

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

*IBM WebSphere Adapter for SAP
Software User Guide
Version 7 Release 5 Fix Pack 3
(7.5.0.3)*



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

*IBM WebSphere Adapter for SAP
Software User Guide
Version 7 Release 5 Fix Pack 3
(7.5.0.3)*



Note

Before using this information and the product it supports, read the information in "Notices" on page 431.

November 2012

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

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

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

© Copyright IBM Corporation 2006, 2012.

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

Contents

Chapter 1. Overview of WebSphere

Adapter for SAP Software 1

Hardware and software requirements	2
Technical overview of WebSphere Adapter for SAP Software	3
The external service wizard	8
Business objects	8
Authentication using connection specification properties	10
Log and Trace Analyzer	10

Chapter 2. Planning for adapter implementation 13

Before you begin	13
Security	13
Support for protecting sensitive user data in log and trace files	13
User authentication	14
Deployment options	15
WebSphere Adapters in clustered environments	19
Migrating to version 7.5.0.3 of WebSphere Adapter for SAP Software	21
Migration considerations	21
Performing the migration	23
Upgrading but not migrating a project	25
Migrating WebSphere Business Integration applications	26
Migrating applications from WebSphere InterChange Server	26
Migration considerations for WebSphere Business Integration adapters	28
Migrating application artifacts from WebSphere InterChange Server	28
Migrating adapter-specific artifacts	29
Changes to the import, export, and WSDL files after migration	32

Chapter 3. SAP interfaces 35

The BAPI interfaces	35
Outbound processing for the BAPI interface	36
Inbound processing for the BAPI interface	38
Business object structure for a simple BAPI	42
Business object structure for a nested BAPI	44
Coordinating BAPI transactions and passing values between BAPI invocations	45
The BAPI work unit interface	46
Outbound processing for the BAPI work unit interface	46
Business object structure for a BAPI work unit	47
The BAPI result set interface	48
Outbound processing for the BAPI result set interface	49
Business object structure for a BAPI result set	49
The ALE interfaces	51
Outbound processing for the ALE interfaces	52

Inbound processing for the ALE interfaces	53
ALE business object structure	60
The ALE pass-through IDoc interface	66
Outbound processing for the ALE pass-through IDoc interface	67
Inbound processing for the ALE pass-through IDoc interface	69
ALE pass-through IDoc business object structure	74
The Advanced event processing interface	76
Outbound processing for the Advanced event processing interface	77
Inbound processing for the Advanced event processing interface	80
Business objects for the Advanced event processing interface	84

Chapter 4. Samples and tutorials 87

Chapter 5. Configuring the module for deployment. 89

Roadmap for configuring the module	89
Performing prerequisite tasks specific to an interface	91
Configuring the SAP system to work with the adapter	91
Creating the data source	94
Creating an IDoc definition file	95
Adding transport files to the SAP server	95
Implementing event-detection mechanisms	96
Creating an authentication alias	104
Creating the project	105
Adding external software dependencies	106
Setting connection properties for the external service wizard	107
Configuring the module for outbound processing	110
Authentication using connection specification properties	110
Passing the connection parameters dynamically	111
Configuring a module for the BAPI interface	112
Configuring a module for the BAPI work unit interface	122
Configuring a module for the BAPI result set interface	131
Configuring a module for ALE outbound processing	140
Configuring a module for ALE pass-through IDoc outbound processing	153
Configuring a module for Query interface for SAP Software processing	159
Configuring a module for Advanced event processing - outbound	170
Configuring the module for inbound processing	178
Configuring a module for BAPI inbound processing	179
Configuring a module for ALE inbound processing	189

Configuring a module for ALE pass-through IDoc inbound processing	206
Configuring a module for Advanced event processing - inbound	214

Chapter 6. Changing interaction specification properties 223

Chapter 7. Modifying artifacts 225

Modifying service import for BAPI outbound processing	225
Modifying service export for BAPI inbound processing	226
Modifying service import for ALE outbound processing	227
Modifying service export for ALE inbound processing	228
Modifying service import for Query interface for SAP software outbound processing	229
Modifying service import for Advanced event processing outbound	230
Modifying service export for Advanced event processing inbound	231

Chapter 8. Deploying the module 233

Deployment environments	233
Deploying the module for testing.	233
Adding external software dependencies	233
Generating and wiring a target component for testing inbound processing	235
Adding the module to the server	236
Testing the module for outbound processing using the test client	237
Deploying the module for production	238
Adding external software dependencies on the server	238
Installing the RAR file (for modules using stand-alone adapters only)	239
Exporting the module as an EAR file	240
Installing the EAR file	241
Deploying the module in a clustered environment	243
Deploying module embedded in the application	243
Deploying module at node level with embedded activation specification	245
Deploying module at node level with JNDI activation specification	247

Chapter 9. Administering the adapter module 251

Configuring logging and tracing	251
Configuring logging properties	251
Changing the log and trace file names	253
Changing configuration properties for embedded adapters	254
Setting resource adapter properties for embedded adapters	254
Setting managed (J2C) connection factory properties for embedded adapters	256
Setting activation specification properties for embedded adapters	258

Changing configuration properties for stand-alone adapters	260
Setting resource adapter properties for stand-alone adapters	260
Setting managed (J2C) connection factory properties for stand-alone adapters	261
Setting activation specification properties for stand-alone adapters	262
Starting the application that uses the adapter.	264
Stopping the application that uses the adapter	264
Managing Advanced event processing	265
Displaying the current events queue.	265
Displaying the future events queue	266
Maintaining the archive table	267
Managing the adapter log file	269
Monitoring SAP gateway connections	271
Monitoring performance using Performance Monitoring Infrastructure	271
Configuring Performance Monitoring Infrastructure	272
Viewing performance statistics	274
Enabling tracing with the Common Event Infrastructure	275
Adding dependency libraries to the deployed resource adapter	276
Stand-alone deployment.	276
EAR deployment	276
Using enhanced EAR editor	277
Using administrative console of the WebSphere Application Server	277

Chapter 10. Troubleshooting and support 279

Techniques for troubleshooting problems	279
Log and Trace Analyzer	281
Detecting errors during outbound processing.	281
Resolving errors during Query interface for SAP Software processing	283
SAP dependencies when using the WebSphere Adapter for SAP Software with the Advanced Event Processing (AEP) interface	287
Resolving memory-related issues	289
Supported codepages for WebSphere Adapter for SAP Software	289
First-failure data capture (FFDC) support	289
Avoiding stale connection problems in the SAP adapter	290
Resolving selector exception error	291
Resolving a service 'sapxxnn' unknown error.	292
Resolving SAP JCo environment setup errors.	292
Adapter returns version conflict exception message	293
Unreleased IDoc segments cause exceptions	293
Invalid meta data exception	294
Disabling end point applications of the passive adapter	294
Frequently Asked Questions	295
Support	296
Searching knowledge bases (Web search)	296
Getting Fixes	297
Self-help resources.	297

Chapter 11. Reference information	299
Business object information	299
Application-specific information	299
Supported data operations	311
Naming conventions	315
Outbound configuration properties	321
Connection properties for the wizard	322
Resource adapter properties	332
Managed connection factory properties	334
Interaction specification properties	348
Inbound configuration properties	352
Connection properties for the wizard	354
Resource adapter properties	364
Activation specification properties for BAPI inbound processing	366
Activation specification properties for ALE inbound processing	383

Activation specification properties for Advanced event processing	405
Globalization	424
Globalization and bidirectional transformation	424
Properties enabled for bidirectional data transformation	426
Fault business objects	426
Adapter messages	428
Related information	428
Video samples	428

Notices	431
Programming interface information	433
Trademarks and service marks	433

Index	435
------------------------	------------

Chapter 1. Overview of WebSphere Adapter for SAP Software

With WebSphere® Adapter for SAP Software, you can create integrated processes that include the exchange of information with an SAP server, without special coding.

Using the adapter, an application component (the program or piece of code that performs a specific business function) can send requests to the SAP server (for example, to query a customer record in an SAP table or to update an order document) or receive events from the server (for example, to be notified that a customer record has been updated). The adapter creates a standard interface to the applications and data on the SAP server, so that the application component does not have to understand the lower-level details (the implementation of the application or the data structures) on the SAP server.

WebSphere Adapter for SAP Software complies with the Java™ Platform, Enterprise Edition (JEE) Connector Architecture (JCA) version 1.5. JCA 1.5 standardizes the way application components, application servers, and enterprise information systems, such as an SAP server, interact with each other. WebSphere Adapter for SAP Software makes it possible for JCA-compliant application servers to connect to and interact with the SAP server. Application components running on the JCA-compliant server can then communicate with the SAP server in a standard way (using business objects or JavaBeans).

The following example assumes you are setting up an adapter using IBM® Integration Designer and deploying the module that includes the adapter to IBM Business Process Manager or WebSphere Enterprise Service Bus.

Suppose a company uses SAP Software to coordinate most of its business operations. SAP includes a business function that returns a list of customers in response to a range of customer IDs. An application component might be able to use this function as part of an overall business process. For example, the promotions department within the company sends advertising material to customers, and, as part of that process, needs to first obtain a list of customers.

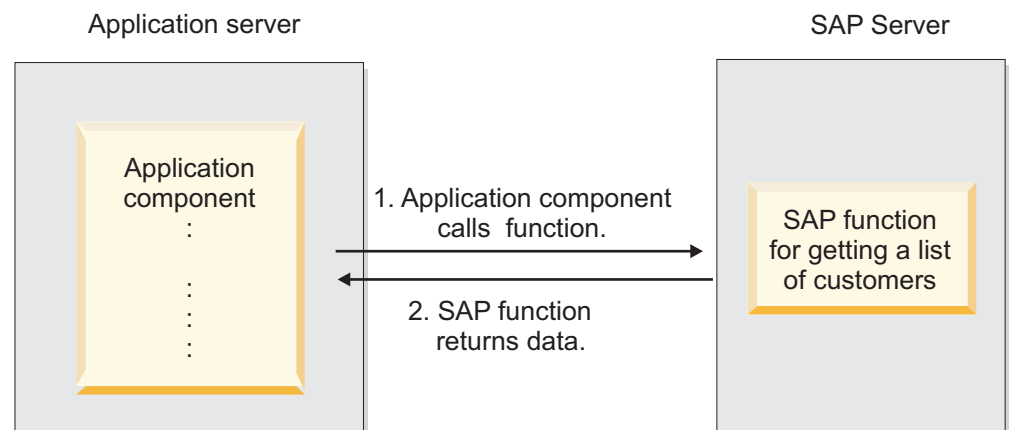


Figure 1. An application component calls an SAP function to obtain data

The SAP function does not have a web service interface. Therefore, the application component used by the promotions department needs to understand the low-level

API and the data structures of the SAP function to call the function. Information technology resources and time would be needed to create the linkage between the application component and the SAP function.

With WebSphere Adapter for SAP Software, you can automatically generate an interface to the SAP function to hide the lower-level details of the function. Depending on how you want to use the adapter, you can embed it with the deployed module, or install the adapter as a stand-alone component, to be used by more than one application. The adapter is deployed to IBM Business Process Manager or WebSphere Enterprise Service Bus. The application component interacts with the adapter instead of with the SAP function.

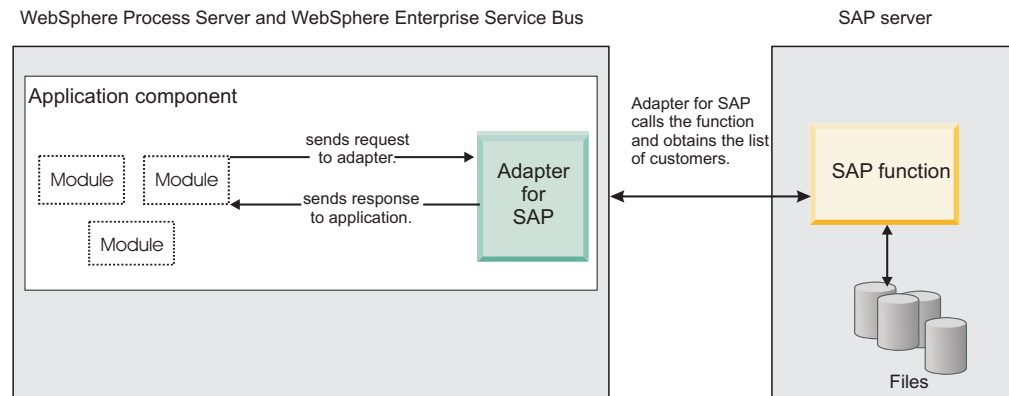


Figure 2. An application component calls the SAP adapter, and the SAP adapter interacts with the SAP function to obtain the data

The adapter, which you generate with the external service wizard of IBM Integration Designer, uses a standard interface and standard data objects. The adapter takes the standard data object sent by the application component and calls the SAP function. The adapter then returns a standard data object to the application component. The application component does not have to deal directly with the SAP function; it is the SAP adapter that calls the function and returns the results.

For example, the application component that needed the list of customers would send a standard business object with the range of customer IDs to the SAP adapter. The application component would receive, in return, the results (the list of customers) in the form of a standard business object. The application component does not need to know how the function worked or how the data was structured. The adapter would perform all the interactions with the actual SAP function.

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

Hardware and software requirements

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

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

Additional information

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

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

Technical overview of WebSphere Adapter for SAP Software

WebSphere Adapter for SAP Software provides multiple ways to interact with applications and data on SAP servers. Outbound processing (from an application to the adapter to the SAP server) and inbound processing (from the SAP server to the adapter to an application) are supported.

For outbound processing, the adapter client invokes the adapter operation to create, update, or delete data on the SAP server or to retrieve data from the SAP server.

For inbound processing, an event that occurs on the SAP server is sent from the SAP server to the adapter. The ALE inbound and BAPI inbound interfaces start event listeners that detect the events. Conversely, the Advanced event processing interface polls the SAP server for events. The adapter then delivers the event to an endpoint, which is an application or other consumer of the event from the SAP server.

You configure the adapter to perform outbound and inbound processing by using the external service wizard to create a deployable module that includes the interface to the SAP application as well as business objects based on the functions or tables it discovers on the SAP server.

Overview of the outbound processing interfaces

WebSphere Adapter for SAP Software provides multiple interfaces to the SAP server for outbound processing.

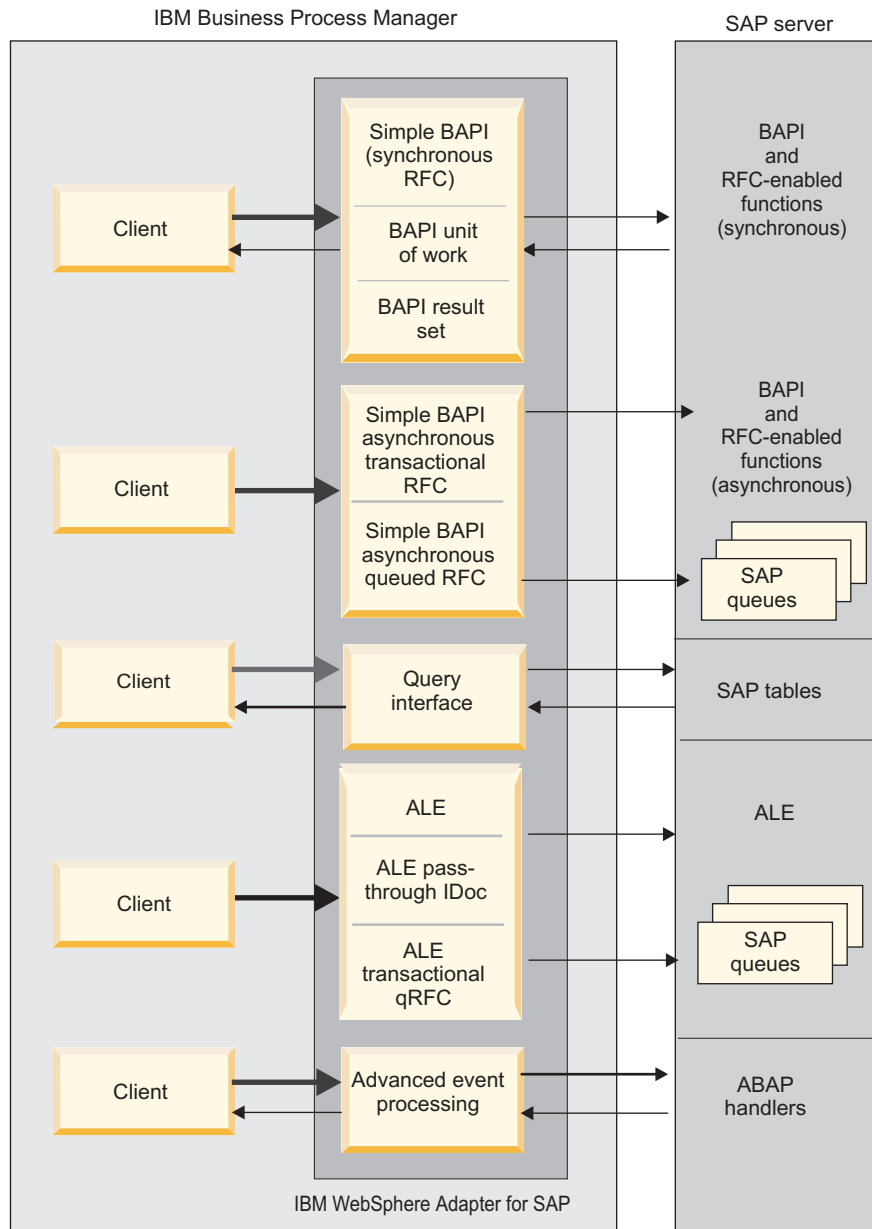


Figure 3. Outbound interfaces

- Through its BAPI interfaces, the adapter issues remote function calls (RFCs) to RFC-enabled functions, such as a Business Application Programming Interface (BAPI) function. These remote function calls create, update, or retrieve data on an SAP server .
 - The BAPI interface works with individual BAPIs (simple BAPIs). For example, you might want to check to see whether specific customer information exists in an SAP database.
 - The BAPI work unit interface works with ordered sets of BAPIs. For example, you might want to update an employee record. To do so, you use three BAPIs to lock the record (to prevent any other changes to the record), update the record, and have the record approved.
 - The BAPI result set interface uses two BAPIs to select multiple rows of data from an SAP database.

BAPI calls are useful when you need to perform data retrieval or manipulation, and a BAPI or RFC function that performs the task already exists.

Simple BAPIs can be invoked through the Synchronous RFC, Asynchronous Transactional RFC, or Asynchronous Queued RFC protocol.

- With Synchronous RFC, both the adapter and the SAP server must be available when the call is made from the adapter to the SAP server. The adapter sends a request to the SAP server and waits for a response.
- With Asynchronous Transactional RFC, a transaction ID is associated with the call from the adapter to the SAP server. The adapter does not wait for a response from the SAP server. Only the transaction ID is returned to the client application.
- With Asynchronous Queued RFC, the call from the adapter is delivered to a predefined queue on the SAP server. As with Asynchronous Transactional RFC, a transaction ID is associated with the call, and the adapter does not wait for a response from the SAP server.

This interface is useful when the event sequence must be preserved.

- The Query interface for SAP Software retrieves data from specific SAP application tables. It can return the data or check for the existence of the data. You can use this type of interaction with SAP if you need to retrieve data from an SAP table without using an RFC function or a BAPI.
- With the Application Link Enabling (ALE) interface, you exchange data using SAP Intermediate Data structures (IDocs). For outbound processing, you send an IDoc or a packet of IDocs to the SAP server.

The ALE interface, which is particularly useful for batch processing of IDocs, provides asynchronous exchange. You can use the Queued Transactional (qRFC) protocol to send the IDocs to a queue on the SAP server. The qRFC protocol ensures the order in which the IDocs are received. It is often used for system replications or system-to-system transfers.

- With the ALE pass-through IDoc interface, the adapter sends the IDoc to the SAP server with no conversion of the IDoc. The business object contains stream data representing the IDoc.
- With the Advanced event processing interface, you send data to the SAP server. The data is then processed by an ABAP handler on the SAP server.

Overview of the inbound processing interfaces

WebSphere Adapter for SAP Software provides the following interfaces to the SAP server for inbound processing.

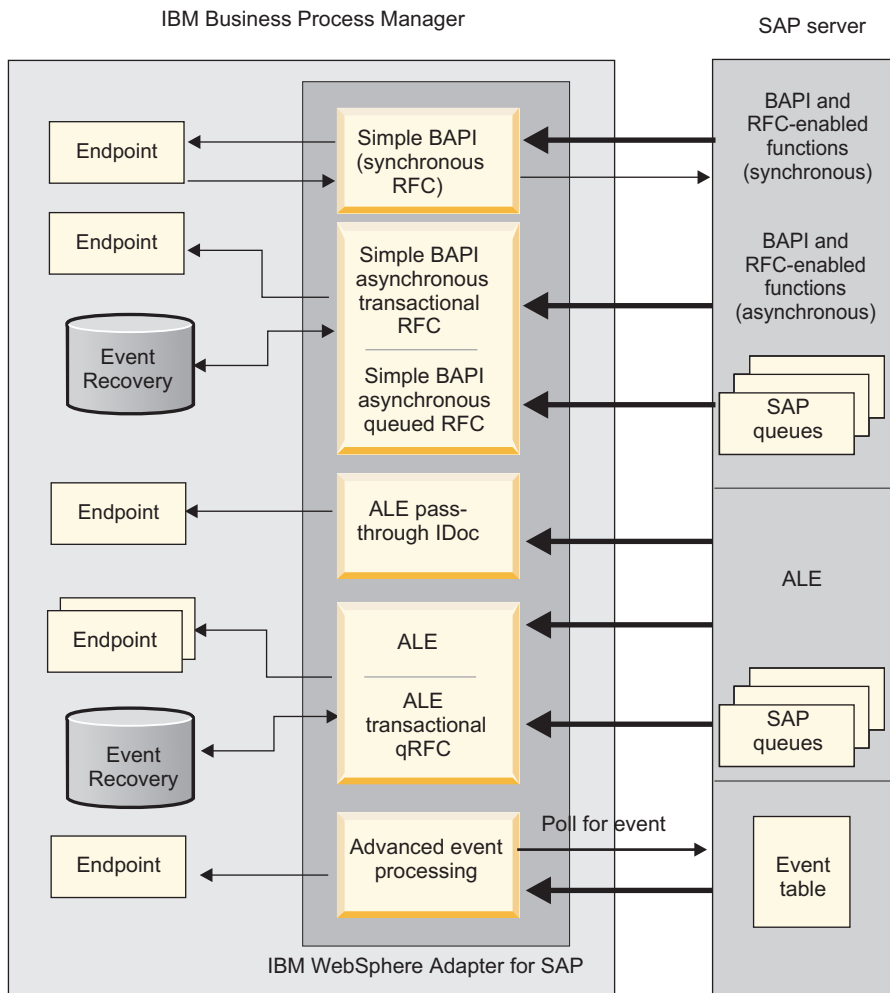


Figure 4. Inbound interfaces

- Through its BAPI inbound interface, the adapter listens for events and receives notifications of RFC-enabled function calls from the SAP server.
 - With Synchronous RFC, both the adapter and the SAP server must be available when the call is made from the SAP server to the adapter. The adapter sends the request to a predefined application and returns the response to the SAP server.

Note: In version 6.1.0 of the WebSphere Adapter for SAP Software, inbound synchronous processing of RFC-enabled functions was known as the *Synchronous callback interface*.

- With Asynchronous Transactional RFC, the event will be delivered to the adapter even if the adapter is not available when the call is made. The SAP server stores the event on a list of functions to be invoked and continues to attempt to deliver it until the adapter is available.

Note: You can also use Asynchronous Transactional RFC if you want to deliver the functions from a predefined queue on the SAP server. Delivering the files from a queue ensures the order in which the functions are sent.

If you select assured-once delivery, the adapter uses a data source to persist the event data received from the SAP server. Event recovery is provided to track and recover events in case a problem occurs when the adapter attempts to deliver the event to the endpoint.

- With the ALE inbound processing interface, the adapter listens for events and receives one or more IDocs from the SAP server. As with ALE outbound processing, ALE inbound processing provides asynchronous exchange.

You can use the qRFC interface to receive the IDocs from a queue on the SAP server, which ensures the order in which the IDocs are received.

If you select assured-once delivery, the adapter uses a data source to persist the event data, and event recovery is provided to track and recover events in case a problem occurs when the adapter attempts to deliver the event to the endpoint.

- With the ALE pass-through IDoc interface, the SAP server sends the IDoc through the adapter to the endpoint with no conversion of the IDoc. The business object contains stream data representing the IDoc.
- The Advanced event processing interface polls the SAP server for events. It discovers events waiting to be processed. It then processes the events and sends them to the endpoint.

How the adapter interacts with the SAP server

The adapter uses the SAP Java Connector (SAP JCo) API to communicate with SAP applications. An application sends a request to the adapter, which uses the SAP JCo API to convert the request into a BAPI function call. The SAP system processes the request and sends the results to the adapter. The adapter sends the results in a response message to the calling application.

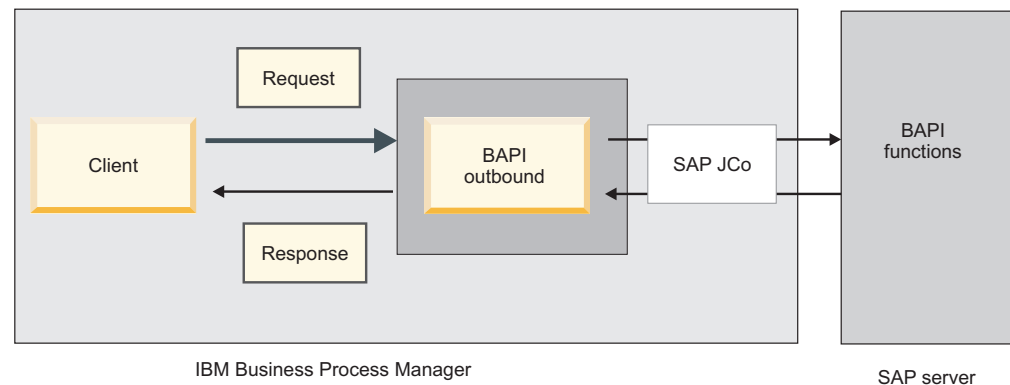


Figure 5. How the adapter connects a calling application with an SAP application

How the adapter is packaged

WebSphere Adapter for SAP Software is packaged and delivered as two RAR files, and the one you use depends on whether the invoked SAP function supports transactional behavior.

