categories of configuration properties, some of which you set during the enterprise service discovery process and some of which you can set or change after you deploy the adapter application to the WebSphere Process Server or WebSphere Enterprise Service Bus.

Enterprise service discovery connection properties

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.

Table 45. Enterprise service discovery connection properties for Adapter for SAP Software

Property	Type	Description	Default value
User Name	String	Name of the adapter user account on the SAP system.	
Password	String	Password of the adapter user account on the SAP system.	
Client	Integer	SAP client number under which the adapter logs in. This value is often `100`.	
Language	String	Language in which the adapter logs in.	`E`, for English
Codepage Number	Integer	This value must correspond to the value specified in the Language property. For example, if Language is set to `JA` (Japanese), then Codepage Number must be set to `8000`, as dictated by the SAP application.	`1100`
System Number	Integer	System number of the application server. The value, which is a two-digit number, is often `00`.	`00`

Table 45. Enterprise service discovery connection properties for Adapter for SAP Software (continued)

Property	Type	Description	Default value
Application Server Host	String	When the adapter is configured to run without load balancing, specifies the IP address or the name of the application server that the adapter logs into. In both cases, the adapter assumes that the name of the gateway host is the same as the value specified for this property.	
RFC Trace On	Boolean	Specifies whether to generate a text file detailing the RFC activity for each event listener. You can specify a value of true (checked) or false (unchecked). A value of true activates tracing, which generates a text file. Use these text files in a development environment only, because the fields can grow rapidly.	`False` (unchecked)
Select the SAP Interface	String	Indicates whether you are creating business objects for the ALE, BAPI, or SAP Query interface. Possible values are `ALE`, `BAPI`, and `SQI`.	`ALE`
Maximum number of hits for the discovery	Integer	Defines the maximum number of SAP elements displayed by the wizard per discovery. Possible values are `50`, `100`, `300`, `500`, and `All`.	`100`
Bidi transformation	Boolean	Turns bidirectional support on or off.	`False` (bi-di support is turned off)
Bidi ordering schema	String	Determines the type of text schema used–either `Implicit` (logical) or `Visual`.	`Implicit`
Text direction	String	Determines the text direction used. Possible values are `LTR` (left to right), `RTL` (right to left), `ContextualLTR` (contextual left to right), and `ContextualRTL` (contextual right to left.)	`LTR`
BiDi Symmetric Swapping	Boolean	Determines whether symmetric swapping is turned on or off.	`True` (symmetric swapping is turned on)
Bidi shaping	String	Determines the bi-di format used by the enterprise service discovery wizard when it communicates with the SAP application. Possible values are `Initial`, `Nominal`, `Shaped`, `Final`, `Middle`, and `Isolated`.	`Nominal`
Bidi numeric shaping	String	Determines the bi-di format used by the enterprise service discovery wizard when it communicates with the SAP application. Possible values are `Nominal`, `National`, and `Contextual`.	`Nominal`
Log file output location	String	Specifies the location of the log file for enterprise service discovery.	The .metadata directory of the workspace.
Logging Level	String	Specifies the type of information logged during enterprise service discovery. Possible values are `ALL`, `OFF`, `FINE`, `FINER`, `FINEST`, `CONFIG`, `INFO`, `SEVERE`, and `WARNING`.	`SEVERE`

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. You configure these properties using WebSphere Process Server administrative console.

When you configure the adapter, specify the resource adapter properties listed below.

Table 46. Resource adapter properties for the Adapter for SAP Software

Property name in enterprise service discovery	Property name in administrative console	Type	Description	Default value
EIS BiDi Format	biDiContextEIS	String	The bi-di format used by SAP for its business data.	
EIS BiDi Special Format	biDiContextSpecialFormat	String	Signifies a category of values that are subject to special treatment during invocation of bi-di transformation to ensure accurate transformation of the category. Categories are predefined (for example, FTP URLs and e-mail addresses).	
	biDiContextTurnBiDiOff	Boolean	A flag used to turn off (explicitly exclude) bi-di support. This property takes precedence over the BiDiSkip property, and it allows users who do not require bidirectional script data support to turn it off.	
	enableHASupport	String	When the enableHASupport property is set to `true`, 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 might result in event duplication. Do not change the value of enableHASupport to `false` for single server environments.	`True`
Log file name	logFilename	String	The full path of the log file. This property is required.	
Log file size	logFileSize	Integer	Size of the log files in kilobytes. If no value is specified, the file has no maximum size. This property is not required.	

Table 46. Resource adapter properties for the Adapter for SAP Software (continued)

Property name in enterprise service discovery	Property name in administrative console	Type	Description	Default value
Log files	logNumberofFiles	Integer	The number of log files to use. When a log file reaches its maximum size, the adapter starts using another log file. If no value is specified, the number is set to 1. This property is not required.	
Metadata BiDi Format	biDiContextMetadata	String	Specifies the bi-directional format of meta configuration data in the SAP system.	
Skip BiDi Transformation	biDiContextSkip	Boolean	Controls invocation of bi-di transformation. Acceptable values: true or false. A blank value invokes the lookup mechanism.	
Trace file name	traceFileName	String	The full path to the trace file. This property is required.	
Trace file size	traceFileSize	Integer	Size of the trace files in kilobytes. If no value is specified, the file has no maximum size. This property is not required.	
Trace files	traceNumberOfFiles	Integer	The number of trace files to use. When a trace file reaches its maximum size, the adapter starts using another trace file. If no value is specified, the number is set to 1. This property is not required.	

Managed (J2C) connection factory properties

ManagedConnectionFactory (MCF) configuration 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.

When you configure the adapter, specify the properties listed below.

Note: The enterprise service discovery wizard refers to these properties as managed connection properties and WebSphere Process Server refers to these as (J2C) connection factory properties.

Table 47. Managed (J2C) connection factory properties for Adapter for SAP Software

Property name in enterprise service discovery	Property name in administrative console	Type	Description	Default value
ABAP Debug	ABAPDebug	Boolean	This configuration property is supported in a Windows environment only. The property specifies whether the adapter invokes the ABAP Debugger for the appropriate function module when the adapter begins processing a business object. When the property is set to `true`, the adapter opens the ABAP Debugger. You must have proper user authorizations to use the debugger. You can add breakpoints only after the debugger opens. **Important**: This property should always be set to `false` in a production environment.	`False`
Application Server Host	ApplicationServerHost	String	When the adapter is configured to run without load balancing, this property specifies the IP address or the name of the application server that the adapter logs in to. Regardless of whether it is configured for load balancing, the adapter assumes that the name of the gateway host is the same as the value specified for this property.	
Client	Client	String	SAP Client number under which the adapter logs in. This number is often `100`.	
Codepage Number	codepage	Integer	Establishes a connection to the appropriate language. This property must correspond to the value specified in the Language property. For example, if Language is set to `JA` (Japanese), then Codepage must be set to `8000`, as dictated by the SAP application. Refer to the SAP documentation for the exact Language and Codepage values.	
EIS BiDi Format	biDiContextEIS	String	The bi-di format used by SAP for its business data.	`ILYNN`

Table 47. Managed (J2C) connection factory properties for Adapter for SAP Software (continued)

Property name in enterprise service discovery	Property name in administrative console	Type	Description	Default value
EIS Bidi Special Format	biDiContextSpecialFormat	String	Signifies a category of values that are subject to special treatment during invocation of bi-di transformation to ensure accurate transformation of the category. Categories are predefined (for example: FTP, URLs, and e-mail addresses).	
Gateway Host	GatewayHost	String	Host where the gateway service is running.	
Gateway Service	GatewayService	String	The gateway server identifier; often `sapgw00`. The 00 is the system number of the server running the SAP Gateway (usually an application server). The value cannot be 00 if there is more than one server.	`sapgw00`
Ignore BAPI Return		Boolean	When this property is set to `true`, the BAPI ignores any errors found in the BAPI RETURN structure and returns the full business object. When this property is set to `false`, the adapter generates an exception if the BAPI RETURN contains an error code.	`False`
Language	Language	String	Language in which the adapter logs in.	`E`, for English
Message Server Host	MessageServerHost	String	When the adapter is configured for load balancing, this property specifies the name of the message server.	
Metadata BiDi Format	biDiContextMetadata	String	Specifies the bi-di format of meta configuration data in the SAP system.	`ILYNN`
Partner CharSet	partnerCharset	String	Specifies PartnerCharset encoding. When an encoding is specified, it is used; otherwise the default encoding is used.	`None`