- If the targeted function (for example, BAPI) supports transactions, use the CWYAP_SAPAdapter_Tx.rar adapter because it supports local transaction behavior and, as such, can participate in the transaction managed by the WebSphere Application Server Transaction Manager.
- If the targeted function (for example, BAPI) does not support transactions, use the CWYAP_SAPAdapter.rar adapter because it indicates to the WebSphere

Application Server Transaction Manager that the interaction performed with the SAP system cannot participate in and follow transactional semantics.

The external service wizard

The external service wizard is a tool you use to create services. The external service wizard establishes a connection to the SAP server, discovers services (based on search criteria you provide), and generates business objects, interfaces, and import or export files, based on the services discovered.

Using IBM Integration Designer, you establish a connection to the SAP server to browse the metadata repository on the SAP server. The SAP metadata repository, which is a database of the SAP data, provides a consistent and reliable means of access to that data.

The result of running the external service wizard is a module that contains the interfaces and business objects along with the adapter. You can deploy this module on IBM Business Process Manager or WebSphere Enterprise Service Bus.

The external service wizard also produces an import file (for outbound processing) or an export file (for inbound processing).

- The import file contains the managed connection factory property settings you provided in the wizard.
- The export file contains the activation specification property settings you provided in the wizard.

Business objects

The business object is a structure or a container to exchange data between application components and the adapter. The data can represent either a business entity, such as an invoice or an employee record, or unstructured text.

For outbound processing, the application component uses business objects to send data to SAP or to obtain data (through the adapter) from SAP. In other words, the application component sends a business object to the adapter and the adapter converts the data in the business object to a format that is compatible with an SAP API call. The adapter then invokes the SAP API with this data.

For inbound processing, the SAP server sends a function call through the adapter to an endpoint. The adapter converts the function call into a business object.

The adapter uses the metadata that is generated by the external service wizard to construct a business-object definition. This metadata contains information such as the operation of the business object and import or export parameters.

How data is represented in business objects

The way data is represented in a business object depends on the interface to SAP that you are using.

For example, a BAPI business-object definition, which is generated by the external service wizard, is modeled on the BAPI function interface in SAP. The business-object definition represents a BAPI function.

SapBapiCustomerGetlist	
ControlIndicatorReadOneTimeCustomersOnly	string
MaximumNumberOfCustomers	int
SapReturn	SapReturn
SapAddressdata	SapAddressdata []
SapIdrange	SapIdrange []
SapSpecialdata	SapSpecialdata []

Figure 6. A sample business object

For the ALE interface, the business-object definition is based on standard or extension IDocs available on the SAP server.

For the Query Interface for SAP Software, the data in the business object represents the columns of the associated table in SAP.

For the advanced event processing interface, business objects are based on custom IDocs, standard IDocs, or extension IDocs available on the SAP server.

How business-object definitions are created

You create business-object definitions by using the external service wizard, launched from IBM Integration Designer. The wizard connects to the application, discovers data structures in the application, and generates business-object definitions to represent them. It also generates other artifacts needed by the adapter, such as the interface information that indicates the input and output parameters.

During adapter configuration, you can optionally choose to generate a business graph. In version 6.0.2, each top-level business object is contained in a business graph, which includes a verb that an application can use to specify additional information about the operation to be performed. Beginning version 7.0, business graphs are optional; they are required only when you are adding business objects to a module created with an earlier version. If business graphs exist, they are processed, but the verb is ignored.

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

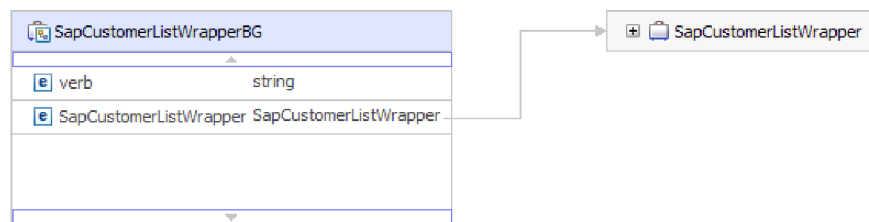


Figure 7. Example of a BAPI business graph

Authentication using connection specification properties

WebSphere Adapter for SAP Software uses connection properties either through Managed Connection Factory properties or a Java Authentication and Authorization Services (JAAS) alias. If you want to change the connection properties used for authentication with either one of these authentication methods, you can change the connection properties through the IBM Business Process Manager administrative console and restart the Java EE application or change the JAAS security settings.

In addition to the methods explained, the connection parameters can also be specified through the ConnectionSpec properties. The ConnectionSpec properties are used by an application component to pass connection-related properties.

You can specify the relevant ConnectionSpec properties for the outbound request. If you specify both ConnectionSpec properties and Managed Connection Factory properties during run time, the adapter uses the values specified in the ConnectionSpec properties to create a connection and ignores the values in the Managed Connection Factory properties.

The following table lists the ConnectionSpec properties for various authentication mechanisms:

Table 1. ConnectionSpec properties

Authentication mechanism	Properties
Basic	User Name, Password
Single Sign On (SSO)	ssoLogonTicket
X509 Certificate	x509cert

The following table lists the properties available to enable Secure Network Connection (SNC):

Table 2. Secure Network connection properties

Authentication mechanism	Properties
Secure Network Connection (SNC)	sncMyname, sncPartnername, sncQop, sncMode, sncLib

Refer to “Enable Secure Network Connection” on page 344, for more information.

Note: The SSO token / X509 authentication mechanism can also be accomplished by setting the username to "\$X509CERT\$" (for X509 certificate based authentication) and "\$MYSAPSSO2\$" (for SSO token based authentication) and setting the password to a base 64 encoded string, which comprises of a SSO token / X509 certificate.

To configure the adapter to create an SAP server connection, see “Passing the connection parameters dynamically” on page 111.

Log and Trace Analyzer

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

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

The adapter's component ID for filtering and highlighting is a string composed of the characters SAPRA plus the value of the adapter ID property. For example, if the adapter ID property is set to 001, the component ID is SAPRA001.

If you run multiple instances of the same adapter, ensure that the first eight characters of the adapter ID property are unique for each instance so that you can correlate the log and trace information to a particular adapter instance. By making the first seven characters of an adapter ID property unique, the component ID for multiple instances of that adapter is also unique, allowing you to correlate the log and trace information to a particular instance of an adapter. For example, when you set the adapter ID property of two instances of WebSphere Adapter for SAP Software to 001 and 002. The component IDs for those instances, SAPRA001 and SAPRA002, are short enough to remain unique, enabling you to distinguish them as separate adapter instances. However, instances with longer adapter ID properties cannot be distinguished from each other. If you set the adapter ID properties of two instances to Instance01 and Instance02, you will not be able to examine the log and trace information for each adapter instance because the component ID for both instances is truncated to SAPRAInstance.

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

For more information, see the “Adapter ID (AdapterID)” on page 333 property.

Chapter 2. Planning for adapter implementation

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

Before you begin

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

To configure and use WebSphere Adapter for SAP Software, you should understand and have experience with the following concepts, tools, and tasks:

- The business requirements of the solution you are building.
- Business integration concepts and models, including the Service Component Architecture (SCA) programming model.
- The capabilities provided by the integration development tools you will use to build the solution. You should know how to use these tools to create modules, test components, and complete other integration tasks.
- The capabilities and requirements of the runtime environment you will use for the integration solution. You should know how to configure and administer the host server and how to use the Process Administrative Console to set and modify property definitions, configure connections, and manage events.
- The SAP server environment in which you are working. This includes a detailed understanding of the SAP GUI, RFC-enabled functions (such as BAPIs), and ALE IDocs.

Security

The adapter uses the J2C authentication data entry, or the authentication alias feature of Java EE security to provide secure user name and password authentication. For more information about security features, see the documentation for IBM Business Process Manager or WebSphere Enterprise Service Bus. The adapter also supports secure network connections for both outbound and inbound processing.

Support for protecting sensitive user data in log and trace files

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

Log and trace files for the adapter can contain data from your SAP server, which might contain sensitive or confidential information. Sometimes these files might be seen by individuals without authorization to view sensitive data. For example, a support specialist must use the log and trace files to troubleshoot a problem.

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

See “Managed connection factory properties” on page 334 for information about this optional property.

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

- The contents of a business object
- The contents of the object key of the event record
- User name, Password, Environment, and Role
- The URL used to connect to the SAP server
- Business object data in an intermediate form, such as the fields in a BAPI

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

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

User authentication

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

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

- When the external service wizard connects to SAP server to extract, or *discover*, information about the objects and services that you can access with the adapter.
- At run time on IBM Business Process Manager or WebSphere Enterprise Service Bus, when the adapter connects to SAP server to process outbound requests and inbound events.

Authentication in the wizard

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

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

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

Authentication at run time

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

- Adapter properties
- J2C authentication alias

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

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

Deployment options

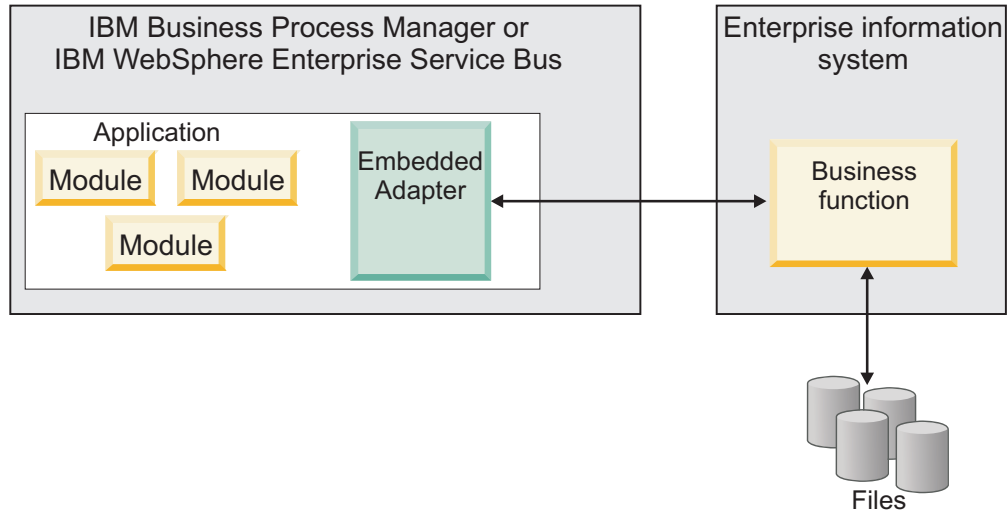
There are two ways to deploy the adapter. You can either embed it as part of the deployed application, or you can deploy it as a stand-alone RAR file. The requirements of your environment affect the type of deployment option you choose.

The following are the deployment options:

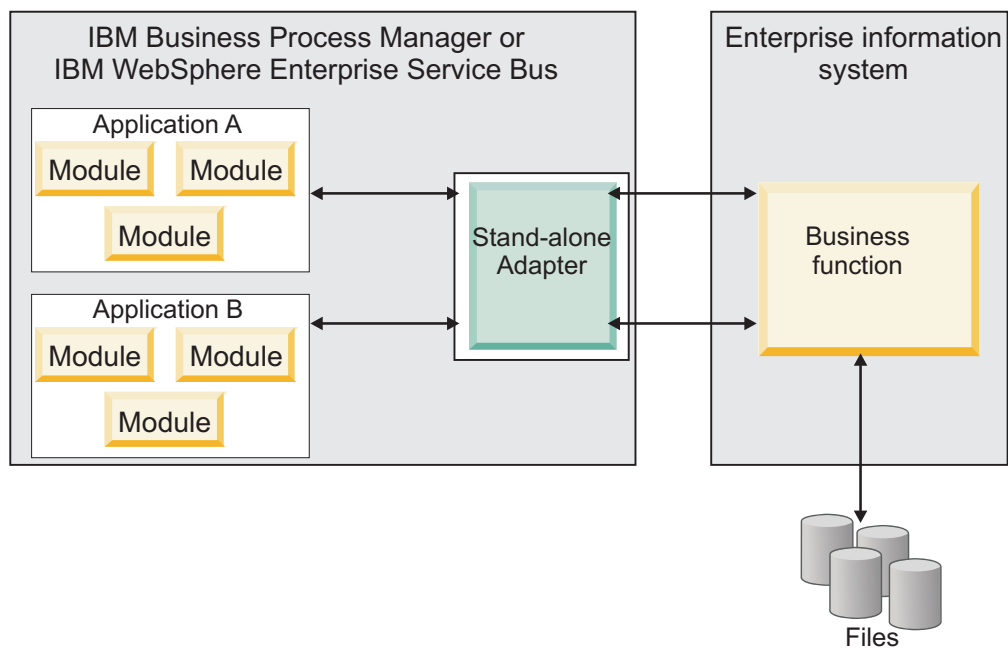
- **With module for use by single application:** With the adapter files embedded in the module, you can deploy the module to any application server. Use an embedded adapter when you have a single module using the adapter or if multiple modules need to run different versions of the adapter. Using an embedded adapter enables you to upgrade the adapter in a single module without the risk of destabilizing other modules by changing their adapter version.
- **On server for use by multiple applications:** If you do not include the adapter files in a module, you must install them as a stand-alone adapter on each application server where you want to run the module. Use a stand-alone adapter when multiple modules can use the same version of the adapter and you want

to administer the adapter in a central location. A stand-alone adapter can also reduce the resources required by running a single adapter instance for multiple modules.

An embedded adapter is bundled within an enterprise archive (EAR) file and is available only to the application with which it is packaged and deployed.



A stand-alone adapter is represented by a stand-alone resource adapter archive (RAR) file, and when deployed, it is available to all deployed applications in the server instance.



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

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

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

Considerations for embedding an adapter in the application

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

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

Considerations for using a stand-alone adapter

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

- Stand-alone adapters have no class loader isolation.
Because stand-alone adapters have no class loader isolation, only one version of any defined Java artifact is run and the version and sequence of that artifact is undetermined. For example, when you use a stand-alone adapter there is only *one* resource adapter version, *one* adapter foundation class (AFC) version, or *one* third-party JAR version. All adapters deployed as stand-alone adapters share a single AFC version, and all instances of a defined adapter share the code version. All adapter instances using a given third-party library must share that library.
- If you update any of these shared artifacts, all applications using the artifacts are affected.
For instance, if you have an adapter that is working with server version X, and you update the version of the client application to version Y, your original application might stop working.
- Adapter Foundation Classes (AFC) is compatible with previous versions, but the latest AFC version must be in every RAR file that is deployed in a stand-alone manner.
If more than one copy of any JAR file is in the class path in a stand-alone adapter, the one that is used is random; therefore, they all must be the latest version.

Considerations while deploying adapters with different versions

When you install multiple adapters with different versions of CWYBS_AdapterFoundation.jar, and if a lower version of the CWYBS_AdapterFoundation.jar is loaded during run time, the adapter returns the

ResourceAdapterInternalException error message, due to a version conflict. For example, when you install Oracle E-Business Suite adapter version 7.0.0.3 and WebSphere Adapter for SAP Software version 7.5.0.3, the following error message is displayed "The version of CWYBS_AdapterFoundation.jar is not compatible with IBM WebSphere Adapter for SAP Software" as IBM WebSphere Adapter for SAP Software loads file:/C:/IBM/WebSphere/ProcServer7/profiles/ProcSrv01/installedConnectors/CWYOE_OracleEBS.rar/CWYBS_AdapterFoundation.jar with version 7.0.0.3. However, the base level of this jar required is version 7.5.0.3.

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

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

Deploying a stand-alone Adapter

1. Install WebSphere Adapter for Flat Files version 7.0.1.0 through the IBM Business Process Manager administrative console.
2. Install WebSphere Adapter for SAP Software version 7.5.0.0 through the administrative console.
3. Create ActivationSpec for an ALE pass-through inbound operation.
4. Create an application in IBM Integration Designer for a stand-alone ALE pass-through inbound operation.
5. Install and start the application through the administrative console.
6. Verify the error.

Note: An error message is generated in the log/trace area of IBM Business Process Manager, to indicate an AFC version conflict.

Deploying an embedded Adapter

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

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

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

Deploying a combination of stand-alone and embedded Adapters

1. Install WebSphere Adapter for JDBC version 7.0.1.0 through the IBM Business Process Manager administrative console.
2. Create an ActivationSpec for a JDBC inbound operation.

3. Create an application in IBM Integration Designer for a JDBC inbound operation, for the stand-alone Adapter deployment.
4. Deploy the JDBC inbound application and trigger your inbound events.
5. Create an application in IBM Integration Designer for a WebSphere Adapter for SAP Software version 7.5.0.0 inbound embedded Adapter deployment.
6. Deploy an SAP inbound application, and trigger your inbound events.

Note: You can resolve the AFC version conflict by using different class loaders for the stand-alone and embedded deployments. With this approach, the migration process handles different CWYBS_AdapterFoundation.jar files and do not conflict with each other. You can start both JDBC and SAP inbound applications successfully, and process Inbound events without exception.

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

WebSphere Adapters in clustered environments

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

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

- IBM Business Process Manager or WebSphere Enterprise Service Bus
- WebSphere Application Server Network Deployment
- WebSphere Extended Deployment

To deploy and configure WebSphere Adapter for SAP Software in a clustered environment, see: “Deploying the module in a clustered environment” on page 243. When you set up a server cluster, you create a Deployment Manager profile. The HAManager, a subcomponent of the Deployment Manager, notifies the Java Platform, Enterprise Edition (JEE) Connector Architecture (JCA) container to activate an adapter instance. For information about creating clustered environments, see the following link: http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r5mx/index.jsp?topic=/com.ibm.wbpm.imuc.z.doc/topics/tins_zos_create_cluster.html.

Using WebSphere Extended Deployment, you can optionally enhance the performance of adapter instances in your clustered environment. WebSphere Extended Deployment extends the WebSphere Application Server Network Deployment capabilities by using a dynamic Workload Manager instance instead of a static Workload Manager. The dynamic Workload Manager instance can optimize the performance of adapter instances in the cluster by dynamically balancing the load of the requests. It means that application server instances can be automatically stopped and started based on the load variations, allowing systems with different capacities and configurations to handle load variations evenly. For information about the benefits of WebSphere Extended Deployment, see <http://publib.boulder.ibm.com/infocenter/wxdinfo/v6r1m1/index.jsp>.

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

High availability for inbound processes

Inbound processes are based on events triggered as a result of updates to data in the SAP server. WebSphere Adapter for SAP Software is configured to detect updates through event listeners or by polling an event table. The adapter then publishes the event to its endpoint.

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

Note: In the active-passive configuration mode of the adapters, the endpoint application of the passive adapter instance also listens to the events/messages even if the `enableHASupport` property is set to `True`. This is because the `alwaysactivateAllMDBs` property in the JMS activation specification is set to `True`. To stop the endpoint application of the passive adapter instance from listening to the events, you must set the `alwaysactivateAllMDBs` property value to `False`. For more information, see “Disabling end point applications of the passive adapter” on page 294.

If the value for the `enableHASupport` property is `false`, all adapter instances listen for events on the Inbound cluster. Any number of SAP adapters can be made Active on a HA cluster – all in the active mode. When more than one adapter instance actively polls in a cluster setup, it serves as a load balancer. If one of the cluster instances fails, the other active instances in the cluster handles the events. If an IDoc fails, when SAP resubmits it, the other active instances of the adapter handles the events.

Note: In clustered environments, when the adapter works in a Active-Active configuration, it provides both high availability and load balancing support. This functionality is useful in production environments where high performance is needed.

High availability for outbound processes

In clustered environments, multiple adapter instances are available to perform outbound process requests. So, if your environment has multiple applications that interact with WebSphere Adapter for SAP Software for outbound requests, then you might improve performance by deploying the module to a clustered environment. In a clustered environment, multiple outbound requests can be processed simultaneously, so that they are not attempting to process the same record.

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

request at a time. For more information about workload management, see the following link: http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/trun_wlm.html.

Migrating to version 7.5.0.3 of WebSphere Adapter for SAP Software

By migrating to version 7.5.0.3 of WebSphere Adapter for SAP Software, you automatically upgrade from the previous version of the adapter. Additionally, you can migrate your applications that embed an earlier version of the adapter, so that the applications can use features and capabilities present in version 7.5.0.3.

Migration considerations

WebSphere Adapter for SAP Software version 7.5.0.3 may have some features and updates that might affect your existing adapter applications. Before migrating applications that use WebSphere Adapter for SAP Software, you must consider some factors that might affect your existing applications.

Compatibility with earlier versions

When migrating to version 7.5.0.3 of WebSphere Adapter for SAP Software, always run the migration wizard after importing the old project into your IBM Integration Designer workspace.

This will modify the old artifacts and make them compatible with the WebSphere Adapter for SAP Software version 7.5.0.3. The migration tool works with artifacts that are generated using WebSphere Adapter for SAP Software version 6.1.x, version 6.2.x, version 7.0.x and version 7.5.0 and version 7.5.0.2

The migration wizard replaces (upgrades) version 6.1.x, version 6.2.x, version 7.0.x, version 7.5.0, or version 7.5.0.2 of the adapter with version 7.5.0.3 and enables version 7.5.0.3 features and functionality for use with your applications.

Note: The migration wizard does not create components or modify existing components, such as mappers and mediators to work with version 7.5.0.3 of the adapters. If any of your applications embed an adapter that is version 7.5.0.2 or earlier and you are upgrading to version 7.5.0.3, and you want your applications to take advantage of the features and functions in version 7.5.0.3, you might need to change those applications.

If the artifacts within a module have inconsistent versions, the entire module is marked as unavailable for migration and cannot be selected. Version inconsistencies are recorded in the workspace log, as they indicate that a project might be corrupted.

The adapter migration wizard in IBM Integration Designer Version 7.5.1 only supports the migration of adapters from version 6.1.x, version 6.2.x, version 7.0.x, version 7.5.0, and version 7.5.0.2 to version 7.5.0.3. It does not support the adapter migration from lower versions to any of the versions before version 7.5.0.3.

Deciding whether to upgrade or to upgrade and migrate

The default processing of the migration wizard is to perform an upgrade of the adapter and to migrate the application artifacts so that the applications can use

features and functions in version 7.5.0.3 of the adapter. When you choose to upgrade the adapter by selecting a project, the wizard automatically selects the associated artifacts for migration.

If you decide that you want to upgrade the adapter from version 6.1.x, version 6.2.x, version 7.0.x, version 7.5.0, or version 7.5.0.2 to version 7.5.0.3, but you do not want to migrate the adapter artifacts, you can do so by clearing the adapter artifacts from the appropriate area of the migration wizard.

Running the migration wizard without selecting any adapter artifacts installs and upgrades your adapter. As the artifacts are not migrated, your applications cannot take advantage of the features and capabilities that exist in version 7.5.0.3 of the adapter.

Migrating multiple adapters referred within a project

When a module contains one or more connector projects, each of which references to different adapters (for example, a module project that contains connector projects referring to JDBC and SAP adapters), the migration wizard identifies the artifacts belonging to each adapter and migrates these artifacts without disrupting the artifacts of other adapters.

When you select the module project and launch the migration wizard:

- The **Source connector** field lists the connector projects with the selected module project.
- The **Dependent artifact projects** area lists only the selected module project.

If you select the connector project and launch the migration wizard:

- The **Source connector** field lists only the selected connector project.
- The **Dependent artifact projects** area lists all projects which reference the selected connector project, including the module project.

Run the migration wizard in a test environment

Because adapter migration might require you to change those applications that use version 7.5.0.3 of WebSphere Adapter for SAP Software, you must always perform the migration in a development environment first and test your applications before deploying the application to a production environment.

The migration wizard is fully integrated with the development environment.

Deprecated features

If you currently have Version 6.1.x, version 6.2.x, version 7.0.x and Version 7.5.0 and version 7.5.0.2 of the adapter installed, review the deprecated features and note if there are any compatibility conflicts between the versions before upgrading the adapter.

A deprecated feature is one that is supported but no longer recommended and that might become obsolete. The following features from earlier versions of WebSphere Adapter for SAP Software have been deprecated in version 7.5.0.2 and might require changes to your applications:

- The **IgnoreBAPIReturn** property is no longer a Managed Connection Factory property. It is now set as part of the interaction spec.

- The DataDelimiter property has been removed from the application-specific information for Query interface for SAP Software business objects.

Performing the migration

You can migrate a project or EAR file to version 7.5.0.3 using the adapter migration wizard. When the tool is finished, the migration is complete and you can work in the project or deploy the module.

Before you begin

Review the information in *Migration considerations*.

About this task

To perform the migration in IBM Integration Designer, complete the following steps.

Note: After migration is complete, the module is no longer compatible with previous versions of the IBM Business Process Manager or WebSphere Enterprise Service Bus runtimes or IBM Integration Designer.

The following steps describe how to run the adapter migration wizard from the connector project menu while in the Java EE perspective in IBM Integration Designer.

Procedure

1. Import the PI (project interchange) file for an existing project into the workspace.

Note: Ensure that you do not modify the contents of the RAR or copy the adapter JAR file outside the connector project.

2. When projects are created in an earlier version of IBM Integration Designer, the Workspace Migration wizard starts automatically and selects the projects to migrate. Follow the wizard and complete the workspace migration. For more information, see <http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r5m1/topic/com.ibm.wbpm.wid.imuc.doc/topics/tmigrscart.html>.
3. Change to the Java EE perspective.
4. Right-click the module and select **Migrate connector project**. For example, the adapter RAR module.

You can also launch the adapter migration wizard in the following ways:

- Right-click the project in the Java EE perspective and select **Migrate adapter artifacts**.
 - From the Problems view, right-click a migration-specific message and select **Quick Fix** to correct the problem.
5. In the Select Projects window, perform the following steps:
 - a. The **Source connector** field displays the name of the connector project that you are migrating. If you are migrating a module project, this field lists all the connector projects in the module project. Select the source project from the list. For more information, see “Migrating multiple adapters referred within a project” on page 22.