Table 47. Managed (J2C) connection factory properties for Adapter for SAP Software (continued)

Property name in enterprise service discovery	Property name in administrative console	Type	Description	Default value
Password	Password	String	The password for the user account of the adapter on the SAP system. If bidirectional language support is enabled, this property is affected by the following properties, which are set using the enterprise service discovery wizard: • **Password BiDi Format**: Controls the bi-di format for this property. • **Skip BiDi Transformation for Password**: Controls invocation of bi-di transformation for this property.	
Password BiDi Format	biDiContext_PasswordEIS	String	Controls the bi-di format for the Password property.	`ILYNN`
RFC Trace On	RFCTraceOn	Boolean	Specifies whether to generate a text file detailing the RFC activity for each event listener. You can specify a value of `true` or `false`. A value of `true` activates tracing, which generates a text file. Use these text files in a development environment only, because the files can grow rapidly.	`False`
System ID	SAPSystemID	String	When the adapter is configured for load balancing, this property specifies the logical name of the SAP system (also known as R3*name*).	
Skip BiDi Transformation	biDiContextSkip	Boolean	Controls invocation of bi-di transformation. Possible values are `true` or `false`. A blank value invokes the lookup mechanism.	`True`
Skip BiDi Transformation for Password	biDiContext_PasswordSkip	Boolean	Controls invocation of bi-di transformation for the Password property. Possible values are `true` or `false`. A blank value invokes the lookup mechanism.	`True`
Skip BiDi Transformation for User Name	biDiContext_UserNameSkip	Boolean	Controls invocation of bi-di transformation for the Username property. Possible values are `true` or `false`. A blank value invokes the lookup mechanism.	`True`
System Number	SystemNumber	Integer	System number of the application server. The value is a two-digit number, often `00`.	`00`

Table 47. Managed (J2C) connection factory properties for Adapter for SAP Software (continued)

Property name in enterprise service discovery	Property name in administrative console	Type	Description	Default value
User name	userName	String	The user account for the adapter on the SAP system. If bidirectional language support is enabled, this property is affected by the following properties, which are set using the enterprise service discovery wizard: • **User Name BiDi Format**: Controls the bi-di format for this property. • **Skip BiDi Transformation for User Name**: Controls invocation of bi-di transformation for this property.	
User Name BiDi Format	biDiContext_UserNameEIS	String	Specifies the bi-di format for the Username property.	ILYNN

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.

J2C activation specification properties (also referred to as message endpoint properties) correspond to the ActivationSpec interface of the J2EE Connector Architecture Specification. These properties are relevant to working with the ALE interface because an activation specification is used during endpoint activation. Endpoint activation is the process of notifying the adapter of eligible event listeners. For inbound processing, the adapter uses these event listeners to receive events from SAP before forwarding them to the endpoint (a message driven bean).

When you configure the adapter, specify the activation specification properties listed below.

Table 48. Activation specification properties

Property name in enterprise service discovery	Property name in administrative console	Type	Description	Default value
Ale Failure Code	aleFailureCode	String	Specifies the status code for dispatch failure. You must specify a value for this property (68 or 58) to cause the adapter to update the SAP failure status code after the adapter has retrieved an IDoc object for event processing. SAP converts this value to 40.	None

Table 48. Activation specification properties (continued)

Property name in enterprise service discovery	Property name in administrative console	Type	Description	Default value
Ale Failure Text	aleFailureText	String	Specifies the descriptive text for dispatch failure. Specifying a value for this property is optional, even when you set AleUpdateStatus to `true`.	None
Ale Selective Update	aleSelectiveUpdate	String	Specifies which IDoc Type and MessageType combinations are to be updated when the adapter is configured to update a standard SAP status code. You can define values for this property only if AleUpdateStatus has been set to `true`. The syntax for this property is: `IDocType: MessageType [;IDocType: MessageType [;...]]` where a slash (/) delimiter separates each IDoc Type and MessageType, and a semicolon (;) delimiter separates entries in a set. The following example illustrates two sets. In the example, MATMAS03 and DEBMAS03 are the IDocs, and MATMAS and DEBMAS are the message types: `MATMAS03/ MATMAS;DEBMAS03/DEBMAS`	
Ale Status Message Code	aleStatusMsgCode	String	If required, this property specifies the message code to use when the adapter posts the ALEAUD Message IDoc (ALEAUD01). Configure this message code in the receiving Partner Profile. You can set a value for this property only if AleUpdateStatus has been set to `true`.	None