- b. The **Target connector** field displays the name of the connector to which you are migrating. If you are working with more than one adapter version, this list displays the names of all the compatible connectors. Select the connector you want to migrate.
 - c. The **Target version** field displays the version corresponding to the target connector that you selected in the previous step.
 - d. The **Dependent artifacts project** area lists the adapter artifacts that are migrated. If you are migrating a module project, this area lists only the selected module project. If you are migrating a connector project within the module project, this area lists all projects which reference the selected connector project, including the module project. By default, all the dependent artifact projects are selected. If you do not select a dependent artifact project, that project is not migrated. You can migrate any project that you have not selected at a later time. Previously migrated projects, projects with a current version, and projects that contain errors are unavailable for migration and are not selected. For more information, see “Upgrading but not migrating a project” on page 25.
 - e. Click **Next**. A warning window is displayed with the message, “Properties that are not supported in this version of the target adapter will be removed during the migration”.
 - f. Click **OK**.
6. In the Review Changes window, review the migration changes that occur in each of the artifacts that you are migrating. To view the details, expand each node by clicking the + sign.
 7. To complete the migration:
 - Click **Finish**.
 - If the files that need to be updated during migration are in read-only mode, you will be unable to click on the **Finish** button. To view these files, click **Next**. The Update Read-only files window displays the read-only files. To update these files and continue with the migration, click **Finish**. To exit the wizard without migrating the adapter, click **Cancel**.

Before running the migration process, the wizard performs a backup of all projects affected by the migration. The projects are backed up to a temporary folder within the workspace. If the migration fails for any reason, or if you decide to cancel the migration before it completes, the wizard deletes the modified projects and replaces them with the projects stored in the temporary folder.

Upon completing the migration successfully, all backed up projects are deleted.

8. After completing the migration, the following steps have to be completed for the SAP adapter:
 - a. Right click **Connection Project** (CWYAP_SAPAdapter_Tx or CWYAP_SAPAdapter)
 - b. Click **Properties**
 - c. In the left pane, click **Java Build Path**
 - d. Open the **Libraries** tab in the right pane
 - e. Select the **JCo 218 jar** if it is present and click **Remove**
 - f. Click **Add External JARs**
 - g. Browse and select the JCo3 API jar
 - h. Click **Open**
 - i. Click **OK** on the Properties window

9. If you are migrating an EAR file, optionally create a new EAR file with the migrated adapter and artifacts, and deploy it to IBM Business Process Manager or WebSphere Enterprise Service Bus. For more information about exporting and deploying an EAR file, see the topics devoted to it in this documentation.

Results

The project or EAR file is migrated to version 7.5.0.3. You do not need to run the external service wizard after exiting the adapter migration wizard.

Upgrading but not migrating a project

You can upgrade the adapter from an earlier version, to version 7.5.0.3 while choosing not to migrate the adapter project artifacts.

About this task

Running the migration wizard without selecting any adapter artifacts installs and upgrades your adapter. As the artifacts are not migrated, your applications cannot take advantage of the features and capabilities that exist in version 7.5.0.3 of the adapter.

Procedure

1. Import the PI (project interchange) file into the workspace.
2. When projects are created in an earlier version of IBM Integration Designer, the Workspace Migration wizard starts automatically and selects the projects to migrate. Follow the wizard and complete the workspace migration. For more information, see <http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r5m1/topic/com.ibm.wbpm.wid.imuc.doc/topics/tmigrscart.html>.
3. In the Java EE perspective, right-click the project name and click **Migrate connector project**. The **Adapter Migration** wizard is displayed.
4. In the Select Projects window, clear the dependent artifact projects, and click **Next**. A warning window is displayed with the message, "The properties that are not supported in the version of the target adapter will be removed during the migration."
5. Click **OK**.
6. In the Review Changes window, review the migration changes that occur during updating the project. To view the details, expand each node by clicking the + sign.
7. To complete the migration:
 - Click **Finish**.
 - If the files that need to be updated during migration are in read-only mode, you will be unable to click on the **Finish** button. To view these files, click **Next**. The Update Read-only files window displays the read-only files. To update these files and continue with the migration, click **Finish**. To exit the wizard without migrating the adapter, click **Cancel**.
8. After completing the migration, the following steps have to be completed for the SAP adapter:
 - a. Right click **Connection Project** (CWYAP_SAPAdapter_Tx or CWYAP_SAPAdapter)
 - b. Click **Properties**
 - c. In the left pane, click **Java Build Path**
 - d. Open the **Libraries** tab in the right pane

- e. Select the **JCo 218 jar** and click **Remove**
- f. Click **Add External JARs**
- g. Browse and select the JCo3 API jar
- h. Click **Open**
- i. Click **OK** on the Properties window

Results

The project can now be used with WebSphere Adapter for SAP Software, version 7.5.0.3.

Migrating WebSphere Business Integration applications

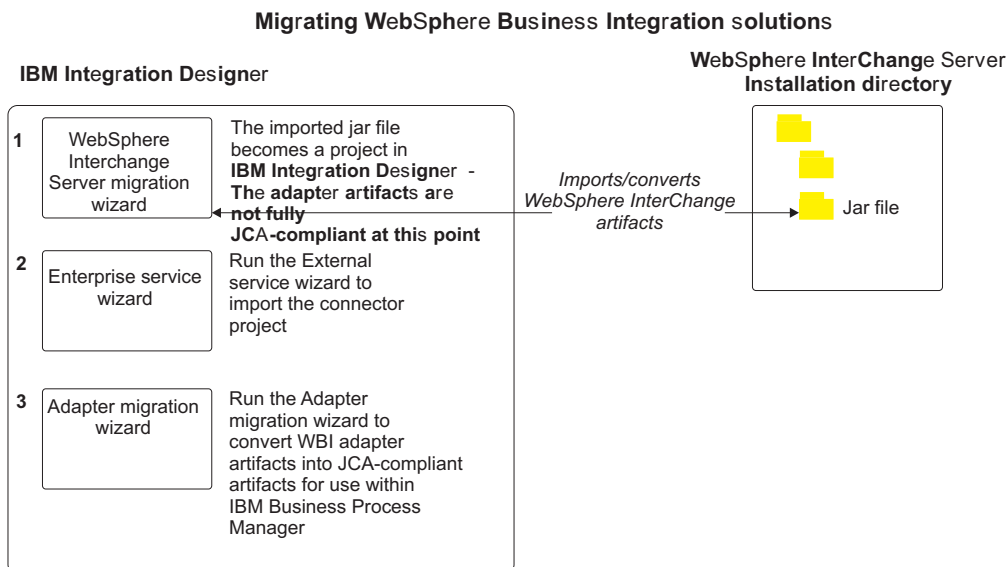
You need to migrate the WebSphere Business Integration applications so that they become compatible with Version 7.5.0.3 of your adapter.

About this task

Migrating WebSphere Business Integration applications for use with Version 7.5.0.3 of your WebSphere adapter is a multistep process. First, the artifacts from WebSphere InterChange Server are migrated and converted. A project is then created for the artifacts in IBM Integration Designer. In the remaining steps, the adapter-specific artifacts are migrated and converted into the JCA-compliant format supported by Version 7.5.0.3 of the adapter.

Example

The following diagram shows the wizards that you use to migrate WebSphere Business Integration solutions from WebSphere InterChange Server, so that these applications can be used with Version 7.5.0.3 of your adapter.



Migrating applications from WebSphere InterChange Server

To use Version 7.5.0.3 of WebSphere Adapter for SAP Software with applications from WebSphere InterChange Server, you need to migrate the application artifacts and convert them so that they can be deployed and run on IBM Business Process

Manager or WebSphere Enterprise Service Bus. Understanding this task at a high level helps you perform the steps that are needed to accomplish the task.

The following figure illustrates the flow of the migration task. The steps that follow the figure describe this task at a high level only. See the topics following this roadmap for the details on how to perform each of these steps.

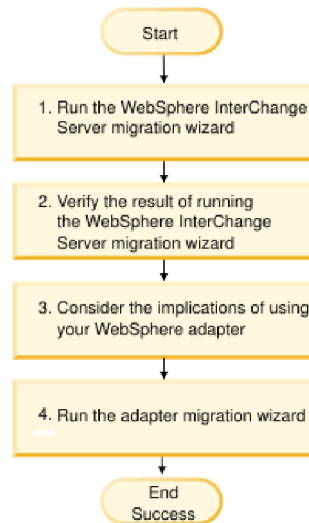


Figure 8. Roadmap for migrating applications from WebSphere InterChange Server

Migrating applications from WebSphere InterChange Server

This task consists of the following steps:

1. Run the WebSphere InterChange Server migration wizard.
The WebSphere InterChange Server migration wizard moves the application artifacts into IBM Integration Designer. The migrated adapter artifacts are not fully JCA-compliant at the completion of this task.
2. Verify that the WebSphere InterChange Server migration is successful.
Review all messages from the Migration results window and take action if required.
3. Consider the implications of using Version 7.5.0.3 of WebSphere Adapter for SAP Software.
In addition to considerations for migrating WebSphere InterChange Server applications, you need to consider how Version 7.5.0.3 of WebSphere Adapter for SAP Software works with the migrated applications. Some of the adapter operations supported by WebSphere InterChange Server applications might be supported and implemented differently with Version 7.5.0.3 of the adapter.
4. Run the adapter migration wizard.
Run the adapter migration wizard to update adapter-specific artifacts such as the schemas and service definition files (.import, .export, and .wsdl files) for use with Version 7.5.0.3 of the adapter.

Migration considerations for WebSphere Business Integration adapters

By migrating to WebSphere Adapter for SAP Software Version 7.5.0.3, you have an adapter that is compliant with the Java Platform, Enterprise Edition (JEE) Connector Architecture (JCA) and designed specifically for service-oriented architecture.

Supported interfaces

The following are the interfaces supported for migrating from WBI SAP Adapter to JCA SAP Adapter.

- Simple BAPI
- BAPI Transaction
- ALE
- HDR
- ABAP Extension

Application artifacts

Before running the adapter migration wizard, use the WebSphere InterChange Server migration wizard to generate the application artifacts for the WebSphere Business Integration adapter, including the business objects, maps, and collaborations. Then you can run the adapter migration wizard to update the adapter-specific artifacts such as the schemas and service definition files (.import, .export, and .wsdl) so that they are suitably converted into a format that is compliant with JCA.

Run the migration wizard in a test environment first

Because migrating from a WebSphere Business Integration adapter to WebSphere Adapter for SAP Software might require changes to the applications that use Version 7.5.0.3 of WebSphere Adapter for SAP Software, always perform the migration in a development environment first and test your applications before deploying the application to a production environment.

Migrating application artifacts from WebSphere InterChange Server

To migrate the application artifacts into IBM Integration Designer, run the WebSphere InterChange Server migration wizard. The wizard imports and converts most of the artifacts into a format that is compatible with IBM Business Process Manager or WebSphere Enterprise Service Bus.

Before you begin

Launch the WebSphere InterChange Server migration wizard from within IBM Integration Designer to migrate the application artifacts from WebSphere InterChange Server format into artifacts that are compatible with IBM Business Process Manager or WebSphere Enterprise Service Bus.

For information about how to prepare to migrate artifacts from WebSphere InterChange Server and for detailed instructions on performing the migration and verifying that the migration was successful, see <http://publib.boulder.ibm.com/>

infocenter/dmndhelp/v7r5m1/index.jsp?topic=/com.ibm.wbpm.wid.imuc.doc/topics/twics.html.

About this task

Running WebSphere InterChange Server migration wizard might not fully convert adapter-specific artifacts (such as service descriptors, service definitions, and business objects) into IBM Business Process Manager or WebSphere Enterprise Service Bus compatible artifacts. To complete the migration of adapter-specific artifacts, run the adapter migration wizard after you have successfully run the WebSphere InterChange Server migration wizard.

Note: While you run the WebSphere InterChange Server migration wizard, ensure that you set each connector in the repository to the same adapter version.

Results

The project and application artifacts are migrated and converted into IBM Business Process Manager compatible artifacts.

What to do next

Run the adapter migration wizard to migrate the adapter-specific artifacts.

Migrating adapter-specific artifacts

After a project is created for the artifacts in IBM Integration Designer, you can migrate the project using the adapter migration wizard. The adapter migration wizard updates adapter-specific artifacts such as the schemas and service definition files (.import, .export, and .wsdl) for use with version 7.5.0.3 of the adapter. When you finish running the adapter migration wizard, the migration is complete and you can work in the project or deploy the module.

Before you begin

Before running the adapter migration wizard, do the following steps:

- Review the information in “Migration considerations” on page 21.
- Run the WebSphere InterChange Server migration wizard to migrate the project and convert data objects for use with IBM Business Process Manager or WebSphere Enterprise Service Bus.

About this task

After migration is complete, the module will work only with Version 7.5.0.3 of your adapter.

To perform the migration in IBM Integration Designer, complete the following steps.

Procedure

1. Import the PI (project interchange) file for an existing project into the workspace.
2. When projects are created in an earlier version of IBM Integration Designer, the Workspace Migration wizard starts automatically and selects the projects to migrate. Follow the wizard and complete the workspace migration. For more

information, see <http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r5m1/topic/com.ibm.wbpm.wid.imuc.doc/topics/tmigrscart.html>.

3. Change to the Java EE perspective.
4. Right-click the connector project and select **Migrate connector project**.
You can also launch the adapter migration wizard by using the right-click option and selecting the module project in the Java EE perspective and selecting **Migrate adapter artifacts**.

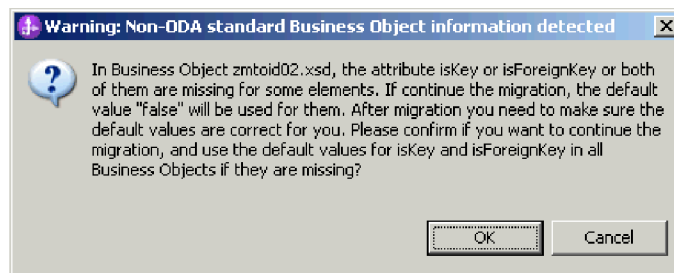
Note:

If the adapter type (for example, CICS/IMS adapter) is not supported by the migration wizard, the **Migrate connector project** and **Migrate adapter artifacts** menus are not available for selection. If the adapter project is of the latest version and the module projects referencing this adapter project are also of the latest version, these menus are disabled.

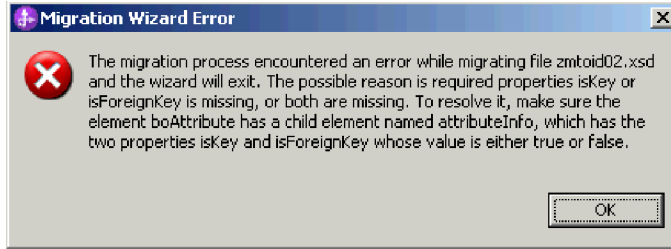
When you launch the migration wizard from the connector project while in the Java EE perspective, by default all the dependent artifact projects are selected. If you do not select a dependent artifact project, that project is not migrated.

5. In the Select Projects window, perform the following steps:
 - a. The **Source connector** field displays the name of the connector project that you are migrating. Select the source project from the list.
 - b. The **Target connector** field displays the name of the connector to which you are migrating. If you are working with more than one adapter version, this list displays the names of all the compatible connectors. Select the connector to which you want to migrate.
 - c. The **Target version** field displays the version corresponding to the target connector you selected in the previous step.
 - d. The **Dependent artifacts project** area lists the adapter artifacts that are migrated.
 - e. Review the tasks and warnings presented on the welcome page, and click **Next**. A warning window is displayed with the message, "The properties that are not supported in the version of the target adapter are removed during the migration."

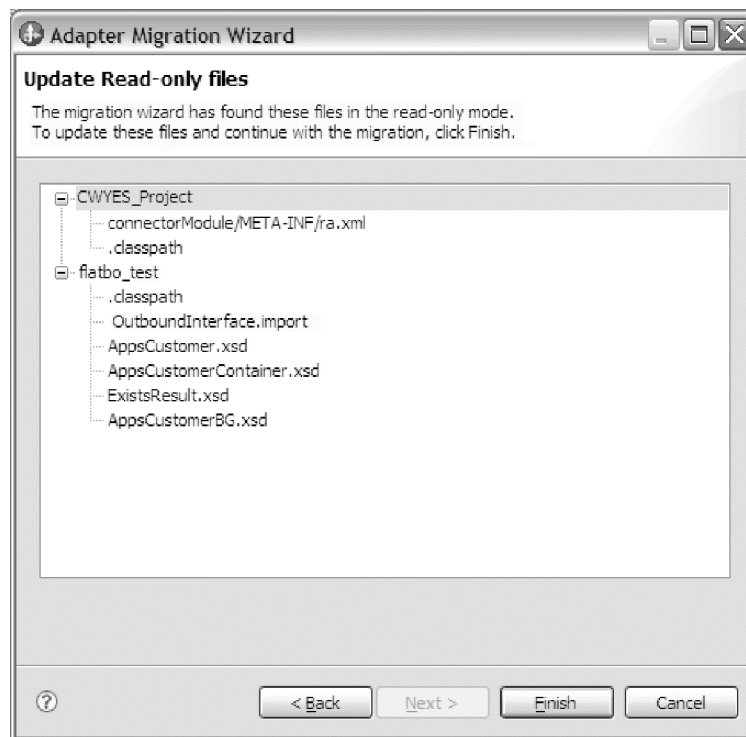
Note: If the business object does not contain the attribute `isKey`, `isForeignKey`, or both, the wizard displays a warning message.



- Click **OK**, to continue with the migration and to apply the default value for the attributes. The migration wizard uses the default value "false" for `isKey` and `isForeignKey` in all the business objects, if they are missing. After migration, you have to manually verify to make sure that the default value is appropriate and working.
- Click **Cancel**, to stop the migration process. The migration wizard displays an error message. Correct the problem manually and run the migration wizard again.



- f. Click **OK**.
6. In the Review Changes window, review the migration changes that occur in each of the artifacts that you are migrating. To view the details, expand each node by clicking the + sign.
7. To complete the migration:
 - Click **Finish**.
 - If the files that need to be updated during migration are in read-only mode, you will be unable to click on the **Finish** button. To view these files, click **Next**. The Update Read-only files window displays the read-only files. To update these files and continue with the migration, click **Finish**. To exit the wizard without migrating the adapter, click **Cancel**.



Before performing the migration process, the wizard backs up all projects affected by the migration. The projects are backed up to a temporary folder within the workspace. If the migration fails for any reason, or if you decide to cancel the migration before it completes, the wizard deletes the modified projects and replaces them with the projects stored in the temporary folder.

8. Select **Project > Clean**, to refresh and rebuild the workspace for the changes to take effect.

Note: The gatewayHost property in WebSphere Adapter for SAP Software does not have an equivalent property in WebSphere Business Integration Adapter for SAP. You must specify the value for the gatewayHost property manually after running the migration wizard.

9. If you are migrating an EAR file, create a new EAR file with the migrated adapter and artifacts, and deploy it to IBM Business Process Manager or WebSphere Enterprise Service Bus. For information about exporting and deploying an EAR file, see “Deploying the module for production” on page 238.

Results

The project is migrated to Version 7.5.0.3. You do not need to run the external service wizard after exiting the adapter migration wizard.

Changes to the import, export, and WSDL files after migration

When the WebSphere InterChange Server migration wizard moves the application artifacts into IBM Integration Designer, changes made are reflected in the service definition files: the import, export, and WSDL files.

The migrated adapter artifacts are not fully JCA-compliant at the completion of this task. You can complete the migration of the adapter-specific artifacts (such as service descriptors, service definitions, and business objects) to a JCA compatible format by running the adapter migration wizard.

Changes to the import file

During migration, the affected module artifacts are migrated to an import file. The existing JMS Binding property is changed to the EIS Binding property in the import file. The other property details added in the import file include information about the data binding configuration, changes to the connection information in the Managed Connection Factory properties, and several new method bindings.

Changes to the export file

During migration, the affected module artifacts are migrated to an export file. The existing JMS Binding property is changed to the EIS Binding property in the export file. The other property details added in the export file include information about the data binding configuration, changes to the connection information in the Activation Specification properties, and several new method bindings.

Changes to the WSDL file after migration

During migration, the affected module artifacts are migrated to corresponding WSDL files that include SAP specific service description WSDL artifacts. The service description files become JCA compatible. The WSDL files have an input and output type for each operation. Both the inbound and outbound operations work on their specific input types to produce corresponding output types after the operations are performed.

Note:

- After you migrate the adapter artifacts, you need to manually set the language property in the export and import files to required values as per your environment. Otherwise, the adapter might fail to establish a connection to the SAP server.

- When you migrate multiple top level inbound business objects in the project, only the first top-level business object inbound feature works correctly. For the other top level inbound business object to work correctly, you must manually modify the "emit + [verb name] + after image + [business object name]" method in the Input_Processing.java and Input_Async_Processing.java class to call the correct destination services.
- The WebSphere Business Integration Adapter for SAP properties that are either not valid or not supported by WebSphere Adapter for SAP Software are removed from the migrated artifacts.

Chapter 3. SAP interfaces

The SAP interfaces enable business process integration and asynchronous data communication between two or more SAP systems or between SAP and external systems.

The following interfaces exist for SAP software:

- BAPI interfaces
- BAPI work unit interface
- BAPI result set interface
- ALE interface
- ALE pass-through IDOC interface
- Query interface
- Advanced event processing interface

The BAPI interfaces

WebSphere Adapter for SAP Software supports outbound processing and inbound processing for simple BAPIs. In outbound processing, the client applications call BAPIs and other RFC-enabled functions on the SAP server. In inbound processing, the SAP server sends an RFC-enabled function (such as a BAPI function) through the adapter to an endpoint.

Suppose you want to build a service that creates a new customer on the SAP server. You first run the external service wizard to discover the BAPI_CUSTOMER_CREATEFROMDATA function. The wizard then generates the business object definition for BAPI_CUSTOMER_CREATEFROMDATA along with other SCA service artifacts. During BAPI outbound processing, the adapter receives the service request and converts the data into a BAPI invocation.

A simple BAPI performs 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.

Simple BAPIs can be used for outbound or inbound processing. You can specify synchronous RFC processing or asynchronous transactional RFC (tRFC) processing when you configure a module for a simple BAPI. In addition, for outbound processing, you can specify asynchronous queued RFC (qRFC) processing, in which BAPIs are delivered to a predefined queue on the SAP server.

- In synchronous RFC processing, the SAP server and the adapter must be available during processing.
 - In outbound processing, the client application sends a request and then waits for a response from the SAP server.
 - In inbound processing, the SAP server sends a request through the adapter to an endpoint and waits for a response from the adapter.
- In asynchronous tRFC outbound processing, the adapter associates a transaction ID with the function call to the SAP server. The adapter does not wait for a response from the SAP server. If the delivery is unsuccessful, the client application can use the TID to make the request again.

- In asynchronous tRFC inbound processing, the adapter does not have to be available when the SAP server invokes the function call. The function call is placed on a list of functions to be invoked, and the call is attempted until it is successful.
To send function calls from a user-defined outbound queue on the SAP server, you also specify asynchronous tRFC inbound processing.
- In asynchronous qRFC outbound processing, the process is similar to asynchronous tRFC outbound processing. A TID is associated with the function call, and the adapter does not wait for a response from the SAP server. Additionally, the BAPIs are delivered to a predefined queue on the SAP server. By sending BAPIs to the predefined queue, you can ensure the order in which they are delivered.

The adapter supports the following data types in the import, export, and changing parameter list:

1. Elementary types
2. Complex data type
 - a. Table types
 - 1) Line types
 - a) Structure (example - BAPIRET2)
 - b) Table type (example - TRTEXTS)
 - c) Data Element (example - TRACKTEXT)
 - d) Views (example - T001W_BIW)
 - 2) Predefined types (example - BIC_ADD_DATA_TT)
 - b. Structure

The adapter supports the following data types in the table parameter list -

1. Tables with flat line structure (example - BAPIRET2)

Note: If you select the **Generate business objects in SAP XI standard** option in the Specify the Discovery Properties window, the table type element will have a wrapper anonymous complex type, whose name is appended with a suffix “_Item”.

Outbound processing for the BAPI interface

In BAPI outbound processing, a client application sends a request to the SAP server. For simple BAPIs, you can request that processing be handled synchronously or asynchronously (the client application does not wait for a response from the SAP server).

During configuration you can make a selection about the type of remote RFC call you want to make.

Synchronous RFC

If you select **Synchronous RFC** (the default) during configuration for a simple BAPI, the following processing steps occur:

1. The adapter receives a request from a client application in the form of a BAPI business object.
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 process the BAPI or RFC function call in the SAP application.
4. After passing the data to the SAP server, the adapter handles the response from SAP and converts it back into the business object format required by the client application.
5. The adapter then sends the response back to the client application.

Asynchronous transactional RFC

If you select **Asynchronous transactional RFC** during configuration, the following processing steps occur:

1. The adapter receives a request from a client application in the form of a BAPI business object.
2. The adapter checks the business object to see whether the SAPTransactionID attribute has a value assigned.
 - If the SAPTransactionID has a value, the adapter uses that value during processing.
 - If the attribute does not have a value, the adapter makes a call to the SAP server and gets a transaction ID from the SAP server.
3. The adapter converts the BAPI business object to an SAP JCo function call.
4. The adapter uses the transactional Remote Function Call (tRFC) protocol to make the call to the SAP server .

The adapter does not wait for a response from the SAP server.

5. After the function data is passed to the SAP application, control returns to the adapter.
 - If the call to the SAP server fails, the SAP server throws an ABAPException.
 - If the call to the SAP server succeeds but contains invalid data, no exception is returned to the adapter. For example, if the adapter sends a request that contains an invalid customer number, the adapter does not respond with an exception indicating that no such customer exists.
6. The adapter passes the TID information to the client.

Asynchronous queued RFC

If you select **Asynchronous queued RFC** during configuration, the following processing steps occur:

1. The adapter receives a request from a client application in the form of a BAPI business object.
2. The adapter checks the business object to see whether the SAPTransactionID attribute has a value assigned.
 - If the SAPTransactionID has a value, the adapter uses that value during processing.
 - If the attribute does not have a value, the adapter makes a call to the SAP server and gets a transaction ID from the SAP server.
3. The adapter converts the BAPI business object to an SAP JCo function call.
4. The adapter uses the tRFC protocol to make the call to the specified queue on the SAP server .

The adapter does not wait for a response from the SAP server.

5. After the function data is passed to the SAP application, control returns to the adapter.
 - If the call to the SAP server fails, the SAP server throws an ABAPException.
 - If the call to the SAP server succeeds but contains invalid data, no exception is returned to the adapter. For example, if the adapter sends a request that contains an invalid customer number, the adapter does not respond with an exception indicating that no such customer exists.

6. The adapter passes the TID information to the client.

Wait on BAPI Commit

When using the BAPI interface to create data on SAP, the data will not be committed to the SAP Database until the BAPI_TRANSACTION_COMMIT BAPI is invoked explicitly. When creating a large amount of data using a BAPI, if you invoke the BAPI_TRANSACTION_COMMIT without setting the WAIT parameter, the Update process is initiated on SAP and exits immediately without waiting for the Update process to complete.

If you select **Wait until commit call to SAP database is completed and returned** during configuration, the adapter will wait until all time critical updates to SAP database are completed before invoking the commit call. You can set the value for the Wait on a BAPI Commit call to be either a non-space character, which is interpreted as 'True', or set it a space character, which is interpreted as 'False.' If you set the value to 'True', then the BAPI call will wait indefinitely until the update process is completed before exiting.

There is a performance impact when you use the WAIT parameter as processing is delayed till the database update is completed. The need for a Wait on a BAPI Commit should to be evaluated based on your requirement.

If it is not selected, the commit call is performed immediately regardless of whether all the execution within a transaction is complete or not.

Note: This functionality is supported for BAPI interface and BAPI work unit interface.

Inbound processing for the BAPI interface

The adapter supports inbound processing (from the SAP server to the adapter) for simple BAPIs. A client application on the SAP server invokes a function through the adapter to an endpoint.

Synchronous and asynchronous RFC

For BAPI inbound processing, you can specify that the processing be handled synchronously (in which both the client application and the adapter must be available during processing) or asynchronously (in which the adapter does not have to be available when the client application invokes the function call). In synchronous processing, the client application waits for a response from the adapter. In asynchronous processing, the client application does not wait for a response.

The BAPI interface has two sets of activation specification properties (one for synchronous RFC and one for asynchronous RFC), which you use to set up inbound processing. You specify values for the properties with the external service wizard or through the administrative console.

The sequence of processing actions that result from an inbound request differ, depending on the selection you make during configuration from the **SAP Remote Function Call (RFC) type** list.

Synchronous RFC

If you select **Synchronous RFC** (the default) during configuration, the following processing steps occur:

1. The adapter starts event listeners, which listen for an RFC-enabled function event (which you specified with the RFCProgramID property) on the SAP server.
2. The RFC-enabled function event is pushed to the adapter by way of the event listeners.
3. The adapter resolves the operation and business object name using the received RFC-enabled function name.
4. The adapter sends the business object to an endpoint in a synchronous manner.
5. The adapter receives the response business object from the endpoint.
6. The adapter maps the response business object to an RFC-enabled function and returns it to the SAP server.

The adapter does not listen for events until the endpoint is active and available.

Asynchronous transactional RFC

If you select **Asynchronous Transactional/Queued RFC** during configuration, the following processing steps occur:

1. A client on the SAP server invokes an RFC-enabled function call on the adapter.

Note: To send the RFC-enabled functions from a queue on the SAP server, the client program on the SAP server delivers the events to a user-defined outbound queue.

A transaction ID is associated with the call.

The calling program on the SAP server does not wait to see whether the call to the adapter was successful, and no data is returned to the calling program.

2. The RFC-function call is placed on a list of functions to be delivered.
You can see the event list by entering transaction code SM58 on the SAP server.
3. The RFC-function call is invoked on the adapter. If the adapter is not available, the call remains in the list on the SAP system, and if the scheduler in SAP is configured, then the call is repeated at regular intervals until it can be processed by the adapter. If the scheduler is not configured, you need to process it manually.

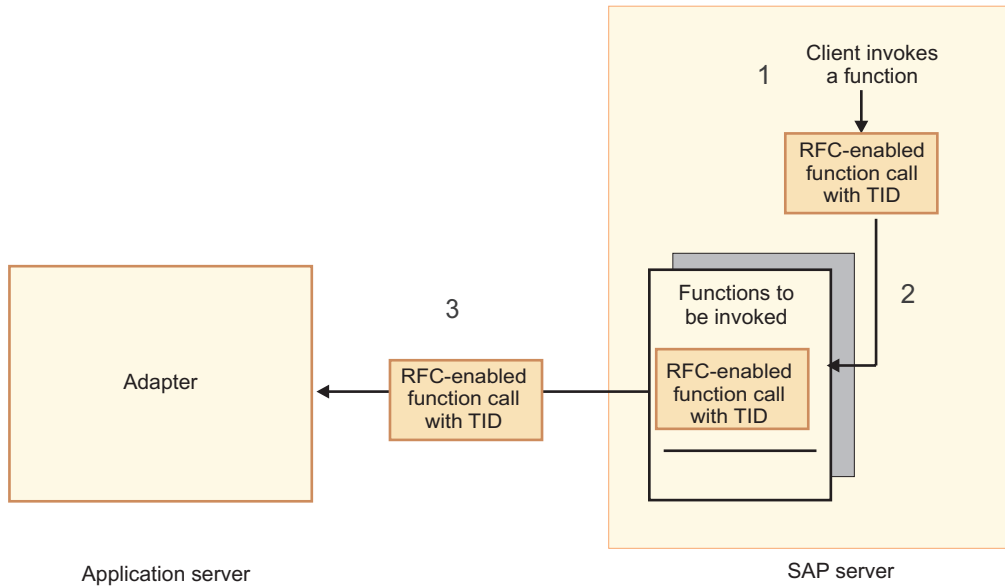


Figure 9. An RFC-enabled function call is invoked on the adapter.

When the SAP server successfully delivers the call event, it removes the function from the list.

4. If you selected **Ensure once-only event delivery**, the adapter sets the transaction ID in the event persistent table.
This is to ensure the event is not processed more than once.
5. The adapter resolves the operation and business object name using the received RFC-enabled function name.
6. The adapter sends the business object to an endpoint.

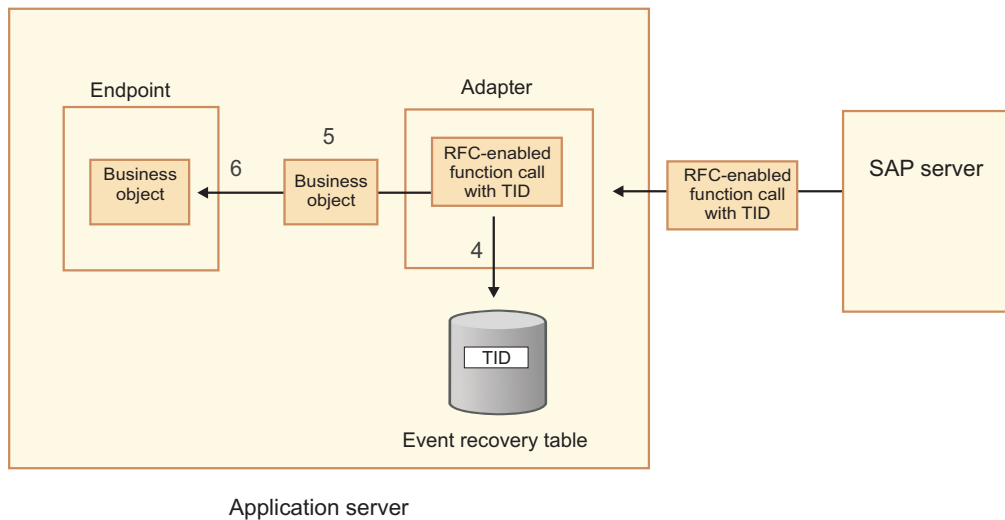


Figure 10. The adapter stores the TID, converts the function to a business object, and delivers the business object to the endpoint.

If you are sending functions from a user-defined queue on the SAP server, the functions are delivered in the order in which they exist on the queue. You can see the contents of the queue by entering the transaction code SMQ1 on the SAP server.

7. If the delivery is successful, and if you selected **Ensure once-only event delivery**, the adapter removes the transaction ID from the event persistent table.

If there is a failure when the adapter attempts to deliver the business object, the transaction ID remains in the event table. When another event is received from the SAP server, the following processing occurs:

- a. The adapter checks the transaction ID.
- b. If the event matches an ID in the table, the adapter processes the failed event once.

In other words, it does not send the event with the duplicate ID, ensuring that the event is processed only once.

Event recovery

You can configure the adapter for BAPI inbound processing so that it supports event recovery in case a failure occurs while the event is being delivered from the adapter to the endpoint. When event recovery is specified, the adapter persists the event state in an event recovery table that resides on a data source.

Event recovery is not the default; you must specify it by enabling once-only delivery of events during adapter configuration. You must also set up the data source before you can create the event recovery table.

Data source

Event recovery for BAPI inbound processing requires that a JDBC data source be configured. You use the administrative console to configure the data source. You select a JDBC provider (for example, Derby) and then create a new data source.

Event recovery table

You can create the event recovery table manually, or you can have the adapter create the event table. The value of the EP_CreateTable configuration property 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 3. 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 BAPI transactions specified in the application is observed. Event ordering is also affected by the number of event listeners. However, at some point all the BAPI transactions are transferred.
EVNTSTAT	INTEGER	Event processing status. Possible values are: <ul style="list-style-type: none"> • 0 (Created) • 1 (Executed) • 3 (In Progress) • -1 (Rollback)

Table 3. Event recovery table fields (continued)

Table field name	Type	Description
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	Not used for BAPI inbound processing.
BQPROC	INTEGER	Not used for BAPI inbound processing.
EVNTDATA	VARCHAR(255)	Not used.

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, Update, Retrieve, and Delete operations, each operation is represented by a business object, with the business objects being grouped together within a wrapper.

Note: The business object wrapper can be associated with multiple operations, but for a simple BAPI, each business object is associated with only one operation. For example, while a wrapper business object can contain BAPIOs for Create and Delete operations, BAPI_CUSTOMER_CREATE is associated with the Create operation, not the Delete operation.

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 process the simple BAPI call. Only one BAPI, the one that is associated with the operation to be performed, is called at a time.

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

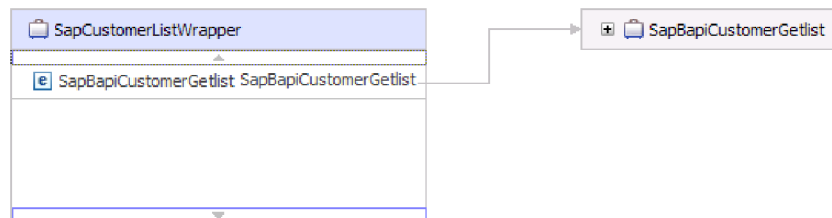


Figure 11. Example of a BAPI wrapper business object

If you selected **Asynchronous Transactional RFC** (for outbound or inbound processing) or **Asynchronous Queued RFC** (for outbound processing), the BAPI wrapper business object also contains a transaction ID. The transaction ID is used to resend the BAPI call if the receiving system is not available at the time of the initial call.

SapCustomerWrapper	
SAPTransactionID	string
SapBapiCustomerGetlist	SapBapiCustomerGetlist

Figure 12. Example of a BAPI wrapper business object with a transaction ID

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

SapBapiCustomerGetlist	
ControlIndicatorReadOneTimeCustomersOnly	string
MaximumNumberOfCustomers	int
SapReturn	SapReturn
SapAddressdata	SapAddressdata []
SapIdrange	SapIdrange []
SapSpecialdata	SapSpecialdata []

Figure 13. Example of a simple BAPI business object

Note: This object, which contains the results of the BAPI operation, is named according to the convention *Sap + Name of the structure*. For more information on naming conventions, refer to “Naming conventions for BAPI business objects” on page 315.

Note: If you select the **Generate business objects in SAP PI standard** option in the Specify the Discovery Properties window, the table type element will have a wrapper anonymous complex type, whose name is appended with a suffix “_Item”.

Additional information about the business object can be found in the application-specific information of the business object. For example, the application-specific information for a top-level object lists the type of BAPI and operation information.

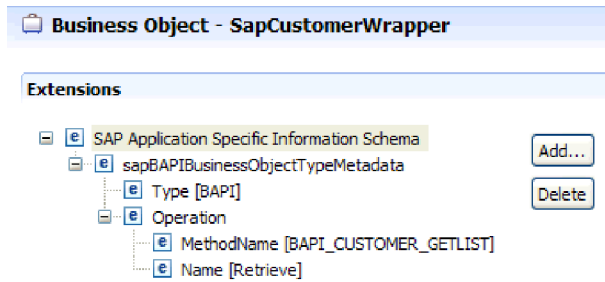


Figure 14. Application-specific information for a top-level object

Business objects for BAPI can also be generated without a wrapper. This is the recommended approach. If you are generating business objects without a wrapper, the execute operation is associated with each of the selected BAPIs by default. In the case of **Asynchronous Transactional RFC** or **Asynchronous Queued RFC** the transaction ID property is present in the top-level 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 figure displays an example of the BAPI business object that contains both simple parameters (for example, LanguageOfTheTexts) and structure parameters (for example, SapLinesDescr).

SapDdifFieldInfoGet	
<input type="checkbox"/>	UseParameterLfieldnameInstead string
<input type="checkbox"/>	TakeNamedIncludesIntoConsideration string
<input type="checkbox"/>	LanguageOfTheTexts string
<input type="checkbox"/>	IfFilledOnlyFieldWithThisLongName string
<input type="checkbox"/>	NameOfTheTableOfTheTypeForWhichInformationIsRequired string
<input type="checkbox"/>	UnicodeLengthWithWhichRuntimeObjectWasGenerated hexBinary
<input type="checkbox"/>	KindOfType string
<input type="checkbox"/>	SapDfiesWa SapDfiesWa
<input type="checkbox"/>	SapLinesDescr SapLinesDescr []
<input type="checkbox"/>	SapX030lWa SapX030lWa
<input type="checkbox"/>	SapDfiesTab SapDfiesTab []
<input type="checkbox"/>	SapFixedValues SapFixedValues []

Figure 15. The SapDdifFieldInfoGet business object

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

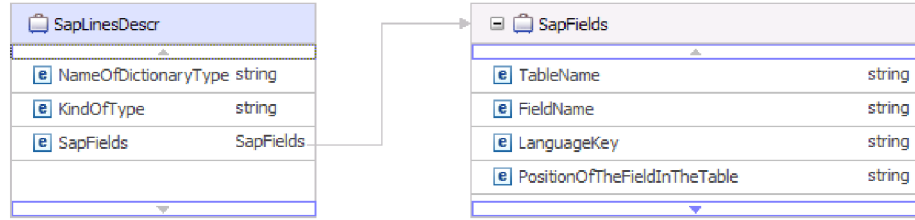


Figure 16. The SapLinesDescr business object

Note: The adapter also processes table types for import and export parameters.

Coordinating BAPI transactions and passing values between BAPI invocations

The transaction capability of IBM Business Process Manager or WebSphere Enterprise Service Bus can be used in a scenario where multiple BAPI requests are to be executed in an atomic fashion.

You can use the built-in transaction capabilities of IBM Business Process Manager or WebSphere Enterprise Service Bus in scenarios where chained BAPI invocation requests are to be executed in an atomic fashion.

To coordinate several BAPI requests, the various adapter outbound operations need to be joined in a transaction. To do this, set the Join Transaction Quality Of Service Qualifier value to **True** for the appropriate interfaces.

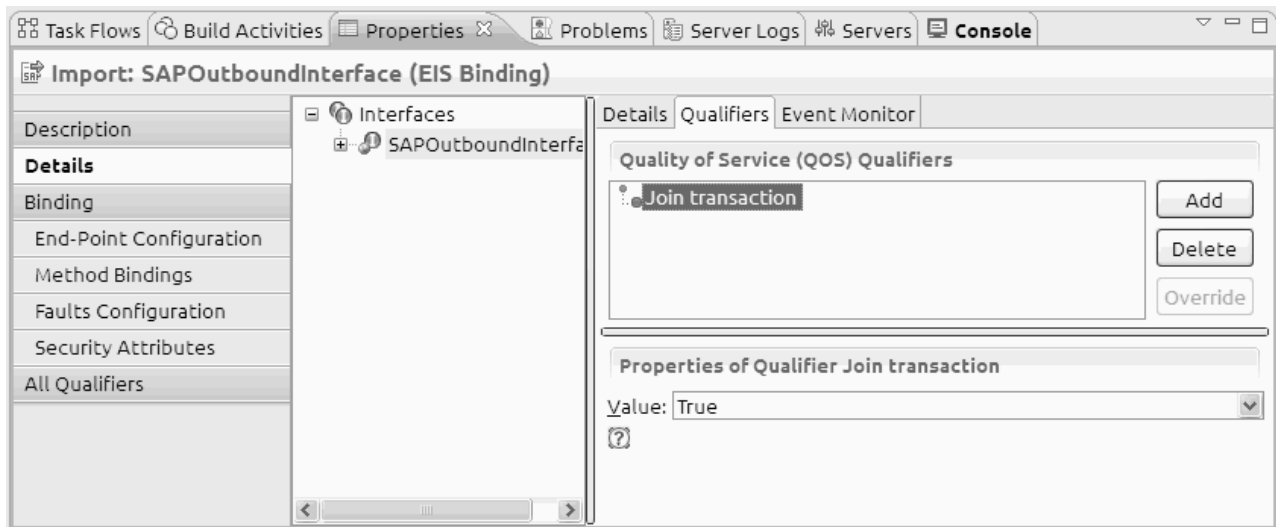


Figure 17. Join transaction Quality of Service Qualifier

The outbound calls can be made from either a Java component or a Mediation module. Once the interfaces are joined in a transaction, set the value of the Transaction Quality of Service Qualifier to **Global** for the appropriate Java component or Mediation module implementation in IBM Integration Designer. With this setting, IBM Business Process Manager or WebSphere Enterprise Service Bus ensures that all the BAPI calls execute successfully or all of them fail.

The whole transaction is committed after all the BAPI executions complete successfully. Any exceptions raised during the execution of the BAPIs will result in

a rollback of all changes made by the previous requests.

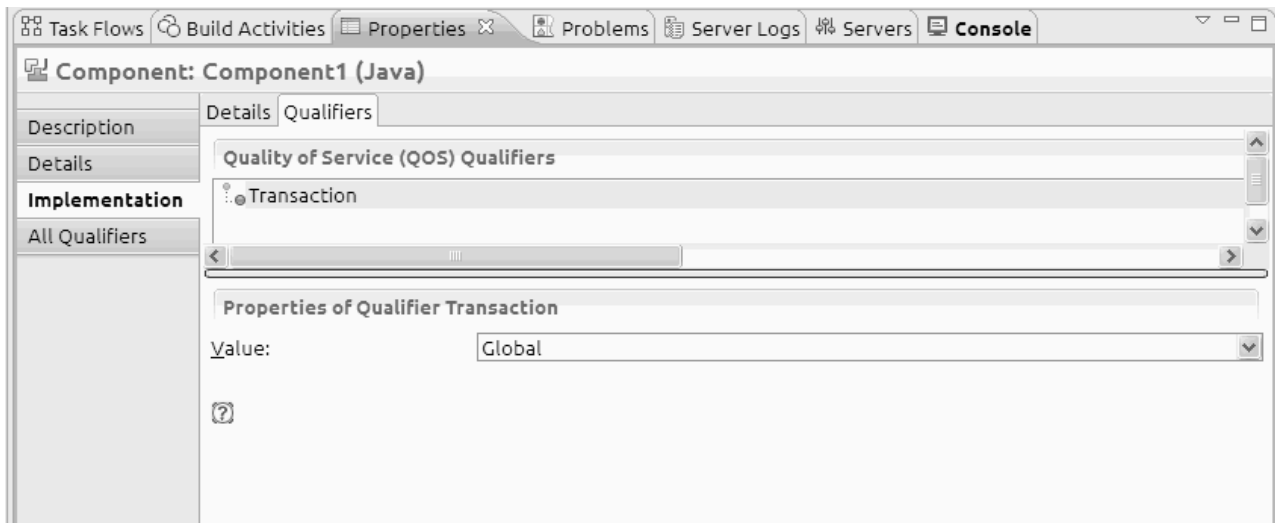


Figure 18. Transaction Quality of Service Qualifier

In order to achieve chaining of BAPIs in sequence (output of one BAPI becomes the input of the other), you can use the **Interface** and **Data mapping** features available in IBM Business Process Manager or WebSphere Enterprise Service Bus.

You can find the transaction qualifiers for setting the Join transaction and Global transaction in the **Qualifiers** tab in **Details** section of the properties.

The BAPI work unit interface

WebSphere Adapter for SAP Software supports outbound processing for BAPI units of work. A BAPI work unit consists of a set of BAPIs that are processed in sequence to complete a task.

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 work unit. The following three BAPIs illustrate the kind of sequence that forms such a work unit:

- BAPI_ADDRESSEMP_REQUEST
- BAPI_ADDRESSEMP_CHANGE
- BAPI_ADDRESSEMP_APPROVE

The first BAPI locks the employee record, the second updates the record, and the third approves the update. The advantage of using a BAPI work unit is that the client application can request the employee record change with a single call, even though the work unit consists of three separate functions. In addition, if SAP requires that the BAPIs be processed in a specific sequence for the business flow to complete correctly, the work unit supports this sequence.

Outbound processing for the BAPI work unit interface

In BAPI work unit outbound processing, a client application sends a request to the SAP server. Processing is handled synchronously (the client application waits for a response from the SAP server).

For BAPI work units, the following processing steps occur:

1. The adapter receives a request from a client application in the form of a BAPI business object.
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 process the BAPI or RFC function call in the SAP application.
4. After passing the data to the SAP server, the adapter handles the response from SAP and converts it back into the business object format required by the client application.
5. The adapter then sends the response back to the client application.

Business object structure for a BAPI work unit

A business object that represents a BAPI work unit (also known as a BAPI transaction) is actually a wrapper object that contains multiple child BAPI objects. Each child BAPI object within the wrapper object represents a simple BAPI.

The adapter supports a BAPI work unit using a top-level wrapper business object that consists of multiple child BAPIs, each one representing a simple BAPI in the sequence. The BAPI wrapper object represents the complete work unit, while the child BAPI objects contained within the BAPI wrapper object represent the individual operations that make up the work unit.

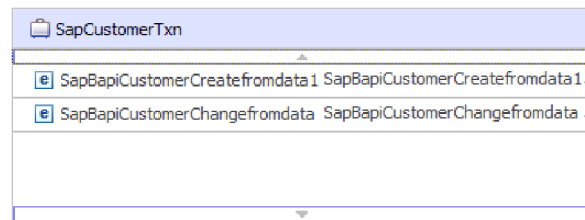


Figure 19. Example of a top-level wrapper object for a BAPI work unit

The adapter uses the sequence of operations in the operation metadata to process the BAPIs in the work unit.

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.

Note: This object, which contains the results of the BAPI operation, is named according to the convention *Sap + Name of the structure*. For more information about naming conventions, refer to “Naming conventions for BAPI business objects” on page 315.

Additional information about the business object can be found in the application-specific information of the business object. For example, the application-specific information for a BAPI work unit lists the type of BAPI and the operations that make up the work unit.

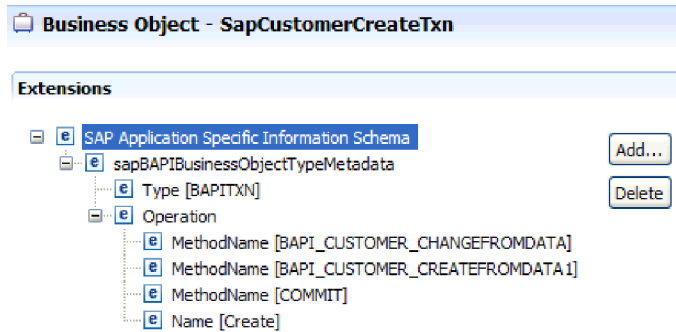


Figure 20. Application-specific information for a BAPI work unit

Note: The adapter does not provide an automated rollback mechanism for BAPI work units. Rollback of a BAPI work unit 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.

The BAPI result set interface

The WebSphere Adapter for SAP Software supports outbound processing for BAPI result sets. In outbound processing, client applications call BAPIs and other RFC-enabled functions on the SAP server.

BAPI result set interface

BAPI result sets use the GetList and GetDetail functions to retrieve an array of data from the SAP server. The information returned from the GetList function is used as input to the GetDetail function.

For example, if you want to retrieve information about a set of customers, you use BAPI_CUSTOMER_GETLIST, which acts as the query BAPI, and BAPI_CUSTOMER_GETDETAIL, which acts as the result BAPI. The BAPIs perform the following steps:

1. The BAPI_CUSTOMER_GETLIST call returns a list of keys (for example, CustomerNumber).
2. Each key is mapped dynamically to the business object for BAPI_CUSTOMER_GETDETAIL.
3. BAPI_CUSTOMER_GETDETAIL is processed multiple times, so that an array of customer information is returned.

You use the external service wizard to discover the BAPI_CUSTOMER_GETLIST and BAPI_CUSTOMER_GETDETAIL functions and to build the key relationship between the two BAPIs. The wizard then generate business object definitions for these BAPIs along with other SCA service artifacts. At run time, the client sets the values in the BAPI_CUSTOMER_GETLIST business object, and the adapter returns the corresponding set of customer detail records from the SAP server.

Outbound processing for the BAPI result set interface

In BAPI result set outbound processing, a client application sends a request to the SAP server. Processing is handled synchronously (the client application waits for a response from the SAP server).

For BAPI result sets, the following processing steps occur:

1. The adapter receives a request from a client application in the form of a BAPI business object.
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 process the BAPI or RFC function call in the SAP application.
4. After passing the data to the SAP server, the adapter handles the response from SAP and converts it back into the business object format required by the client application.
5. The adapter then sends the response back to the client application.

Business object structure for a BAPI result set

The top-level business object for a result set is a wrapper that contains a GetDetail business object. The GetDetail business object contains the results of a query for SAP data. The GetDetail business object also contains, as a child object, the query business object. The query business object represents a GetList BAPI. These two BAPIs work together to retrieve information from the SAP server.

An example of a business object for a BAPI result set is shown in the following figure. This is a wrapper object that contains the result method business object.