Table 48. Activation specification properties (continued)

Property name in enterprise service discovery	Property name in administrative console	Type	Description	Default value
Ale Success Code	aleSuccessCode	String	Specifies the success status code for Application Document Posted. You must specify a value for this property (52 or 53) to cause the adapter to update the SAP success status code after the interface has retrieved an IDoc object for event processing. SAP converts this value to status 41 (Application Document Created in Receiving System).	None
Ale Success Text	aleSuccessText	String	Specifies the descriptive text for successful Application Document Posted. Specifying a value for this property is optional, even when you set AleUpdateStatus to `true`.	None
Ale Update Status	aleUpdateStatus	Boolean	Specifies whether an audit trail is required for all message types. Set this property to `true` if you want the adapter to update a standard SAP status code after the adapter has retrieved an IDoc object for event processing.	`False`
Application Server Host	applicationServerHost	String	When the adapter is configured to run without load balancing, this property specifies the IP address or the name of the application server that the adapter logs in to. Regardless of whether it is configured for load balancing, the adapter assumes that the name of the gateway host is the same as the value specified for this property.	None

Table 48. Activation specification properties (continued)

Property name in enterprise service discovery	Property name in administrative console	Type	Description	Default value
Assured Once only Delivery	assuredOnceDelivery	Boolean	If this property is set to `true`, the adapter attempts to store XID information in the event store. If it is set to `false`, the adapter does not attempt to store the information. A value of `true` provides assured once event delivery. A value of `false` does not provide assured once event delivery, but it provides better performance. This property is used only if the MDB is transactional. If it is not, no transaction can be used, regardless of the configuration.	`True`
Auto Create Event Table	EP_CreateTable	Boolean	Flag that indicates whether the adapter should create the event recovery table automatically if it does not already exist.	`True`
Client	client	String	SAP Client number under which the adapter logs in. This value is often `100`.	`100`
Codepage Number	codepage	Integer	Establishes a connection to the appropriate language. This property must correspond to the value specified in the Language property. For example, if Language is set to `JA` (Japanese), then Codepage must be set to `8000`, as dictated by the SAP application. Refer to SAP documentation for the exact Language and Codepage values.	
Database Schema Name	EP_SchemaName	String	Schema used for automatically creating the event recovery table.	None
DataSource Name BiDi Special Format	biDiContext_EDTDatasource BiDiSpecialFormat	String	Controls the bidirectional format specific for all EP properties.	

Table 48. Activation specification properties (continued)

Property name in enterprise service discovery	Property name in administrative console	Type	Description	Default value
EIS BiDi Format	biDiContextEIS	String	The bi-di format used by SAP for its business data.	
EIS BiDi Special Format	biDiContextSpecialFormat	String	Specifies the category of values that are subject to special treatment during invocation of bi-di transformation to ensure accurate transformation of the category. Categories are predefined (for example: FTP, URLs, and e-mail addresses).	
EIS Connection Retry Interval	retryInterval	Integer	Specifies the time delay before trying to restart the ALE event listeners. The value should be specified in milliseconds.	
EIS Connection Retry Limit	retryLimit	Integer	Specifies the number of times the adapter tries to restart the ALE event listeners. A value of `0` indicates an infinite number of retries.	`0`
Event Persistence BiDi Format	biDiContext_EDTTablenameEIS	String	Specifies the bidirectional format for the Event Persistence property.	`ILYNN`
Event Recovery DataSource (JNDI) Name	EP_DataSource_JNDIName	String	JNDI name of the data source configured for event recovery.	None
Event Recovery Table Name	EP_TableName	String	Name of the event recovery table.	
Gateway Host	gatewayHost	String	SAP gateway host where the gateway service is running.	None
Gateway Service	gatewayService	String	The gateway server identifier. This value is often `sapgw00`. `00` is the system number of the server running the SAP gateway (which is usually an application server). This value cannot be `00` if you have more than one server.	`sapgw00`

Table 48. Activation specification properties (continued)