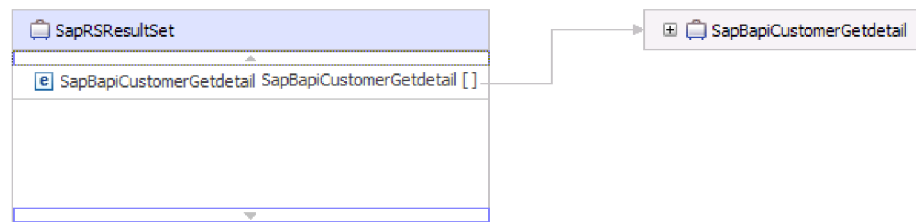


Figure 21. Example of a business object for a BAPI result set

The following figure shows an example of the `SapBapiCustomerGetdetail` business object:

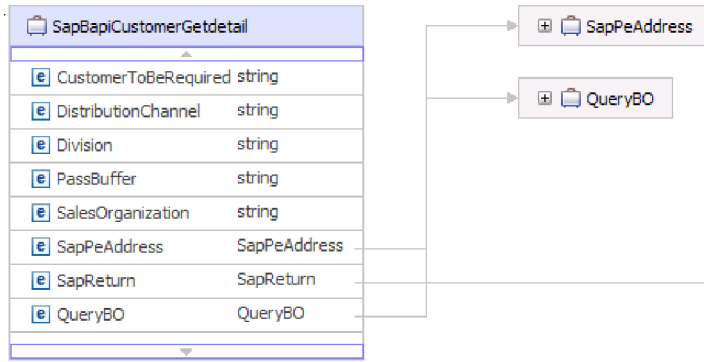


Figure 22. Example of a GetDetail business object

Note that the last property is the query business object.

The following figure shows an example of the query business object (SapBapiCustomerGetList).

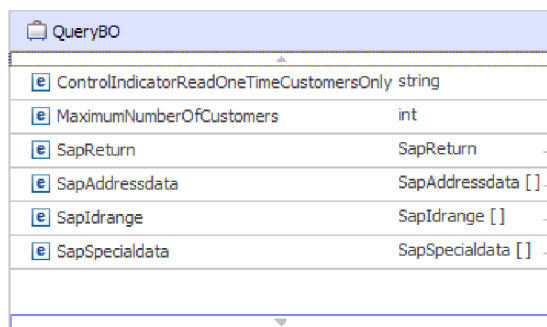


Figure 23. Example of a query business object

Note: This object, which contains the results of the BAPI operation, is named according to the convention *Sap + Name of the structure*. For more information about naming conventions, refer to “Naming conventions for BAPI business objects” on page 315.

Additional information about the business object can be found in the application-specific information of the business object. For example, the application-specific information for SapBapiCustomerGetdetail lists the type of BAPI and operation information.

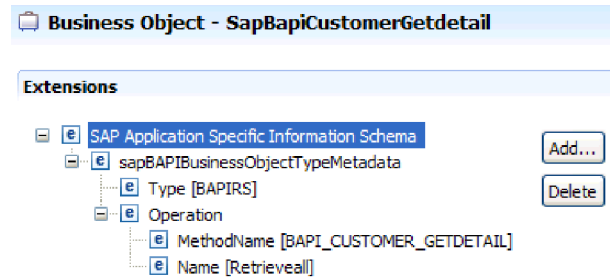


Figure 24. Application-specific information for SapBapiCustomerGetdetail

The ALE interfaces

The SAP ALE interface enables business process integration and asynchronous data communication between two or more SAP systems or between SAP and external systems. The data is exchanged in the form of Intermediate Documents (IDocs).

The adapter supports outbound and inbound processing by enabling the exchange of data in the form of business objects.

- For inbound processing, SAP pushes the data in IDocs to the SAP adapter. The adapter converts the IDocs to business objects and delivers them to the endpoint.
- For outbound processing, the SAP adapter converts the business object to an IDoc and delivers it to SAP.

To use the ALE interface for inbound processing, you must make sure that your SAP server is properly configured (for example, you must set up a partner profile and register a program ID to listen for events).

Application systems are loosely coupled in an ALE integrated system, and the data is exchanged asynchronously.

IDocs

IDocs are containers for exchanging data in a predefined (structured ASCII) format across system boundaries. The IDoc type indicates the SAP format that is to be used to transfer the data. An IDoc type can transfer several message types (the logical messages that correspond to different business processes). IDocs are used for outbound and inbound processing. The SAP adapter supports the basic and extension type of IDocs.

For example, if an application developer wants to be notified when a sales order is created on the SAP server, the developer runs the external service wizard to discover the ORDERS05 IDoc on the SAP server. The wizard then generates the business object definition for ORDERS05. The wizard also generates other artifacts such as an EIS export component and SCA interfaces. These artifacts are saved in an integration module. The application developer can then use this SAP inbound module to build the application.

IDocs are exchanged for inbound and outbound events, and IDocs can be exchanged either as individual documents or in packets.

For outbound processing, the adapter uses the IDoc business object to populate the appropriate RFC-enabled function call made to the SAP server.

For inbound processing, IDocs can be sent from the SAP server as parsed or unparsed documents

- For parsed documents, the adapter parses the IDoc and creates a business object that reflects the internal structure of the IDoc.
- For unparsed IDocs, the adapter processes the IDoc but does not convert the data portion of the IDoc.

Transactional RFC processing

The adapter uses tRFC (transactional RFC) to guarantee delivery and to ensure that each IDoc is exchanged only once with SAP. The tRFC component stores the called RFC function in the database of the SAP system along with a unique transaction identifier (TID).

The most common reason for using transaction ID support is to ensure once and only once delivery of data. To make sure of this feature, select the transaction RAR file (CWYAP_SAPAdapter_Tx.rar) when you configure the adapter.

Note: The SAP transaction ID property is always generated by the external service wizard; 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.

Queued RFC processing

The adapter uses qRFC (queued transactional RFC) to ensure that IDocs are delivered in sequence to a queue on the SAP server or are received in sequence from the SAP server.

Outbound processing for the ALE interfaces

The adapter supports outbound processing (from the adapter to the SAP server) for the ALE interface. ALE uses IDocs for data exchange, and the adapter uses business objects to represent the IDocs.

The following list describes the sequence of processing actions that result from an outbound request using the ALE interface.

Note: The client application that makes the request uses the interface information that was generated by the external service 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. If you are using the CWYAP_SAPAdapter_Tx.rar and have provided a Transaction ID, the adapter uses it while posting the IDocs to the SAP system. If you have not provided a Transaction ID, the adapter will create one before posting. If you are using the qRFC protocol, the adapter passes the IDoc data in the order specified in the wrapper business object to the specified queue on the SAP server. The adapter uses the same Transaction ID for all the IDocs in the wrapper and posts all of them using one call.
4. The adapter throws an exception if the data record is empty. Refer to “ALE business object structure” on page 60 for more information on data records.
5. 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.

Import the CWYAP_SAPAdapter_Tx.rar version of the adapter when you create a module that makes use of transactional (tRFC) or queued transactional (qRFC) processing.

Inbound processing for the ALE interfaces

The adapter supports inbound processing (from the SAP server to the adapter) for the ALE interface.

When you configure a module for the ALE interface, you can indicate whether the IDocs are sent as a packet. In addition, you can specify whether IDocs are sent parsed or unparsed. You make these selections on the Configuration Properties window of the external service wizard. The selections you make are reflected in the application-specific information for the IDoc business object.

The following list describes the sequence of processing actions that result from an inbound request using the ALE interface.

1. The adapter starts event listeners to the SAP server.
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.

If you have selected split IDocs and SAP is sending packet IDocs, the adapter delivers each IDoc inside the packet as an individual event to the end point.

During recovery, the SAP system has to resubmit the whole packet. The adapter only delivers IDocs which have not been delivered in previous attempts from the packet.

Note: The adapter is able to listen to and deliver events from multiple SAP systems using multiple activation specs.

The adapter is also able to deliver events to multiple 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.

Note: When multiple endpoints subscribe to the same events from the same event store, the adapter assures event delivery to active endpoints only. Any endpoints that are inactive do not receive the event. If there are multiple endpoints and any one endpoint is not active, the message is skipped for that endpoint and the adapter delivers the event only to the active endpoints. If all the endpoints are inactive, the event is rolled back, and the event needs to be resubmitted from SAP.

The following table provides an overview of the differences between the ALE interface and the ALE pass-through IDoc interface for inbound processing.

Table 4. differences between the ALE interface and the ALE pass-through IDoc interface

Interface	When to use	SplitIDoc = true	SplitIDoc = false	Parsed IDoc = true	Parsed IDoc = false
ALE inbound	This interface converts the raw incoming IDocs to business objects, which are readily consumable by the client at the endpoint.	On receiving the IDoc packet from SAP, the adapter converts the IDocs to business objects, one by one, before sending each one to the endpoint.	On receiving the IDoc packet from SAP, the adapter converts the IDocs in the packet as one business object before sending it to the endpoint.	The incoming IDoc is parsed for both control record and data record. The individual segments in the IDoc are read and parsed to convert into business objects.	The incoming IDoc is only partially parsed (the control record of the IDoc is parsed but the data record is not). The client at the endpoint is responsible for parsing the data record.

Table 4. differences between the ALE interface and the ALE pass-through IDoc interface (continued)

Interface	When to use	SplitIDoc = true	SplitIDoc = false	Parsed IDoc = true	Parsed IDoc = false
ALE pass-through IDoc	This interface wraps the raw incoming IDoc in a business object before delivering it to the client at the endpoint. The client is responsible for parsing the raw IDoc.	On receiving the IDoc packet from SAP, the adapter wraps each raw IDoc within a business object before sending the objects, one by one, to the endpoint.	On receiving the IDoc packet from SAP, the adapter wraps the raw IDoc packet in a business object before sending it to the endpoint.	This attribute is not applicable to the ALE pass-through IDoc.	This attribute is not applicable to the ALE pass-through IDoc interface. (Neither the control record nor the data record of the IDoc is parsed.)

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.
Log and trace files are in the `/profiles/profile_name/logs/server_name` path of the folder in which IBM Business Process Manager or WebSphere Enterprise Service Bus is installed.
2. The adapter relies on SAP JCo's retry handling to restart the JCo Server.
 - The adapter stops the server after the SAP JCo Server tries again for the specified `RetryLimit` value as specified in the activation specification.

Note: If the retry interval does not behave as specified in the activation specification, download the latest SAP JCo 3.0.6.

3. If all the retry attempts fail, the adapter logs the relevant message and CEI events, stops message end points and stops trying to recover the ALE event listener.

Note: You must restart the adapter or SCA application in this case.

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.

Note: You must restart the adapter or SCA application in this case.

Event recovery

You can configure the adapter for ALE inbound processing so that it supports event recovery in case of abrupt termination. When event recovery is specified, the adapter persists the event state in an event recovery table that resides on a data source. Event recovery is not enabled by default; you must specify it by enabling once-only delivery of events during adapter configuration.

Data source

Event recovery for ALE inbound processing requires that a JDBC data source be configured. You use the Process Administrative Console to configure the data source. You select a JDBC provider (for example, Derby) and then create a new data source.

Event recovery table

You can create the event recovery table manually, or you can have the adapter create the event table. The value of the EP_CreateTable configuration property 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 5. 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: <ul style="list-style-type: none">• 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.

To use event recovery for multiple endpoints, you must configure a separate event recovery table for each endpoint, although you can use the same data source (for example, Derby) to hold all the event recovery tables.

Event processing for parsed IDocs

An inbound event can contain a single IDoc or multiple IDocs, with each IDoc corresponding to a single business object. The multiple IDocs are sent by the SAP server to the adapter in the form of an IDoc packet. During adapter configuration, you can specify whether the packet can be split into individual IDocs or whether it must be sent as one object (non-split).

Event processing begins when the SAP server sends a transaction ID to the adapter. The following sequence occurs:

1. 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 to the adapter.
3. The adapter converts the IDoc to a business object and sends it to the endpoint.

Note: For single IDocs and non-split IDoc packets, 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. For split packets only, the adapter performs the following tasks:
 - a. 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.
 - b. 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.
 - c. 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.

If the adapter encounters an error while processing a split 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 as 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.

This property is not used for single IDocs and for non-split IDoc packets.

- d. The SAP system sends a COMMIT call to the adapter.
 - e. After the adapter delivers all the business objects in the IDoc packet to the message endpoint, it updates the event status to 1 (Executed).
 - f. 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.
5. If an exception occurs either while the adapter is processing the event or if the endpoint generates an exception, the event status is updated to -1 (Rollback).
 6. If no exception occurs, the SAP server sends a CONFIRM call to the adapter.
 7. The adapter 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

Unparsed IDocs are passed through, with no conversion of the data (that is, the adapter 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 consumer of the IDoc parses the IDoc.

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 external service wizard.

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 25. 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 26. 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.
- The DummyKey feature is not supported for unparsed IDocs.

IDoc status updates

To monitor 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 server. 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. Monitoring of IDocs applies only to inbound processing (when the IDoc is sent from the SAP server to the adapter).

The adapter updates a status IDoc (ALEAUD) and sends it to the SAP server.

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

The status codes and their associated text are configurable properties of the adapter, as specified in the activation specification properties and shown in the following list:

- Success Code
- Failure Code
- Success Text
- Failure Text

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 6. 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 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 format is used for inbound and outbound processing.

Wrapper business object

The wrapper business object contains a transaction ID, a queue name, and one or more IDoc business objects. The transaction ID (`SAPTransactionID`) is used to ensure once-only delivery of business objects, and the queue name (`qRFCQueueName`) specifies the name of the queue on the SAP server to which the IDocs should be delivered. If you are not using transaction IDs or queues, these properties are blank.

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 wrapper business object, which, in this example, contains one IDoc business object.

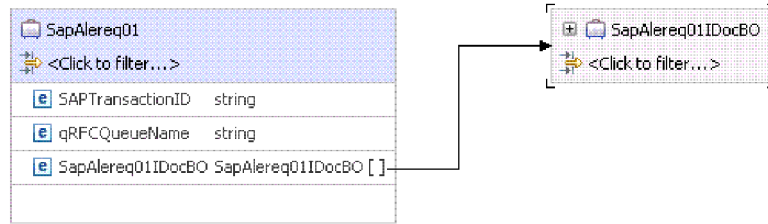


Figure 27. Example of an ALE wrapper business object

Note that the transaction ID and queue name attributes are present in the business object even if you are not using the tRFC or qRFC features.

IDoc business object

The IDoc business object (SapAlereq01IDocBO, in the example) contains a control record, a data record and a dummy key.

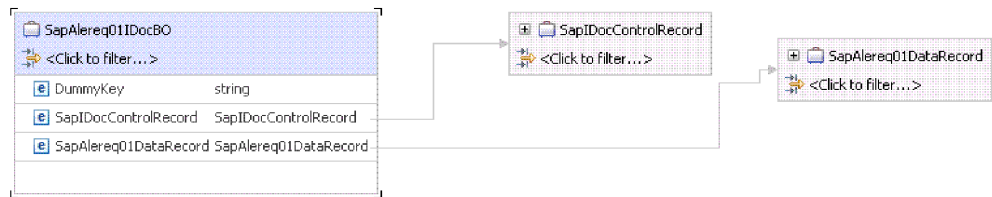


Figure 28. Example IDoc business object structure

Control record

The control record business object contains the metadata required by the adapter to process the IDoc business object.

The control record can be generated from SAP field names or from SAP field descriptions. While configuring the properties for the control record you can specify if you want the control record generated from SAP field names or from SAP field descriptions. Check the check box to use SAP field names to generate attribute names if you want the control record generated from field names.

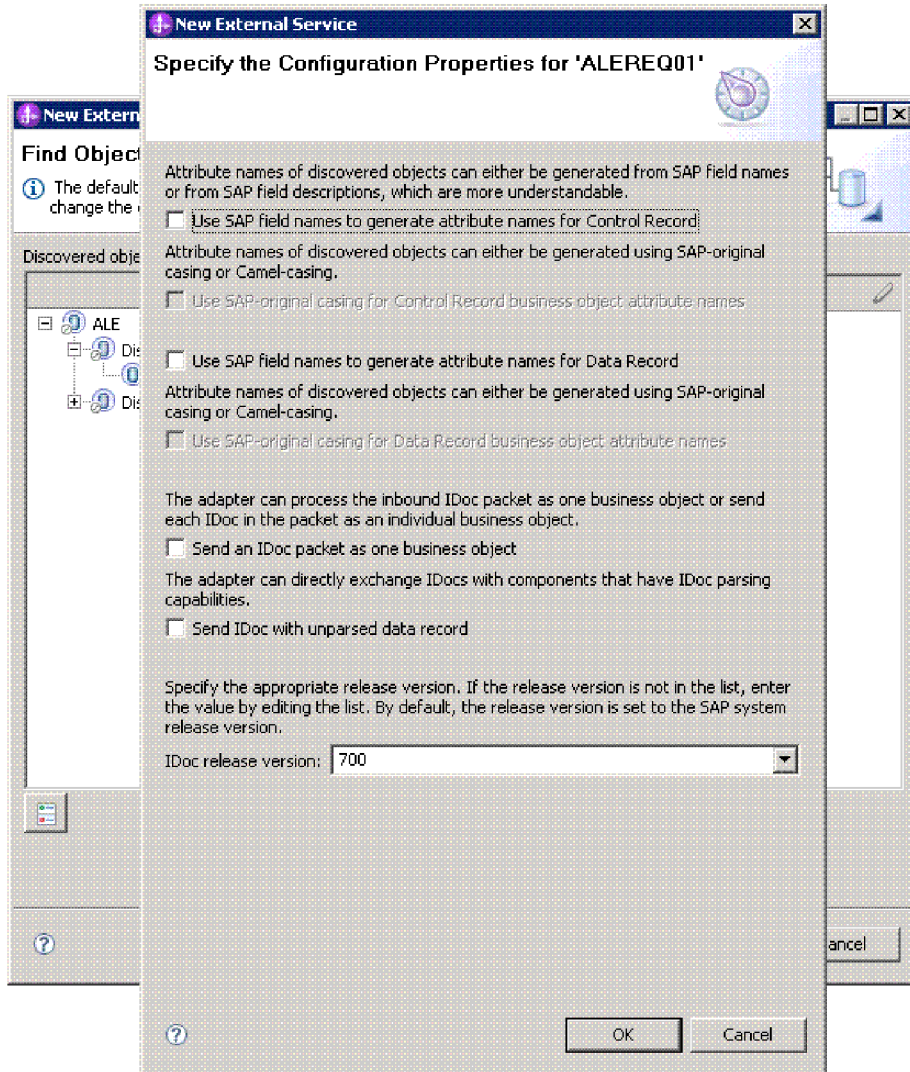


Figure 29. Configuration properties window

The control record gets generated as shown below:

SapIDocControlRecord	
<Click to filter...>	
<input checked="" type="checkbox"/>	RCVPOR string
<input checked="" type="checkbox"/>	SNDPRT string
<input checked="" type="checkbox"/>	SNDPFC string
<input checked="" type="checkbox"/>	TEST string
<input checked="" type="checkbox"/>	MESCOD string
<input checked="" type="checkbox"/>	IDOCTYP string
<input checked="" type="checkbox"/>	SNDPRN string
<input checked="" type="checkbox"/>	STD string
<input checked="" type="checkbox"/>	ARCKEY string
<input checked="" type="checkbox"/>	CRETIM string
<input checked="" type="checkbox"/>	DOCTYP string
<input checked="" type="checkbox"/>	TABNAM string
<input checked="" type="checkbox"/>	CREDAT string
<input checked="" type="checkbox"/>	REFGRP string
<input checked="" type="checkbox"/>	OUTMOD string
<input checked="" type="checkbox"/>	MANDT string
<input checked="" type="checkbox"/>	RCVSAD string
<input checked="" type="checkbox"/>	MESTYP string
<input checked="" type="checkbox"/>	RCVPRN string
<input checked="" type="checkbox"/>	STDVRS string
<input checked="" type="checkbox"/>	RCVLAD string
<input checked="" type="checkbox"/>	DIRECT string
<input checked="" type="checkbox"/>	SNDLAD string
<input checked="" type="checkbox"/>	EXPRESS string
<input checked="" type="checkbox"/>	SNDSAD string
<input checked="" type="checkbox"/>	DOCREL string
<input checked="" type="checkbox"/>	SERIAL string
<input checked="" type="checkbox"/>	SNDPOR string
<input checked="" type="checkbox"/>	DOCNUM string

Figure 30. Control record generated with SAP field names

If you do not check the Use SAP field names check box, the control record structure would be displayed as shown in the figure below:

SapIDocControlRecord	
<Click to filter...>	
ReceiverPort	string
PartnerTypeOfSender	string
PartnerFunctionOfSender	string
TestFlag	string
LogicalMessageCode	string
NameOfBasicType	string
PartnerNumberOfSender	string
EdiStandard	string
EdiArchiveKey	string
IdocCreationTime	string
IdocType	string
NameOfTableStructure	string
IdocCreationDate	string
ReferenceToMessageGroup	string
OutputMode	string
Client	string
IdocSadrFieldsInTotal	string
LogicalMessageType	string
PartnerNumberOfRecipient	string
VersionOfEdiStandard	string
LogicalAddressOfRecipient	string
DirectionForIdocTransmission	string
LogicalAddressOfSender	string
OverridingInInboundProcessing	string
IdocSadrFieldsInTotalSender	string
SapReleaseForIdoc	string
EdiAleSerializationField	string
SenderPort	string
IdocNumber	string
EdiMessageType	string
ReferenceToInterchangeFile	string
PartnerFunctionOfRecipient	string
ReferenceToMessage	string
LogicalMessageFunction	string
Extension	string

Figure 31. Control record generated with SAP field descriptions

Data record

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. The data record business object is generated for a parsed IDoc. The data record business object contains all the segments of the IDoc. Each segment in turn has a child business object as shown below. The segment attributes can also be generated using SAP field names or field descriptions. You can use SAP field names to generate attribute names.

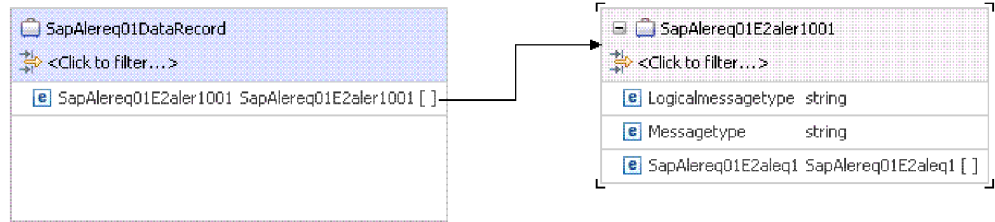


Figure 32. Example data record business object structure

Unparsed IDocs

For an unparsed IDoc, in which the data part of the IDoc is not parsed by the adapter, the IDoc business object contains a dummy key, a control record, and the IDoc data. The IDoc data is of hexBinary type and represents the whole data record containing segments in binary content.

The following figure illustrates a wrapper business object for an unparsed IDoc and the associated IDoc business object.

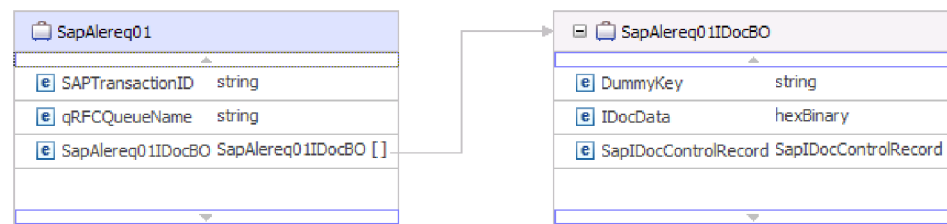


Figure 33. Example of an ALE wrapper business object for an unparsed IDoc

Application-specific information

Additional information about the business object can be found in the application-specific information of the business object. For example, the application-specific information for SapAleReq01 lists whether the IDoc packet is split and provides information about the operation.

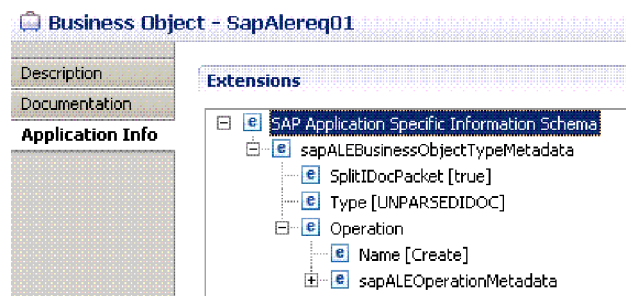


Figure 34. Application-specific information for the business object

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 path to the property from which the value should be set. For example: dataRecord/SapOrders05e2edk01005/IdocDocumentNumber

The following figure shows an example of property-level application-specific information that includes the DummyKey field.

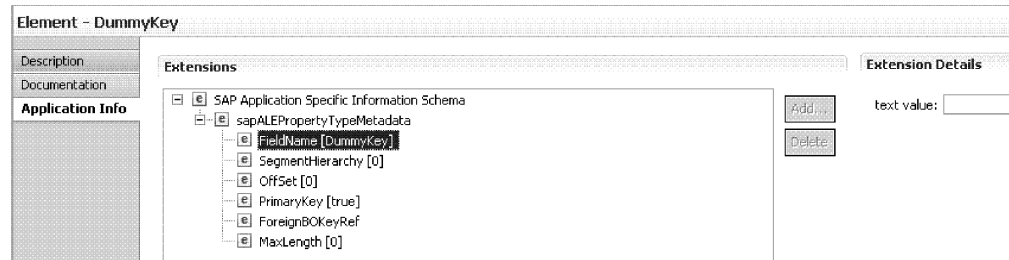


Figure 35. Property-level application-specific information for DummyKey

- Multiple cardinality objects are not supported. If the path contains a multiple cardinality object, the value is ignored and the default first index is used.
- 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.

Dummy key processing is not supported for unparsed IDocs. You can use dummy keys in the ALE inbound interface.

The ALE pass-through IDoc interface

The ALE pass-through IDoc interface enables business process integration and asynchronous data communication between two or more SAP systems or between SAP and external systems. The data is exchanged in the form of Intermediate Documents (IDocs).

The adapter supports outbound and inbound processing by enabling the exchange of data in the form of business objects.

- For inbound processing, SAP pushes the data in IDocs to the SAP adapter. The adapter converts the IDocs to business objects and delivers them to the endpoint.
- For outbound processing, the SAP adapter converts the business object to an IDoc and delivers it to SAP.

To use the ALE pass-through IDoc interface for inbound processing, you must make sure that your SAP server is properly configured (for example, you must set up a partner profile and register a program ID to listen for events).

Application systems are loosely coupled in an ALE integrated system, and the data is exchanged asynchronously.

IDocs

IDocs are containers for exchanging data in a predefined (structured ASCII) format across system boundaries. The IDoc type indicates the SAP format that is to be used to transfer the data. An IDoc type can transfer several message types (the logical messages that correspond to different business processes). IDocs are used for outbound and inbound processing. The Adapter supports basic and extension IDoc types.

IDocs are exchanged for inbound and outbound events, and IDocs can be exchanged either as individual documents or in packets. For both outbound and inbound processing, the adapter does no conversion of the IDoc. This is useful when the client wants to perform the IDoc parsing.

Transactional RFC processing

The adapter uses tRFC (transactional RFC) to guarantee delivery and to ensure that each IDoc is exchanged only once with SAP. The tRFC component stores the called RFC function in the database of the SAP system along with a unique transaction identifier (TID).

The most common reason for using transaction ID support is to ensure once and only once delivery of data. To make sure of this feature, select the transaction RAR file (CWYAP_SAPAdapter_Tx.rar) when you configure the adapter.

Note: The SAP transaction ID property is always generated by the external service wizard; 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.

Queued RFC processing

The adapter uses qRFC (queued transactional RFC) to ensure that IDocs are delivered in sequence to a queue on the SAP server or are received in sequence from the SAP server.

Outbound processing for the ALE pass-through IDoc interface

The adapter supports outbound processing (from the adapter to the SAP server) for the ALE pass-through IDoc interface. ALE uses IDocs for data exchange, and the adapter uses business objects to represent the IDocs.

The following list describes the sequence of processing actions that result from an outbound request using ALE pass-through IDoc interface.

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

1. The adapter receives a request, which includes a wrapper business object, from a client application.

Note: The wrapper business object contains a data stream representing the IDoc. No separate IDoc business object exists for pass-through IDocs

2. The adapter supports multiple IDocs in the data stream. The data stream content can be of two types:
 - a. Inline format without a delimiter - as supported in previous versions. In the inline format, multiple IDocs are differentiated using the header **EDIDC_40**
 - b. Delimited content format - supported in the current version. In the delimited format, you can post multiple IDocs to the SAP system by using a delimiter in between, to achieve the standard length of 1063 characters as specified by the SAP adapter. You can use delimiters to mark IDoc boundaries and also mark Control and Data Records boundaries within an IDoc.

When you insert a delimiter between Data Records, the adapter can then identify each Data Record and pad them to meet SAP specifications. The different uses of the delimiter are as follows:

- Single IDoc:

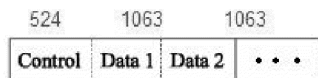


Figure 36. Single IDoc

- Single IDoc with '\n' character to separate data records

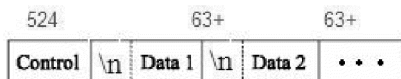


Figure 37. Single IDoc with '\n' character to separate data records

- Multiple IDocs



Figure 38. Multiple IDocs

- Multiple IDocs with the '\n' character separating the Data Records

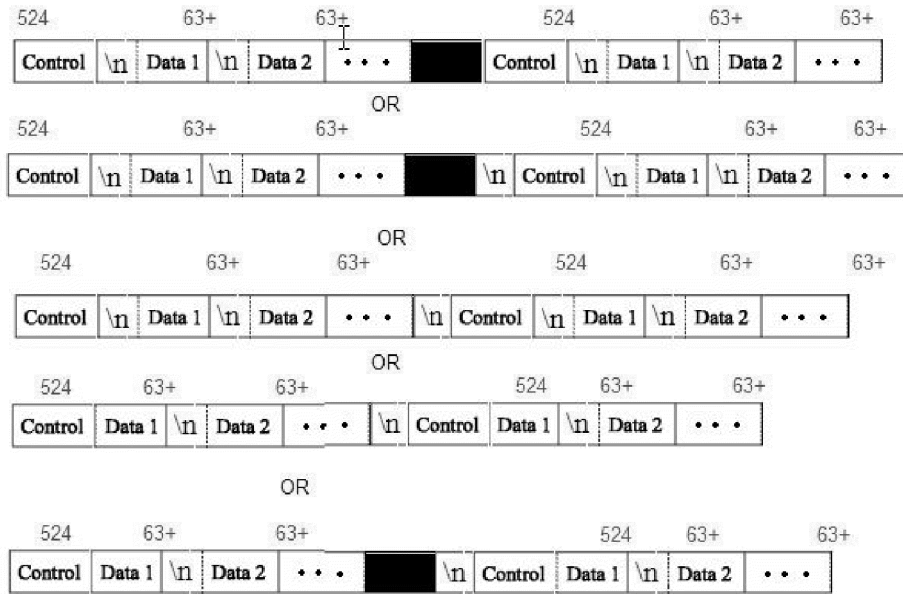


Figure 39. Multiple IDocs with the '\n' character separating the Data Records

3. The adapter uses the IDoc wrapper business object to populate the appropriate RFC-enabled function call used by the ALE interface.
4. The adapter establishes an RFC connection to the ALE interface and passes the IDoc data to the SAP system. If you are using the CWYAP_SAPAdapter_Tx.rar and have provided a Transaction ID, adapter uses it while posting the IDoc to SAP. If you have not provided a Transaction ID, the adapter will create one before posting. If you are using the qRFC protocol, the adapter passes the IDoc data in the order specified in the wrapper business object to the specified queue on the SAP server. The adapter uses the same Transaction ID for all the IDocs in the wrapper and posts all of them using one call.
5. 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.

Import the CWYAP_SAPAdapter_Tx.rar version of the adapter when you create a module that makes use of transactional (tRFC) or queued transactional (qRFC) processing.

Inbound processing for the ALE pass-through IDoc interface

The adapter supports inbound processing (from the SAP server to the adapter) for the ALE pass-through IDoc interface.

When you are configuring a module for the ALE pass-through interface, you can indicate whether the IDocs are sent as a packet. You make this selection on the

Configuration Properties window of the external service wizard. The selection you make is reflected in the application-specific information for the IDoc wrapper business object.

You can also choose to send the IDoc in Flat File format to the end point. You make this selection on the Specify the Configuration Properties for 'object' screen using the **Send IDoc in flat file format** check box. This feature will only work if the adapter is configured to work with unparsed Control Records where the **Parse the IDoc Control Record** check box is disabled.

Note: When you use the ALE pass-through IDoc interface, a wrapper business object contains a data stream representing the IDoc. No separate IDoc business object exists for pass-through IDocs.

The following list describes the sequence of processing actions that result from an inbound request using the ALE interface.

1. The adapter starts event listeners to the SAP server.
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.

If you have selected split IDocs and SAP is sending packet IDocs, the adapter delivers each IDoc inside the packet as an individual event to the end point. During recovery, the SAP system has to resubmit the whole packet. The adapter only delivers IDocs which have not been delivered in previous attempts from the packet.

Note that the adapter is able to listen to and deliver events from multiple SAP systems using multiple activation specs.

The adapter is also able to deliver events to multiple endpoints. You enable delivery to multiple endpoints by configuring multiple activation specifications that exist in the same module.

- 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.

Note: When multiple endpoints subscribe to the same events from the same event store, the adapter assures event delivery to active endpoints only. Any endpoints that are inactive do not receive the event. If there are multiple endpoints and any one endpoint is not active, the message is skipped for that endpoint and the adapter delivers the event only to the active endpoints. If all the endpoints are inactive, the event is rolled back, and the event needs to be resubmitted from SAP.

The following table provides an overview of the differences between the ALE interface and the ALE pass-through IDoc interface for inbound processing.

Table 7. Differences between ALE interface and ALE pass-through interface

Interface	When to use	SplitIDoc = true	SplitIDoc = false	Parsed IDoc = true	Flat File format = true
ALE inbound	This interface converts the raw incoming IDocs to business objects, which are readily consumable by the client at the endpoint.	On receiving the IDoc packet from SAP, the adapter converts the IDocs to business objects, one by one, before sending each one to the endpoint.	On receiving the IDoc packet from SAP, the adapter converts the IDocs in the packet as one business object before sending it to the endpoint.	The incoming IDoc is only partially parsed (the control record of the IDoc is parsed but the data record is not). The client at the endpoint is responsible for parsing the data record.	not applicable
ALE pass-through IDoc	This interface wraps the raw incoming IDoc in a business object before delivering it to the client at the endpoint. The client is responsible for parsing the raw IDoc.	On receiving the IDoc packet from SAP, the adapter wraps each raw IDoc within a business object before sending the objects, one by one, to the endpoint.	On receiving the IDoc packet from SAP, the adapter wraps the raw IDoc packet in a business object before sending it to the endpoint.	Control record of the incoming IDoc is parsed. In case of the packet IDoc, only the first IDoc from the control record will be parsed.	The incoming IDoc would be sent as a HexBinary in Flat File format, where each record (control and data record) is separated with a delimiter.

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.
Log and trace files are in the `/profiles/profile_name/logs/server_name` path of the folder in which IBM Business Process Manager or WebSphere Enterprise Service Bus is installed.
2. The adapter relies on SAP JCo's retry handling to restart the JCo Server.
 - The adapter stops the server after the SAP JCo Server tries again for the specified `RetryLimit` value as specified in the activation specification.

Note: If the retry interval does not behave as specified in the activation specification, download the latest SAP JCo 3.0.6.
3. If all the retry attempts fail, the adapter logs the relevant message and CEI events, stops message end points and stops trying to recover the ALE event listener.

Note: You must restart the adapter or SCA application in this case.

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.

Note: You must restart the adapter or SCA application in this case.

Event recovery

You can configure the adapter for ALE inbound processing so that it supports event recovery in case of abrupt termination. When event recovery is specified, the adapter persists the event state in an event recovery table that resides on a data source. Event recovery is not enabled by default; you must specify it by enabling once-only delivery of events during adapter configuration.

Data source

Event recovery for ALE inbound processing requires that a JDBC data source be configured. You use the Process Administrative Console to configure the data source. You select a JDBC provider (for example, Derby) and then create a new data source.

Event recovery table

You can create the event recovery table manually, or you can have the adapter create the event table. The value of the EP_CreateTable configuration property 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 8. 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: <ul style="list-style-type: none"> • 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.

Table 8. Event recovery table fields (continued)

Table field name	Type	Description
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.

To use event recovery for multiple endpoints, you must configure a separate event recovery table for each endpoint, although you can use the same data source (for example, Derby) to hold all the event recovery tables.

IDoc status updates

To monitor 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 server. 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. Monitoring of IDocs applies only to inbound processing (when the IDoc is sent from the SAP server to the adapter).

The adapter updates a status IDoc (ALEAUD) and sends it to the SAP server.

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

The status codes and their associated text are configurable properties of the adapter, as specified in the activation specification properties and shown in the following list:

- Success Code
- Failure Code
- Success Text
- Failure Text

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 9. Inbound properties of the logical system partner profile

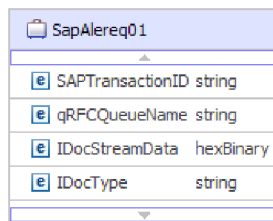
SAP property	Value
Basic Type	ALEAUD01
Logical Message Type	ALEAUD
Function module	IDOC_INPUT_ALEAUD
Process Code	AUD1

ALE pass-through IDoc 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. For inbound operation using pass-through IDocs, you can choose to create a parsed Control Record (Inbound only) along with a HexBinary stream to hold both the Control Record and the Data Record. For the first case, the business object will contain a child Business Object representing the parsed Control Record, and a HexBinary field to hold both the Data Record and Control Record. The same business object format is used for inbound and outbound ALE pass-through IDoc processing. For outbound operations using pass-through IDocs, the business object contains an IDoc stream instead of a child business object.

The control record will be generated using field names, by default.

The following figure illustrates a business object that represents a specific IDoc (Alereq01):



SapAlereq01	
<input type="checkbox"/>	SAPTransactionID string
<input type="checkbox"/>	qRFCQueueName string
<input type="checkbox"/>	IDocStreamData hexBinary
<input type="checkbox"/>	IDocType string

Figure 40. Example of an ALE pass-through business object

Parsed Control Record for Inbound

The business object contains a transaction ID, a queue name, stream data to hold the Data Record and Control Record, a child business object for the parsed Control Record and the IDoc type. The transaction ID (SAPTransactionID) is used to ensure once-only delivery of business objects, and the queue name (qRFCQueueName) specifies the name of the queue on the SAP server to which the IDocs should be delivered. If you are not using transaction IDs or queues, these properties are blank.

To parse the IDoc control record, select the checkbox in the configuration screen during the external service wizard run.

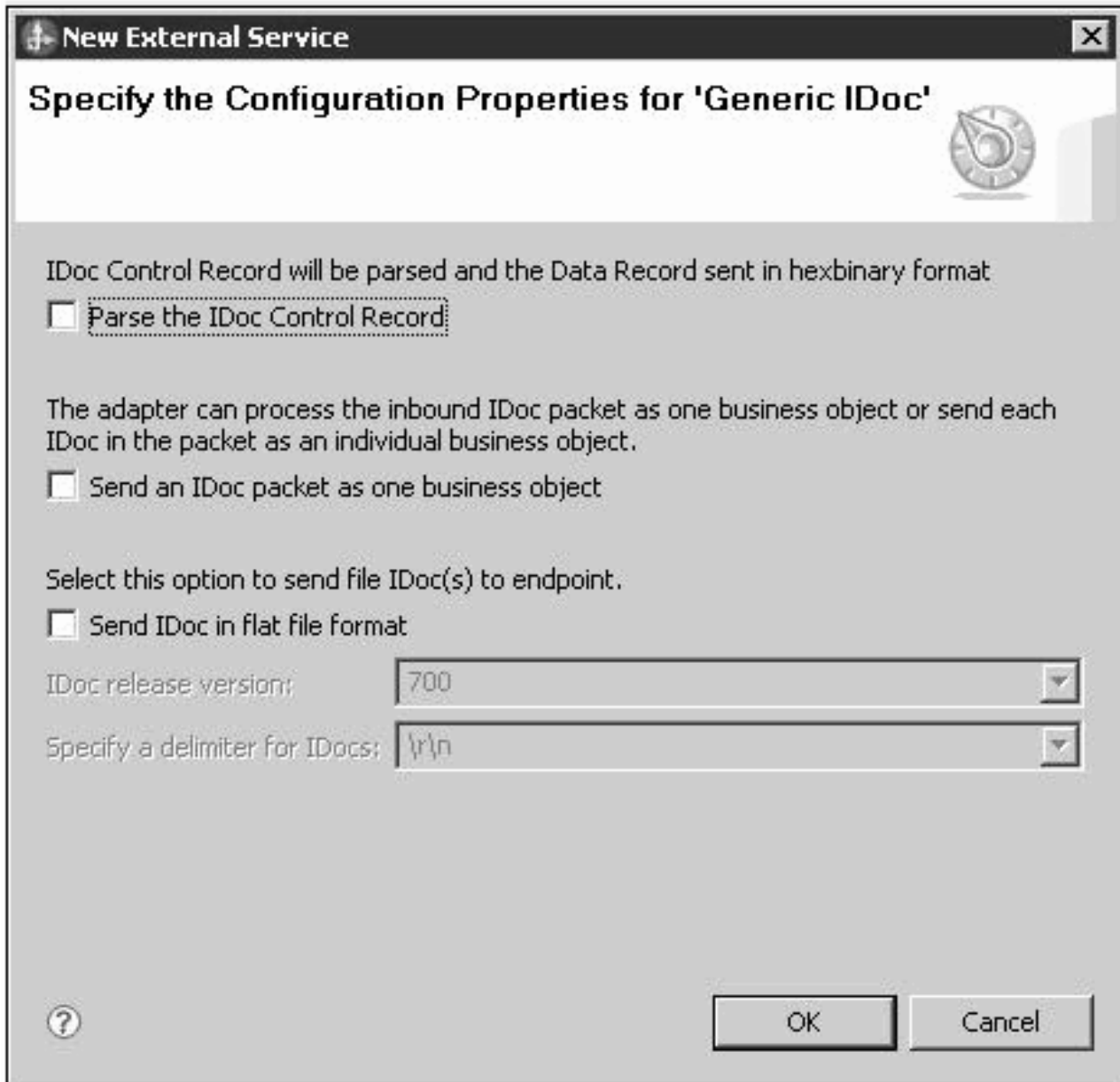


Figure 41. Configuration properties window

The following figure illustrates a business object for a generic IDoc when the **Parse the IDoc Control Record** is unchecked.

SapGenericIDocObject	
⊞	SAPTransactionID string
⊞	qRFCQueueName string
⊞	IDocStreamData hexBinary
⊞	IDocType string

Figure 42. Example of an ALE pass-through generic IDoc business object

When the **Parse the IDoc Control Record** checkbox is selected, a child business object is generated to hold the parsed Control Record.

The following figure illustrates a business object that represents a Generic IDoc when the checkbox is selected:

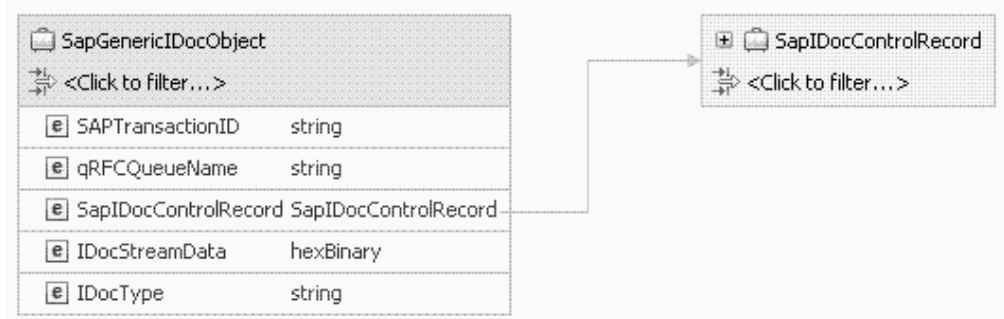


Figure 43. Example of a generic IDoc business object

Additional information about the business object can be found in the application-specific information of the business object. For example, the application-specific information for SapAleReq01 lists whether the IDoc packet is split and provides information about the type of object, which, for pass-through IDoc business objects is always PASSTHROUGHIDOC.

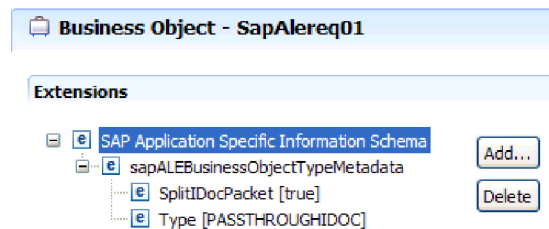


Figure 44. Application-specific information for an ALE pass-through IDoc business object

The ASI for the delimiter and the IDoc version will also be added if the Flat File option is enabled.

The Advanced event processing interface

The Advanced event processing interface of the WebSphere Adapter for SAP Software is used for both inbound and outbound processing. For inbound processing, it polls for events in SAP, converts them into business objects, and sends the event data as business objects to IBM Business Process Manager or WebSphere Enterprise Service Bus. For outbound processing, the adapter processes events sent from an application to retrieve data from or update data in the SAP server.

You can use the WebSphere BI Station tool to monitor inbound events.

Advanced event processing interface supports both inbound and outbound processing. For inbound, the adapter polls the SAP system for events and delivers the events to the endpoint. For this interface, the user needs to write custom ABAP handler on the SAP system. The ABAP handler is invoked by the adapter at run

time. This is the most complex interface to use. The application developer would use this interface if the other interfaces cannot provide the capability needed for the application that is being developed.

Note: You must select the non-transaction RAR, which is `CWYAP_SAPAdapter.rar`, when you configure the adapter to make use of the Advanced event processing interface.

Outbound processing for the Advanced event processing interface

During outbound processing, business object data is converted into an ABAP handler function, which is called on the SAP server. When the data is returned by the ABAP handler function, the data is converted to a business object, and the business object is returned as a response.

The following list describes the sequence of processing actions that result from an outbound request using the Advanced event processing interface.

1. The adapter receives the Advanced event processing business object, which contains business data along with the metadata.
2. The Advanced event processing interface of the adapter uses the metadata of the business object to obtain the type of IDoc specified and to reformat the business object data into the structure of that IDoc.
3. After it reformats the data, the adapter passes the business object data to an object-specific ABAP handler (based on the operation), which handles the integration with an SAP native API.
4. After the object-specific ABAP handler finishes processing the business object data, it returns the response data in IDoc format to the adapter, which converts it to the business object.
5. The adapter returns the results to the caller.

ABAP handler overview

An ABAP handler is a function module that gets data into and out of the SAP application database. For each business object definition that you develop, you must support it by developing a custom ABAP handler.

ABAP handlers reside in the SAP application as ABAP function modules. ABAP handlers are responsible for adding business-object data into the SAP application database (for Create, Update, and Delete operations) or for using the business-object data as the keys to retrieving data from the SAP application database (for the Retrieve operation).

You must develop operation-specific ABAP handlers for each hierarchical business object that needs to be supported. If you change the business-object definition, you must also change the ABAP handler.

The ABAP handler can use any of the SAP native APIs for handling the data. Some of the native APIs are listed below.

- Call Transaction
Call Transaction is the SAP-provided functionality for entering data into an SAP system. Call Transaction guarantees that the data adheres to the SAP data model by using the same screens an online user sees in a transaction. This process is commonly referred to as *screen scraping*.
- Batch data communication (BDC)

Batch Data Communication (BDC) is an instruction set that SAP can follow to process a transaction without user intervention. The instructions specify the sequence in which the screens in a transaction are processed and which fields are populated with data on which screens. All of the elements of an SAP transaction that are exposed to an online user have identifications that can be used in a BDC.

- **ABAP SQL**

ABAP SQL is the SAP proprietary version of SQL. It is database- and platform-independent, so that whatever SQL code you write, you can run it on any database and platform combination that SAP supports. ABAP SQL is similar in syntax to other versions of SQL and supports all of the basic database table commands such as update, insert, modify, select, and delete. For a complete description of ABAP SQL, see your SAP documentation.

Using ABAP SQL, an ABAP handler can modify SAP database tables with business object data for create, update, and delete operations. It can also use the business object data in the 'Where' clause of an ABAP select statement as the keys.

Note: Use of ABAP SQL to modify SAP tables is not recommended, because the integrity of the database might get corrupted . Use ABAP SQL only to retrieve data.

- **ABAP Function Modules and Subroutines**

From the ABAP handler, you can call ABAP function modules or subroutines that implement the required function.

The adapter provides the following tools to help in the development process:

- The adapter includes the Call Transaction Recorder Wizard to assist you in developing the ABAP handlers that use call transactions or BDC sessions.
- The external service wizard generates the required business objects and other artifacts for Advanced event processing. The business objects are based on IDocs, which can be custom or standard.
- The adapter provides samples that you can refer to for an understanding of the Advanced event processing implementation.

ABAP handler creation

For each IDoc object definition that you develop, you must support it by developing a custom ABAP handler.

You can use either standard IDocs or custom IDocs for the Advanced event processing interface. After defining the custom IDoc for an integration scenario, create an ABAP handler (function module) for each operation of the business object that needs to be supported.

Each function should have the following interface to ensure that the adapter can call it:

```
*" IMPORTING
*" VALUE(OBJECT_KEY_IN) LIKE /CWLD/LOG_HEADER-OBJ_KEY OPTIONAL
*" VALUE(INPUT_METHOD) LIKE BDWFAP_PAR-INPUTMETHOD OPTIONAL
*" VALUE(LOG_NUMBER) LIKE /CWLD/LOG_HEADER-LOG_NR OPTIONAL
*" EXPORTING
*" VALUE(OBJECT_KEY_OUT) LIKE /CWLD/LOG_HEADER-OBJ_KEY
*" VALUE(RETURN_CODE) LIKE /CWLD/RFCRC_STRU-RFCRC
*" VALUE(RETURN_TEXT) LIKE /CWLD/LOG_HEADER-OBJ_KEY
*" TABLES
*" IDOC_DATA STRUCTURE EDID4
*" LOG_INFO STRUCTURE /CWLD/EVENT_INFO
```

The following table provides information about the parameters:

Table 10. Interface parameters

Parameter	Description
OBJECT_KEY_IN	Should be no value.
INPUT_METHOD	Indicates whether the IDoc should be processed in a dialog (that is, through Call Transaction). Possible values are: " " - Background (no dialog) "A" - Show all screens "E" - Start the dialog on the screen where the error occurred "N" Default
LOG_NUMBER	Log Number.
OBJECT_KEY_OUT	Customer ID returned from the calling transaction.
RETURN_CODE	0 - Successful. 1 - Failed to retrieve. 2 - Failed to create, update, or delete.
RETURN_TEXT	Message describing the return code.
IDOC_DATA	Table containing one entry for each IDoc data segment. The following fields are relevant to the inbound function module: Docnum - The IDoc number. Segnam - The segment name. Sdata - The segment data.
LOG_INFO	Table containing details regarding events processed with either a success or error message.

Call Transaction Recorder wizard

The adapter provides the Call Transaction Recorder Wizard to assist you in developing the ABAP handlers that use call transactions or BDC sessions.

The Call Transaction Recorder wizard enables you to generate sample code for call transactions to facilitate development. It generates sample code stubs for each screen that is modified during the recording phase.

To access this wizard, enter the /CWLD/HOME_AEP transaction in the SAP GUI.

The following is sample code generated by the wizard. You can adopt this code in the ABAP Handler.

```
* Customer master: request screen chnge/dspl cent.
perform dynpro_new using 'SAPMF02D' '0101' .

* Customer account number
perform dynpro_set using 'RF02D-KUNNR' '1' .

* Function Command
perform dynpro_set using 'BDC_OKCODE' '/00' .

* Function Command
perform dynpro_set using 'BDC_OKCODE' '/00' .
```

```

* Customer master: General data, CAM address, communication
perform dynpro_new using 'SAPMF02D' '0111' .

* Title
perform dynpro_set using 'SZA1_D0100-TITLE_MEDI' 'Mr.' .

* Function Command
perform dynpro_set using 'BDC_OKCODE' '=UPDA' .

* Call Transaction
Call Transaction 'XD02' using bdcdata
    mode input_mode
    update 'S'
    messages into bdc_messages.