Property name in enterprise service discovery	Property name in administrative console	Type	Description	Default value
Logon Group	logonGroup	String	When the adapter is configured for load balancing, this property specifies the name of the logon group that represents a group of application servers.	None
Ignore IDoc Packet Errors	ignoreIDocPacketErrors	Boolean	If the adapter encounters an error while processing the IDoc packet, it can behave in two different ways. When this property is set to `false`, the adapter stops processing further IDocs in that packet and reports an error to the SAP system. When this property is set to `true`, the adapter logs an error and continues processing the rest of the IDocs in that packet. This property is not used for single IDocs and for non-split IDoc packets.	
Language	language	String	Language in which the adapter logs in.	`E` for English
Message Server Host	messageServerHost	String	When the adapter is configured for load balancing, this property specifies the name of the message server.	None
Metadata BiDi format	biDiContextMetadata	String	Specifies the bidirectional format of meta configuration data.	
Number Of Listeners	numberofListeners	Integer	Specifies the number of event listeners that are to be started.	1
Partner Charset	partnerCharset	String	Specifies PartnerCharset encoding. When an encoding is specified, it is used; otherwise the default encoding is used.	None
Password	password	String	Password for the adapter user account on the SAP system.	

Table 48. Activation specification properties (continued)

Property name in enterprise service discovery	Property name in administrative console	Type	Description	Default value
Password BiDi Format	biDiContext_PasswordEIS	String	Specifies the bidirectional format for the Password value.	`ILYNN`
Password to connect to Event Datasource	EP_Password	String	User password for connecting to the database.	None
RFC Program ID	rfcProgramID	String	Program identifier under which the RFC server program registers.	None
RFC Trace On	RFCTraceOn	Boolean	Specifies whether to generate a text file detailing the RFC activity for each event listener. You can specify a value of `true` or `false`. A value of `true` activates tracing, which generates a text file. Use these text files in a development environment only, because the files can grow rapidly.	`False`
SAP System ID	SAPSystemID	String	When the adapter is configured for load balancing, this property specifies the logical name of the SAP system (also known as R3*name*).	None
Skip BiDi Transformation	biDiContextSkip	String	Controls invocation of bi-di transformation. Possible values are `true` or `false`. A blank value invokes the lookup mechanism.	
Skip BiDi Transformation for DataSource Name	biDiContext_EDTDatasourceEIS	Boolean	Controls invocation of support for bidirectional text transformation for the Password property.	`True`
Skip BiDi Transformation for Event Persistence	biDiContext_EDTTablenameSkip	Boolean	Controls invocation of support for bidirectional text transformation for the Event Persistence property.	`True`
Skip BiDi Transformation for Password	biDiContext_PasswordSkip	Boolean	Controls invocation of support for bidirectional text transformation for the Password property.	`True`
Skip BiDi Transformation for User Name	biDiContext_UserNameSkip	Boolean	Controls invocation of support for bidirectional text transformation for the User name property.	`True`

Table 48. Activation specification properties (continued)

Property name in enterprise service discovery	Property name in administrative console	Type	Description	Default value
System Number	systemNumber	String	System number of the application server. The value, which is a two-digit number, is often 00.	00
User Name	userName	String	Name of the adapter user account on the SAP system.	None
User Name BiDi Format	biDiContext_UserNameEIS	String	Specifies the bidirectional format for the User name property.	ILYNN
Username to connect to Event Datasource	EP_UserName	String	User name for connecting to the database.	None

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.

Resource adapter properties

The following resource adapter properties can be set to control bidirectional transformation.

- 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
- EIS Bidi Special Format
- Metadata BiDi Format
- Password BiDi Format
- Skip BiDi Transformation
- Skip BiDi Transformation for Password
- Skip BiDi Transformation for User name
- User name BiDi Format

Activation specification properties

The following activation specification properties can be set to control bidirectional transformation.

- Datasource Name BiDi Special Format
- EIS BiDi Format
- EIS BiDi Special Format
- Event Persistence BiDi Format
- Metadata BiDi Format
- Password BiDi Format
- Skip BiDi Transformation
- Skip BiDi Transformation for Datasource name
- Skip BiDi Transformation for Event Persistence
- 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.

Before you begin

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.

How to perform this task

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 SAP Software.

Additional information you might need

Table 49. 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**.

Chapter 14. Glossary

The glossary of terms for IBM WebSphere Adapters is included in the WebSphere Adapters, version 6.0.2 information center.

You can view it at the following link: WebSphere Adapters glossary.

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

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Printed in USA