```

The wizard does not generate the required business object. You use the external service wizard to generate the business object.

Inbound processing for the Advanced event processing interface

The adapter uses the Advanced event processing interface to poll for events on the SAP server, to process the events, and to send them to an endpoint.

The following list describes the sequence of processing actions that result from an inbound request using the Advanced event processing interface.

1. A triggered event enters the event table with an initial status of pre-queued.
2. When the adapter polls for events, the status of the event changes from pre-queued to queued if there are no database locks for the combination of the user who created the event and the event key.
3. After the event is retrieved from the event table, the status of the event is updated to InProgress.

If locks exist, the status of the event is set to locked and the event is re-queued into the queue. Every event with a pre-queued or locked status is updated with every poll. You can configure the polling frequency using the Poll Frequency property.
4. After preprocessing all pre-queued events, the adapter selects the events.

The property Poll Quantity determines the maximum number of events returned for a single poll call.
5. For each event, the adapter uses the remote function specified for the Retrieve operation to retrieve the data and send it to the endpoint.

If the AssuredOnceDelivery property is set to true, an XID value is set for each event in the event store. After each event is picked up for processing, the XID value for that event is updated in the event table.

If the connection to the SAP system is lost or the application is stopped before the event is delivered to the endpoint, the event is consequently not processed completely. The XID column ensures that the event is reprocessed and sent to the endpoint. After the SAP connection is reestablished or the adapter starts up again, it first checks for events in the event table that have a value in the XID column. It then processes these events first and then polls the other events during the poll cycles.
6. After each event is processed, it is updated or archived in the SAP application.

When the event is processed successfully, it is archived and then deleted from the event table.

The adapter can also filter the events to be processed by business object type. The filter is set in the Event Filter Type property. This property has a comma-delimited list of business object types, and only the types specified in the property are picked for processing. If no value is specified for the property, no filter is applied and all the events are picked up for processing.

Event detection

Event detection refers to the collection of processes that notify the adapter of SAP application object events. Notification includes, but is not limited to, the type of the event (object and operation) and the data key required for the external system to retrieve the associated data.

Event detection is the process of identifying that an event was generated in the SAP application. Typically, adapters use database triggers to detect an event. However, because the SAP application is tightly integrated with the SAP database, SAP allows very limited access for direct modifications to its database. Therefore, the event-detection mechanisms are implemented in the application transaction layer above the database.

Adapter-supported event detection mechanisms

The four event-detection mechanisms supported by the adapter are described in the following list:

- Custom Triggers, which are implemented for a business process (normally a single SAP transaction) by inserting event detection code at an appropriate point within the SAP transaction.
- Batch programs, which involve developing an ABAP program containing the criteria for detecting an event.
- Business workflows, which use the object-oriented event detection capabilities of SAP.
- Change pointers, a variation of business workflows, which use the concept of change documents to detect changes for a business process.

All these event-detection mechanisms support real-time triggering and retrieval of objects. In addition, custom triggers and batch programs provide the ability to delay the retrieval of events. An event whose retrieval is delayed is called a future event.

Note: Each event detection mechanism has advantages and disadvantages that need to be considered when designing and developing a business object trigger. Keep in mind that these are only a few examples of event detection mechanisms. There are many different ways to detect events.

After you determine the business process to support (for example, sales quotes or sales orders) and determine the preferred event-detection mechanism, implement the mechanism for your business process.

Note: When implementing an event detection mechanism, it is a good idea to support all of the functionality for a business process in one mechanism. This limits the impact in the SAP application and makes event detection easier to manage.

See the related topics on implementing the event-detection mechanisms in the *Performing prerequisite tasks specific to an interface* section.

Event table

Events that are detected are stored in an SAP application table. This event table is delivered as part of the ABAP component. The event table structure is as follows.

Table 11. Event table fields

Name	Type	Description
event_id	NUMBER	Unique event ID that is a primary key for the table.
object_name	STRING	Business graph name or business object name.
object_key	STRING	Delimited string that contains the keys for the business object.
object_function	STRING	Operation corresponding to the event (Delete, Create, or Update).
event_priority	NUMBER	Any positive integer to denote the priority of the event.
event_time	DATE	Date and time when the event was generated.
event_status	NUMBER	Event processing status. Possible values are: 0 - Ready for poll 1 - Event delivered 2 - Event prequeued 3 - Event in progress 4 - Event locked -1 - Event failed
Xid	STRING	Unique XID (transaction ID) value for assured-once delivery.
event_user	STRING	User who created the event.
event_comment	STRING	Description of the event.

Event triggers

After an event is identified by one of the event-detection mechanisms, it is triggered by one of the adapter-delivered event triggers. Event triggers can cause events to be processed immediately or in the future.

The function modules that trigger events are described in the following list.

- /CWL/ADD_TO_QUEUE_AEP
This function module triggers events to the current event table for immediate processing.
- /CWL/ADD_TO_QUEUE_IN_FUT_AEP
This function module triggers events to the future event table to be processed at a later time.

Note: Both functions are for real-time triggering.

Current® event table

If the event is triggered in real-time, /CWL/ADD_TO_QUEUE_AEP commits the event to the current event table (/CWL/EVT_CUR_AEP). Specifically, it adds a row of data for the object name, verb, and key that represents the event.

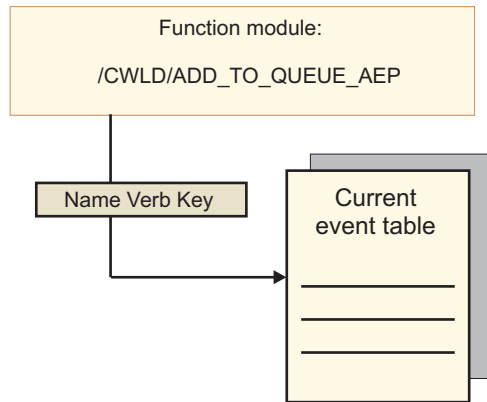


Figure 45. The function module adds a row of data to the current event table

Future event table

If an event needs to be processed at a future date, the processing described in the following list. Figure 46

1. A custom ABAP handler calls /CWLD/ADD_TO_QUEUE_IN_FUT_AEP with the event.
2. The /CWLD/ADD_TO_QUEUE_IN_FUT_AEP module commits the event to the future event table (/CWLD/EVT_FUT_AEP). Specifically, it adds a row of data for the object name, verb, and key that represents the event. In addition, it adds a Date row
3. The adapter-delivered batch program /CWLD/SUBMIT_FUTURE_EVENTS reads the future event table.
4. If scheduled to do so, the batch program retrieves events from the future event table.
5. After it retrieves an event, the batch program calls /CWLD/ADD_TO_QUEUE_AEP.
6. The /CWLD/ADD_TO_QUEUE_AEP module triggers the event to the current event table.

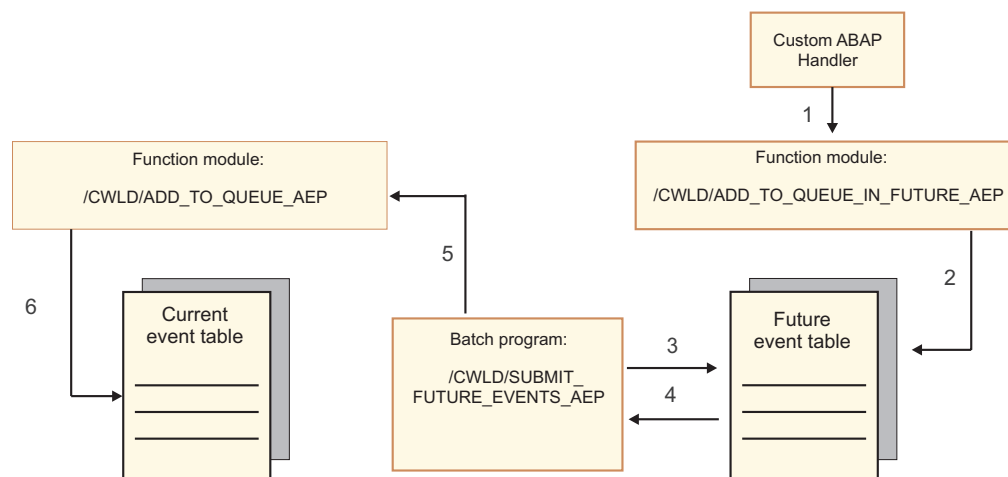


Figure 46. How an event is added to the future event table, retrieved from the table, and added to the current event table

/CWLD/ADD_TO_QUEUE_IN_FUT_AEP uses the system date as the current date when it populates the Date row of the future event table.

Event restriction

Use event restriction to filter out events that you do not want added to the event table. The adapter provides an ABAP include program (TRIGGERING_RESTRICTIONS_USER) that can be modified to filter events.

The, TRIGGERING_RESTRICTIONS_USER program is called from within the event trigger /CWLD/ADD_TO_QUEUE_AEP to enable additional filtering of events.

Note: You must have developer privileges to make changes because the code needs to be recompiled.

To view or modify the include program TRIGGERING_RESTRICTIONS_USER:

1. If IBM WebSphere BI Station is not currently displayed, enter transaction /n/CWLD/HOME_AEP.
2. Click the **Configuration** tab.
3. Click **Event Restriction**.

To upgrade an adapter-provided ABAP handler from one SAP R/3 version to another, check to see if changes were made to program TRIGGERING_RESTRICTIONS_USER. This program is intended for customer modification. If changes were made, you can avoid conflicts by downloading the custom work as text files, not as transport files, for use as a reference.

Upgrade any ABAP code from the old event restriction program to the new event restriction program.

Business objects for the Advanced event processing interface

During Advanced event processing, the adapter exchanges business objects with the SAP application. The business object represents a custom IDoc, or a standard or extension IDoc available on the SAP server

Business object structure

Note: For custom interfaces that you want to support, as a first step, you need to define the custom IDoc in the SAP system. You can then use the external service wizard to discover this custom IDoc and build the required artifacts, including the business-object definition.

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

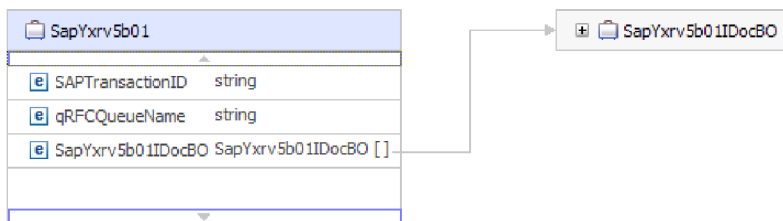


Figure 47. Example of an Advanced event processing wrapper business object

Note that the transaction ID and queue name attributes are present in the business object even if you are not using the tRFC or qRFC features.

The IDoc business object has the structure shown in the following figure.

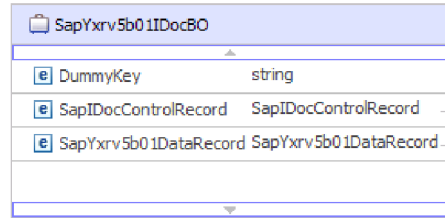


Figure 48. Example IDoc business object structure

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.

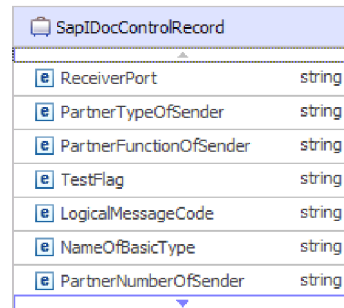


Figure 49. Example control record structure

- 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.

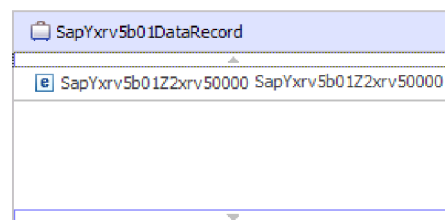


Figure 50. Example data record structure

- The business object data (which is pointed to from the data record) has the following structure:

SapYxrv5b01Z2xrv50000	
Function	string
Customernumber1	string
Title	string
Centralorderblockforcustomer	string

Figure 51. Example business object data

Additional information about the business object can be found in the application-specific information of the business object. For example, the application-specific information lists whether the IDoc packet is split and provides information about the operation.

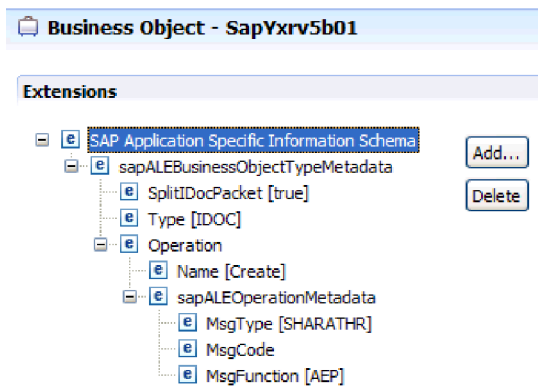


Figure 52. Application-specific information for the business object

Chapter 4. Samples and tutorials

To help you use WebSphere Adapters, samples and tutorials are available from the Business Process Management Samples and Tutorials website.

You can access the samples and tutorials in either of the following ways:

- From the welcome page of IBM Integration Designer, click **Go to Samples and Tutorials**. In the Samples and Tutorials pane, under More samples, click **Retrieve**. Browse the displayed categories to make your selection.
- From the Business Process Management Samples and Tutorials website:
<http://bpmwikilogin.blueworkslive.com/accounts/login/>.

Chapter 5. Configuring the module for deployment

To configure the adapter so that it can be deployed on IBM Business Process Manager or WebSphere Enterprise Service Bus, use IBM Integration Designer to create a module, which is exported as an EAR file when you deploy the adapter. You then specify the business objects you want to discover and the system on which you want to discover them.

Roadmap for configuring the module

Before you can use WebSphere Adapter for SAP Software in a runtime environment, you must configure the module. Understanding this task at a high level helps you perform the steps that are needed to accomplish the task.

You configure the module for the adapter to use by using IBM Integration Designer. The following figure illustrates the flow of the configuration task, and the steps that follow the figure describe this task at a high level only. See the topics that follow this road map for the details on how to perform each of these steps.

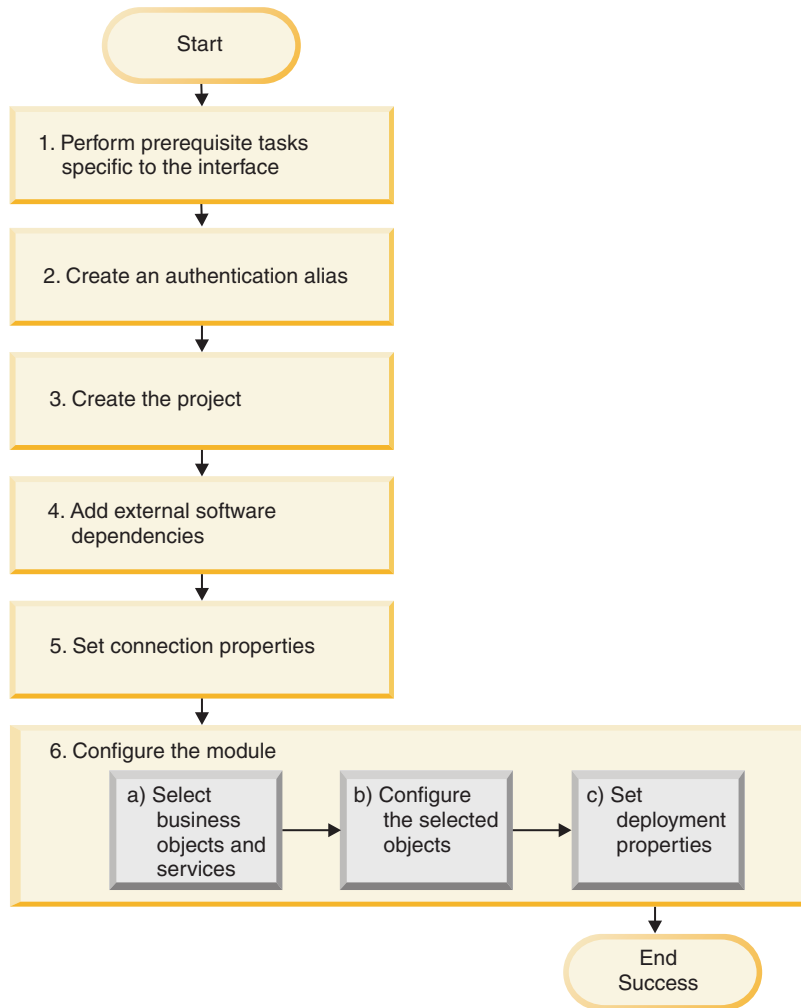


Figure 53. Roadmap for configuring the module

Configuring the module for deployment environment

This task consists of the following step:

1. Perform prerequisite tasks specific to the interface.
2. Create an authentication alias to access the SAP server with an encrypted password. This step is optional, depending on your policy for handling passwords and IDs. You perform this step using the server.
3. Create the project. First, start the external service wizard in IBM Integration Designer to begin the process of creating and deploying a module. The wizard creates a project that is used to organize the files associated with the module.
4. Add the external software dependencies required by WebSphere Adapter for SAP Software to the project. These dependencies are also required when you export the module as an EAR file, and deploy the EAR file to the server.
5. Set connection properties that the external service wizard needs to connect to the SAP server for discovery of objects and services.
6. Configure the module for inbound processing or outbound processing by using the external service wizard to find and select business objects and services from the SAP server, and to generate business object definitions and related artifacts.

- a. Select business objects and services for inbound or outbound processing from the business integration components discovered by the external service wizard.
- b. Configure the selected objects by specifying operations and other properties that apply to all the business objects.
- c. Set deployment properties that the adapter uses to connect to the SAP server at run time. Then, generate the service by using the external service wizard to save the new module, which contains the business object or objects you configured, the import file or export file, and the service interface.

Performing prerequisite tasks specific to an interface

Depending on the interface you will be using, you might have to perform some prerequisite tasks before you use the external service wizard to configure the module. For example, if you are configuring a module for ALE or BAPI inbound processing, you must register a program ID with the SAP server. If you are going to use the Advanced event processing interface, you must install transport files on the SAP server.

Configuring the SAP system to work with the adapter

Before you configure WebSphere Adapter for SAP Software for ALE inbound processing or for BAPI inbound processing, you must configure authorization profiles and register an RFC destination on the SAP server. For ALE processing, you must also configure a receiver port, logical system, distribution model, and partner profile on the SAP server. Contact your system administrator if you are not sure whether these items have been configured.

About this task

Perform the following steps on the SAP server using the SAP GUI.

Note: Only the first task is required for BAPI inbound processing.

Procedure

1. Access levels required for the user name that is used to connect to the SAP system.:
To run the SAP adapter smoothly, configure the following authorization profiles in the SAP system:

Table 12.

OBJECT	DESCRIPTION	AUTHORIZATION
B_ALE_RECV ALE/EDI:	Receiving IDocs via RFC	B_ALE_RC_ALL
S_CTS_ADAMI	Administration Functions in the Change and Transport S	S_CTS_IMPALL
S_RFCACL	Authorization Check for RFC User (for example, Trusted System)	S_RFCACL_ALL
S_TCODE	Authorization Check for Transaction Start	S_TCD_ALL
S_RFC	Authorization check for RFC access	S_RFC_ALL

Table 12. (continued)

OBJECT	DESCRIPTION	AUTHORIZATION
S_TABU_DIS	Table Maintenance (via standard tools such as SM30)	S_TABU_ALL
S_IDOCCTRL	WFEDI: S_IDOCCTRL - General Access to IDoc Functions	S_IDCCTR_AL+
S_IDOCDEFT	WFEDI: S_IDOCDEFT - Access to IDoc Development	S_IDCDFT_ALL

To identify which authorizations are absolutely necessary, perform the following steps:

- a. Open **TCode SM19** and use the Security Audit trace
- b. Run the SAP Adapter
- c. Refer to the system log **SM20** for authorization objects that are either accessed or denied
2. Register an RFC program ID:
 - a. Open transaction **SM59** (Display and Maintain RFC Destinations).
 - b. Click **Create**.
 - c. Type a name for the RFC destination.
 - d. In the **Connection Type** field, select **T**.
 - e. In the **Activation Type** field, select **Registered Server Program**.
 - f. Type a Program ID.
You will use this program ID when you configure the adapter. This value indicates to the SAP gateway which RFC-enabled functions the program ID listens for.
 - g. Under the **MDMP and Unicode** tab, set the RFC destination as **Unicode** or **non-Unicode** by choosing the appropriate radio button.
For error-free operation of the adapter while using multiple language settings, set the RFC destination as Unicode.
 - h. Save your entry.
3. Set up a receiver port (for ALE processing only):
 - a. Open transaction **WE21** (Ports in IDoc processing).
 - b. Select **Transactional RFC**, click **Ports**, and click the Create icon.
 - c. Type a name for the port and select **OK**.
 - d. Type the name of the destination you created in the previous task (or select it from the list).
 - e. Save your entry.
4. Specify a logical system (for ALE processing only):
 - a. Open transaction **BD54** (Change View Logical Systems).
 - b. Click **New Entries**.
 - c. Type a name for the logical system and click the Save icon.
 - d. If you see the Prompts for Workbench request, click the New Request icon. Then enter a short description and click the Save icon.
 - e. Click the Continue icon.
5. Configure a distribution model (for ALE processing only):
 - a. Open transaction **BD64** (Maintenance of Distribution Model).

- b. Click **Distribution Model > Switch processing model**.
 - c. Click **Create model view**.
 - d. Type a name for the model view and click the Continue icon.
 - e. Select the distribution model you created, and click **Add message type**.
 - f. For outbound processing, type the logical system name you created in the previous task as **Sender** and the logical name of the SAP server as **Receiver**. Then select a message type (for example, **MATMAS**) and click the Continue icon.
 - g. Select the distribution model again and click **Add message type**.
 - h. For inbound processing, type the logical name of the SAP server as **Sender** and the logical system name you created in the previous task as **Receiver**. Then select a message type (for example, **MATMAS**) and click the Continue icon.
 - i. Save your entry.
6. Set up a partner profile (for ALE processing only):
- a. Open transaction **WE20** (Partner Profiles).
 - b. Click the Create icon.
 - c. Type the name of the logical system you created in the earlier task and, for **Partner Type**, select **LS**.
 - d. For **Post Processing: permitted agent**, type **US** and your user ID.
 - e. Click the Save icon.
 - f. In the Outbound parameters section, click the Create outbound parameter icon.
 - g. In the Outbound parameters window, type a message type (for example, **MATMAS05**), select the receiver port you created in the earlier task, and select **Transfer IDoc immed**.
 - h. Click the Save icon.
 - i. Press F3 to return to the Partner Profiles view.
 - j. In the Inbound parameters section, click the Create inbound parameter icon.
 - k. In the Inbound parameters window, type a message type (for example, **MATMAS**), and a process code (for example, **MATM**).
 - l. Click the Save icon.
 - m. Press F3 to return to the Partner Profiles view.
 - n. In the Inbound parameters section, click the Create inbound parameter icon.
 - o. In the Inbound parameters window, type the following values: **ALEAUD** for **Message Type**, and **AUD1** for **Process Code**.
 - p. Click the Save icon.
 - q. Press F3 to return to the Partner Profiles view.
 - r. Click the Save icon.

Results

You have performed the tasks (on the SAP server) required to use the BAPI inbound interface or the ALE interface.

What to do next

Configure the adapter for the interface.

Creating the data source

To create a data source, which is used for event tracking and recovery during ALE inbound processing, you use the Process Administrative Console. You select a JDBC provider and then create a data source in the JDBC provider. After configuring the data source, you use the Test Connection button in the Process Administrative Console to test the database connection.

Before you begin

Before configuring the data source, make sure the database is already created and then configure the data source using that database.

About this task

You need a JDBC provider only if you are going to set up an event recovery table to persist inbound events (to ensure once-only delivery).

Procedure

1. In the Process Administrative Console, select a JDBC provider.
 - a. Click **Resources > JDBC > JDBC Providers**.
 - b. Select a JDBC provider.
2. Select **Data sources**.
3. Create a new data source by clicking **New**.
4. Type values for the required fields.
 - a. In the **Data source name** field, type the name of the event table.
A default value is provided. For example, for the Derby JDBC Provider, the default value is **Derby JDBC Driver DataSource**. You can change the default value.
An example of a data source name is: EventRecoveryDS
 - b. In the **JNDI Name** field, type the JNDI name of the data source.
An example is jdbc/EventRecovery.
5. Optionally, select an authentication alias for the JDBC provider from the **Component-managed authentication alias and XA recovery authentication alias** list.
6. Click **Next**.
7. In the Create a data source window, indicate the database to which the data source connects by typing a value in the **Database name** field.
8. Review the information in the Summary table to ensure its accuracy, and then click **Finish**.
9. Save your configurations.
10. From the list of data sources, select the check box next to the data source that you created in the previous steps.
11. Click **Test connection**.
You will see a message that the test was successful.

Note: If the test is not successful, make sure the database drivers are available in lib\ext directory. Also make sure that the database name and port are correct.

Results

A new data source is created.

What to do next

Configure the adapter for ALE inbound processing. Use the database JNDI created in this topic, and use the Auto create event table property to create the event recovery table.

Creating an IDoc definition file

When you configure the adapter for ALE processing, you typically have the external service wizard create a business-object definition based on the IDoc or IDocs it finds on the SAP system. Alternatively, you can have the external service wizard generate the business-object definition based on an IDoc definition file, which you create.

About this task

Use the following general procedure to create the IDoc definition file. Note that the steps for generating these definitions can vary from system release to release. For example, on some versions of the SAP server, you might need to clear the **IDoc record types** check box if it is checked.

Note: Follow this procedure only if you plan to use the **Discover IDoc from File** choice in the external service wizard. If you plan to use **Discover IDoc from System**, you do not need to create an IDoc definition file.

Procedure

1. In the SAP user interface, select transaction WE63 by entering /oWE63.
2. In the **Basic type** field, enter the basic IDoc type (for example, ALEREQ01) or browse to see a list of basic types.
3. Click **Documentation > Parser** or click the Parser icon.
The IDoc definition is displayed on the screen.
4. Save the definition to a directory on your local file system by clicking **System > List > Save > Local File**.
5. From the Save list in file window, select **unconverted** and select the check icon.
Note that **unconverted** is the only supported format.
6. Enter the location where the file should be saved (or browse to the location) and click **Generate**.

Results

The IDoc definition file is located on your local file system.

What to do next

Configure the adapter for ALE outbound or inbound processing.

Adding transport files to the SAP server

To use the Advanced event processing interface, you must first add the adapter-supplied transport files to the SAP server.

About this task

Note: This procedure is for the Advanced event processing interface only. If you are not using the Advanced event processing interface, skip this procedure.

The transport files for WebSphere Adapter for SAP Software contain a variety of objects, such as table structures, functions, and data. These development objects must be imported into the SAP server before you can use the advanced event processing interface.

The transport files are provided as .zip files in the IBM Integration Designer installation directory. The path to the files within that directory is ResourceAdapters\SAP_7.5.0.1_x>\transports. Change the version according to your current installation.

From within transports, the files are located in one of the following directories:

- transports_40_45_46, which you use with SAP version 4.0, 4.5, or 4.6
- transports_47_erp, which you use with SAP version 4.7 and above

Procedure

1. Create the namespace for the adapter before installing the transport files. Provide the following name for the namespace: /CWLD/
2. Import the transport files into the SAP server in the order shown:
 - a. CWYAP_SAPAdapter_AEPTransport_Infrastructure.zip
 - b. CWYAP_SAPAdapter_AEPTransport_Primary.zip

Results

The files needed to use Advanced event processing are installed on the SAP server.

What to do next

Configure the adapter for Advanced event processing.

Implementing event-detection mechanisms

When you use the Advanced event processing interface for inbound processing, you must determine an event-detection mechanism for the business process with which you are working. You then implement that process.

About this task

Note: These procedures are for the Advanced event processing interface only. If you are not using the Advanced event processing interface, skip the procedures.

Sample code and examples are provided to help you implement an event-detection mechanism.

Implementing custom triggers

Custom triggers requires encapsulating a portion of ABAP code in a custom function module. The event-detection code is written as a function module to ensure that the processing remains separate from the transaction. Any tables or variables used from the transaction must be passed to the function module by value and not by reference.

About this task

Note: This procedure is for the Advanced event processing interface only. If you are not using the Advanced event processing interface, skip this procedure.

To minimize the effects of locking a business object when retrieving an event, the function module typically executes in an update-task mode. To avoid inconsistencies, do not use update task if the function module is already being called within a process that is in an update-task mode.

To minimize the impact in the transaction, place the function module within another include program. Using an include program allows you to make changes to custom code rather than to SAP code.

The event-detection code contains logic that identifies the object for the event. For example, the sales order transaction handles many types of orders, but only one order type is required. This logic is in the event-detection code. The general strategy for placing this event-detection code is to insert it just before the data is committed to the database. The function module containing the event detection code is typically created as a part of the function group for the business object.

To implement a custom trigger for event detection:

Procedure

1. Determine which verbs to support: Create, Update, or Delete. This helps define which transactions to investigate.
2. Determine the business-object key for the transaction. This key must be unique to allow the adapter to retrieve the business object from the database.
If a composite key is required, at triggering time you can specify each key attribute and its corresponding value as a name-value pair. When the business object is created at polling time, the adapter automatically populates the attributes with their values.
3. Check that an SAP-provided user exit in the transaction has all the information needed to detect an event.
For example, a user exit might not be able to implement a Delete verb because the business object is removed from the database before that point.
4. If a user exit cannot be used, determine the appropriate location for the event-detection code, and then add the event-detection code using an SAP modification. Select a location that has access to the business object key and other variables used to make the decision. If you are implementing the future events capability, in addition to adding the event-detection code for future events, contact your Basis administrator to schedule the adapter-delivered batch program /CWLD/SUBMIT_FUTURE_EVENTS to run once every day.
5. Research a business process by looking for a “commit work statement” in the code executed by the transaction for the business process. You can use the ABAP debugger to investigate the value of different attributes at that point.
6. Determine the criteria for detecting an event.
7. Create the function module containing the event detection code.
8. Create the include program and then add it to the transaction's code.
9. Test all of the scenarios designed to detect an event.

Example

Example

The following steps describe the process of creating an example SAP customer master using the custom trigger event-detection mechanism. The code that follows is a result of this process.

1. Upon investigation of the SAP customer master transaction, transaction XD01 is found to support the customer master creation business process.
2. The Customer number is determined to be the unique key. The Customer number is stored in table/field KNA1-KUNNR.

Note: Because this event uses a single unique key, the code example uses the OBJKEY parameter to pass the key value.

3. Transaction XD01 has a user exit in the transaction flow as part of the document save process (Form Userexit_save_document). At this point in the transaction, the customer number is available when the user exit is executed.
4. An include statement is added to the user exit that points to the include program.
5. At this time, the include program and a function module must be created.

The following code fragment illustrates the function call to the /CWLD/ADD_TO_QUEUE_AEP event trigger (using a single key value).

```
CASE HEADER_CHANGE_IND.
  WHEN 'I'.
    * The verb will always be a create if KNA1 data is entered.
    IF KNA1_CREATE = 'X'.
      HEADER_EVENT = C_CREATE_EVENT.
    ELSE.
      * Check if an entry is in the config table for converting a create. If
      * no entry is found, the default is to convert the extension of sales
      * area or company code to an update.
      SELECT SINGLE * FROM /CWLD/CONF_VAL
        WHERE CONF_NAME = C_CONVERT_CREATE
          AND CONF_VALUE = C_FALSE_WORD.

      IF SY-SUBRC = 0.
        HEADER_EVENT = C_CREATE_EVENT.
      ELSE.
        HEADER_EVENT = C_UPDATE_EVENT.
      ENDIF.
    ENDIF.

  WHEN 'U'.
    HEADER_EVENT = C_UPDATE_EVENT.
  WHEN 'E' OR 'D'.
    HEADER_EVENT = C_DELETE_EVENT.
ENDCASE.

* See if it's a sold-to company.
SELECT SINGLE * FROM /CWLD/CONF_VAL
  WHERE CONF_NAME = C_AGCUSTOMASTER
    AND CONF_VALUE = KNA1-KTOKD.

* clear temp_obj_type.
CLEAR TEMP_OBJ_NAME.
IF SY-SUBRC = 0.
  * temp_obj_type = 'YXR_V51'.
  TEMP_OBJ_NAME = C_OBJ_CUSTOMERMMASTER.
ELSE.
```

```

* If it's not a sold-to company, check if it's another partner.
SELECT SINGLE * FROM /CWL/CONF_VAL
  WHERE CONF_NAME = C_AGCUSTPARTNER
  AND CONF_VALUE = KNA1-KTOKD.
ENDIF.

CALL FUNCTION '/CWL/ADD_TO_QUEUE_AEP'
  EXPORTING
    OBJ_NAME = TEMP_OBJ_NAME
    OBJKEY = OBJKEY
    EVENT = HEADER_EVENT
  * IDOC_NUMBER =
    GENERIC_RECTYPE = GENERIC_RECTYPE
  IMPORTING
    RECTYPE = RECTYPE
  TABLES
    EVENT_CONTAINER = EVENT_CONTAINER
  EXCEPTIONS
    OTHERS = 1.

```

The following code fragment illustrates the function call to the /CWL/ADD_TO_QUEUE_IN_FUT_AEP event trigger (single key value).

```

DATA: DATE_IN_FUTURE LIKE SY_DATUM.

CALL FUNCTION '/CWL/ADD_TO_QUEUE_IN_FUT_AEP'
  EXPORTING
    OBJ_NAME = TEMP_OBJ_NAME
    OBJKEY = OBJKEY
    EVENT = HEADER_EVENT
    VALID_DATE = DATE_IN_FUTURE
  IMPORTING
    RECTYPE = RECTYPE
  TABLES
    EVENT_CONTAINER = EVENT_CONTAINER
  EXCEPTIONS
    OTHERS = 1.

```

What to do next

Configure the adapter for Advanced event processing.

Implementing batch programs

To implement batch program as an event detection mechanism, you must write an ABAP program that evaluates database information. If the criteria in the ABAP program is fulfilled when the program executes, then an event is triggered.

About this task

Note: This procedure is for the Advanced event processing interface only. If you are not using the Advanced event processing interface, skip this procedure.

To implement batch program for event detection:

Procedure

1. Determine which verb to support: Create, Update, or Delete.
2. Determine the business object key for the transaction.

The business object key must be unique so that the business object can be retrieved from the database. A composite key might be required.
3. Determine the criteria for detecting an event.

You should have knowledge of the database tables associated with a business object.

4. Create an ABAP program containing the criteria for generating an event.
5. If you are implementing the future events capability, in addition to adding the event-detection code for future events, contact your Basis administrator to schedule the adapter-delivered batch program /CWLD/SUBMIT_FUTURE_EVENTS to run once every day.
6. Determine if a background job is required to automate the batch program.
A background job is useful if there is an impact on system resources, which makes it necessary to run the batch program during off-peak hours.

Example

Example

The following steps describe the process of creating a batch program that detects events for all sales quotes created on today's date. The code that follows is a result of this process.

1. Create is determined to be the supported verb.
2. The quote number is determined to be the unique key used to retrieve the events.
3. The creation date (VBAK-ERDAT) and the document category (VBAK-VBTYP) must be checked.

The following sample code supports the SAP sales quote as a batch program:

```
REPORT ZSALESORDERBATCH.
tables: vbak.

parameter: d_date like sy-datum default sy-datum.

data: tmp_key like /CWLD/LOG_HEADER-OBJ_KEY,
      tmp_event_container like swcont occurs 0.

" retrieve all sales quotes for today's date

" sales quotes have vbtyp = B

select * from vbak where erdat = d_date and vbtyp = 'B'.

tmp_key = vbak-vbeln.

CALL FUNCTION '/CWLD/ADD_TO_QUEUE_AEP'
  EXPORTING
    OBJ_NAME = 'SAP4_SalesQuote'
    OBJKEY = tmp_key
    EVENT = 'Create'
    GENERIC_RECTYPE = ''
  IMPORTING
    RECTYPE = r_rectype
  TABLES
    EVENT_CONTAINER = tmp_event_container.

write: / vbak-vbeln.

endselect.
```

What to do next

Configure the adapter for Advanced event processing.

Implementing business workflows

Business workflow is a set or sequence of logically related business operations. The processing logic within a workflow detects events. The business workflow event-detection mechanism relies on the SAP Business Object Repository (BOR), which contains the directory of objects along with their related attributes, methods, and events.

About this task

Note: This procedure is for the Advanced event processing interface only. If you are not using the Advanced event processing interface, skip this procedure.

To implement business workflow for event detection:

Procedure

1. Determine which SAP business object represents the functionality that you need. Check if the events trigger, start, or end a workflow.
You can use the Business Object Builder (transaction SWO1) to search for the appropriate business object.
2. Create a subtype of this SAP business object.
A subtype inherits the properties of the supertype and can be customized for use.
3. Activate the events (such as CREATED, CHANGED, and DELETED) for the business object by customizing the subtype.

Example

Example

The following example of SAP sales quote can be used to implement an event trigger using business workflow:

1. Search the BOR for the appropriate sales quote business object. A search can be done using the short description field and the string '*quot*'. BUS2031 (Customer Quotes) is one of the business objects returned.
2. Upon further investigation of BUS2031, it is determined that the key field is CustomerQuotation.SalesDocument (VBAK-VBELN).
3. A subtype for BUS2031 is created using the following entries:
 - Object type—ZMYQUOTE
 - Event—SAP4_SalesQuote
 - Name—SAP4 Sales Quote
 - Description—Example of an SAP 4 Sales Quote Subtype
 - Program—ZMYSALESQUOTE
 - Application—V
4. The event detection mechanism is activated by adding an entry to the Event Linkage table (transaction SWE3). The create event is activated using the following entries:
 - Object type—ZMYQUOTE
 - Event—SAP4_SalesQuote
 - Receiver FM— /CWLD/ADD_TO_QUEUE_DUMMY_AEP
 - Receiver type FM— /CWLD/ADD_TO_QUEUE_WF_AEP

Note: The Receiver and Receiver type function modules (FM) point to /CWLD/ADD_TO_QUEUE_AEP. The DUMMY function module is used only because sometimes the SAP application requires that both fields be populated. The WF function module translates the SAP standard interface to the one used by /CWLD/ADD_TO_QUEUE_AEP.

The business workflow event-detection mechanism is created and active. It is set up to detect all SAP Customer Quotes that are created.

What to do next

Configure the adapter for Advanced event processing.

Implementing change pointers

A change pointer uses change documents and is one of the more challenging event detection mechanisms to implement. The SAP Business Object Repository (BOR) is used as well as Application Link Enabled (ALE) technology. A change document always refers to a business document object having at least one database table assigned to it. If the data element in a table is marked as requiring a change document and the table is assigned to a business document object, then a change in value of the field defined by the data element generates a change document. The changes are captured in tables CDHDR and CDPOS and are used for event detection.

About this task

Note: This procedure is for the Advanced event processing interface only. If you are not using the Advanced event processing interface, skip this procedure.

To implement change pointer for event detection:

Procedure

1. Activate the global Change pointers flag in transaction BD61.
2. Change the SAP function module CHANGE_POINTERS_CREATE to include the function module call to /CWLD/EVENT_FROM_CHANGE_POINTR.
3. Determine which verbs to support: Create, Update, or Delete.
4. Check if the SAP business process (transaction) utilizes change documents:
 - In the Environment menu for the transaction, does a Change function exist? How about when you click Go To, and then click Statistics?
 - If you change data in the transaction, is there a new entry in table CDHDR that reflects the change?
 - In the database tables associated with a transaction, do any of the data elements have the Change Document flag set?
5. If the answer is Yes to any of these questions, the transaction uses change documents.
 - a. Determine if the data elements that set the Change Document flag capture all of the information needed to detect an event. Changing the Change Document flag is not recommended because it changes an SAP-delivered object.
 - b. Determine the business object key for the transaction. The business object key must be unique so that the business object can be retrieved from the database. A composite key may be required. This is normally table/field CDHDR-OBJECTID.

- c. Determine the criteria for detecting an event. Use table/field CDHDR-OBJECTCLAS as the main differentiator. CDPOS-TABNAME may also be used to detect the event.
- d. Update function module /CWLD/EVENT_FROM_CHANGE_POINTER with the logic to detect the event.

Example

Example

The following example of an SAP sales quote can be used to implement an event trigger using change pointer:

1. Update is determined to be the supported verb. Investigating the sales quote create transaction shows that the Create verb is not detected through this mechanism.
2. When performing the checks of the business for sales quote:
 - The Change function is available from the Environment menu in transaction VA22.
 - Making a change to a sales quote results in a new entry in table CDHDR.
 - Looking at table VBAP, the field ZMENG has the Change Document flag set.
3. No evaluation of data elements was done for this example.
4. The sales quote number is determined to be the unique key in CDHDR-OBJECTID.
5. CDHDR-OBJECTCLAS has a value of VERKBELEG, which is the main differentiator. Only sales quotes should be picked up. The code checks the TCODE field in the header table, but a proper lookup should be done in the VBAK table.

The following sample code is added to /CWLD/
EVENT_FROM_CHANGE_POINTER:

```
when 'VERKBELEG'.
  data: skey like /cwld/log_header-obj_key,
        s_event like swetypecou-event,
        r_genrectype like swetypecou-rectype,
        r_rectype like swetypecou-rectype,
        t_event_container like swcont occurs 1 with header line.

" Quick check. Should check document category (VB TYP) in VBAK.
check header-rcode = 'VA22'.

" Event detection has started
perform log_create using c_log_normal c_blank c_event_from_change_pointer c_blank.

" Set the primary key
skey = header-objectid.

" Set the verb
s_event = c_update_event.

" Log adding the event to the queue
perform log_update using c_information_log text-i44
'SAP4_SalesQuote' s_event skey.

" Event detection has finished.
perform log_update using c_finished_log c_blank
c_blank c_blank c_blank.

call function '/CWLD/ADD_TO_QUEUE_AEP'
```

```
exporting
  obj_name = 'SAP4_SalesQuote'
  objkey = skey
  event = s_event
  generic_rectype = r_genrectype
importing
  rectype = r_rectype
tables
  event_container = t_event_container
exceptions
  others = 1.
```

What to do next

Configure the adapter for Advanced event processing.

Creating an authentication alias

An authentication alias is a feature that encrypts the password used by the adapter to access the SAP server. The adapter can use it to connect to the SAP server instead of using a user ID and password stored in an adapter property.

Before you begin

To create an authentication alias, you must have access to the administrative console. The following procedure shows you how to gain access to the administrative console through IBM Integration Designer. If you are using the administrative console directly (without going through IBM Integration Designer), log in to the administrative console and skip to step 2.

About this task

Using an authentication alias eliminates the need to store the password in clear text in an adapter configuration property, where it might be visible to others.

To create an authentication alias, use the following procedure.

Procedure

1. Start the administrative console.
To start the administrative console through IBM Integration Designer, perform the following steps:
 - a. In the Business Integration perspective of IBM Integration Designer, click the **Servers** tab.
 - b. If the server does not show a status of **Started**, right-click the name of the server (for example, **IBM Business Process Manager or WebSphere Enterprise Service Bus**) and click **Start**.
 - c. Right-click the name of the server and click **Run Process Administrative Console**.
 - d. Log on to the administrative console. If your administrative console requires a user ID and password, type the ID and password and click **Log in**. If the user ID and password are not required, click **Log in**.
2. In the Process Administrative Console, click **Security > Secure administration, applications, and infrastructure**.
3. Under **Authentication**, click **Java Authentication and Authorization Service > J2C authentication data**.

4. Create an authentication alias.
 - a. In the list of J2C authentication aliases that is displayed, click **New**.
 - b. In the **Configuration** tab, type the name of the authentication alias in the **Alias** field.
 - c. Type the user ID and password that are required to establish a connection to the SAP server.
 - d. Optionally type a description of the alias.
 - e. Click **OK**.

The newly created alias is displayed.

The full name of the alias contains the node name and the authentication alias name you specified. For example, if you create an alias on the node `widNode` with the name `ProductionServerAlias`, then the full name is `widNode/ProductionServerAlias`. This full name is the one you use in subsequent configuration windows.

- f. Click **Save**, and then click **Save** again.

Results

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

Creating the project

To begin the process of creating and deploying a module, you start the external service wizard in IBM Integration Designer. The wizard creates a connector project, which is used to organize the files associated with the module.

Before you begin

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

About this task

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

Procedure

1. To start the external service wizard, go to the Business Integration perspective of IBM Integration Designer, and then click **File > New > External Service**. Click **Next**.
2. In the Select the Service Type or Registry window, expand the **Adapters** node, select **SAP**, and then click **Next**.
3. In the Select an Adapter window, select **IBM WebSphere Adapter for SAP Software (IBM : *version*)**, where *version* is the version of the adapter you want to use, and then click **Next**.
4. In the Import a RAR File window, accept the default project name in the **Connector project** field or type a different name.
5. In the **Target runtime environment** field, select the type of server where you want to deploy the module. The wizard creates the artifacts that are appropriate to that server.

6. Click **Next**. The Locate the Required Files and Libraries window is displayed.

What to do next

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

Adding external software dependencies

As part of generating the service, you are prompted by the external service wizard to specify the location of the required sapjco3.jar file and related files.

About this task

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

Procedure

1. Obtain the sapjco3.jar file and the associated files for your operating system from your SAP administrator or from the SAP Web site. Obtain the CWYAP_SAPAdapterExt.jar from the adapter package. The files are listed in the following table.

Table 13. External software dependency files required by SAP Software

Operating system	Files to be copied
Windows	sapjco3.jar, sapidoc3.jar, and any *.dll files that come with the SAP JCo download from the SAP website
UNIX (including UNIX System Services on z/OS®)	sapidoc3.jar, sapjco3.jar and any .so and .o files that come with the SAP JCo download from the SAP website

2. SAP JCo3 requires msvcp71.dll and msucr71.dll on Windows environment. These dlls are found in the system32 directory on most Windows systems. Copy these dlls onto your Windows environment if it does not have them.
3. Add the CWYAP_SAPAdapterExt.jar from the SAP adapter package.
4. From the Required Files and Libraries window, specify the location of the files:
 - a. For each file, click **Browse** and select the location of the file.

Note: You are prompted for the location of the .dll files only if they are not already located in the Windows system path.

- b. Click **Next**.

Results

The sapjco3.jar file and associated files are now part of your project.

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 external service wizard can establish a connection to the server.

Setting connection properties for the external service wizard

To set connection properties for the external service wizard so that it can access the SAP server, specify information such 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 software dependency files (the sapjco3.jar and associated files).

About this task

Specify the connection properties that the external service wizard needs to establish a connection to the SAP server and discover functions or data.

To specify the connection properties, use the following procedure.

Procedure

1. From the Select the Processing Direction window, Select **Inbound** (if you are going to send data from the SAP server) or **Outbound** (if you are going to send data to the SAP server) and click **Next**. The Specify the Discovery Properties window is displayed.
2. From the Specify the Discovery Properties window, specify the configuration properties:
 - a. In the **Host name** field, type the name (or IP address) of your SAP server.
 - b. Optionally, change the default value for **System number**.
 - c. Type your client ID (or use the default value if your client ID is 100).
 - d. If necessary, change the default setting for **Language code** by clicking **Select** and selecting a value from the list.

The default value in the **Code page** field is related to the value in the **Language code** field. For example, if the language code is EN (English), the code page number is 1100. If you change the language code to TH (Thai), the code page number changes to 8600.

Note: You will only get a listing of IDocs in the language selected. The error messages will be displayed in the specified language.

- e. Type the name and password you use to access the SAP server.
The password is case-sensitive.
- f. Choose an interface from the following **SAP interface name** options:
 - Advanced Event Processing (AEP), ALE, ALE pass-through IDoc or BAPI for Inbound, or

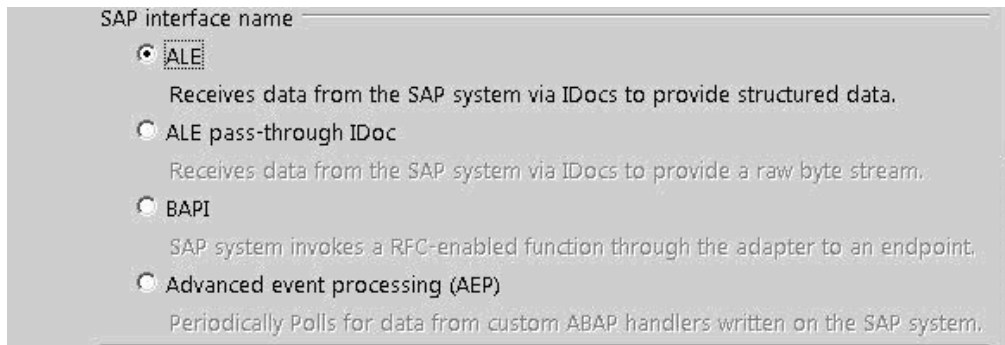


Figure 54. SAP interfaces - inbound

- Advanced Event Processing (AEP), ALE, ALE pass-through IDoc, BAPI, BAPI work unit, BAPI result set or Query interface for SAP Software for Outbound

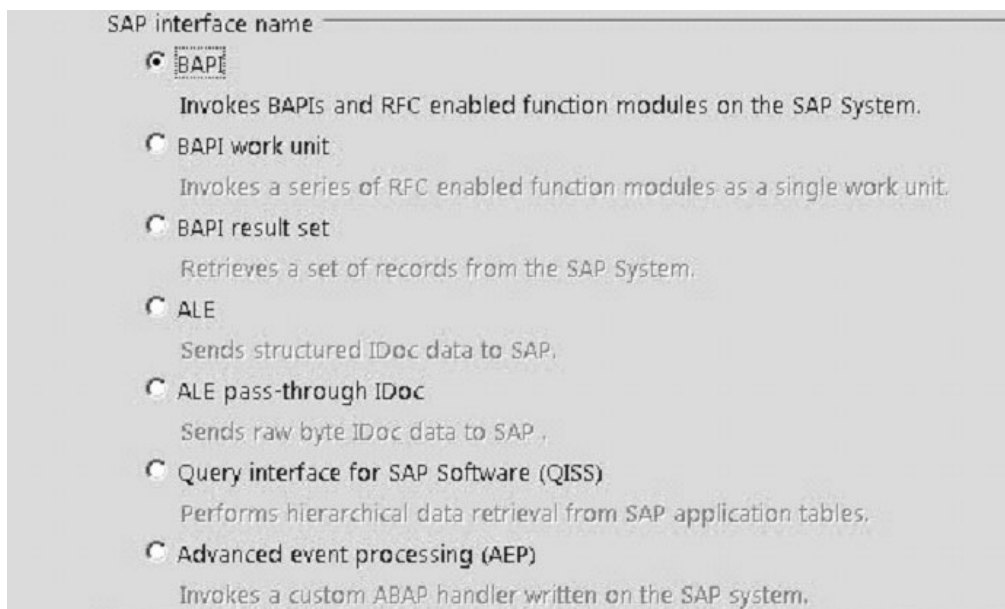


Figure 55. SAP interfaces - outbound

Note: This option is not available if you are modifying existing artifacts.

3. Optional: To set additional advanced properties, click **Advanced**.
4. To set RFC tracing properties, perform the following steps:
 - a. Expand **SAP RFC trace configuration** and select **RFC trace on**.
 - b. Select the **Write JCo traces into adapter logs** checkbox to enable JCo API traces to be generated in the broker trace file. These traces will then be interleaved with the adapter traces. Each JCo API trace logged into the broker trace file will have the string [JCoAPI]. All JCo API traces are logged at the INFO message log level.
 - c. Select a tracing level from the **RFC trace level** list.
 - d. Click **Browse** and select a location to which the RFC trace files will be saved.

Note: Enabling RFC tracing in turn enables CPIC tracing as well. CPIC tracing cannot be disabled individually.

Note: To avoid tracing files getting created in multiple locations, create an environment variable (for example RFC_TRACE_DIR) and set it to a valid folder.

If a value is set for the environment variable RFC_TRACE_DIR, all RFC will be generated in the folder set in the environment variable RFC_TRACE_DIR

All CPIC traces will be generated in the current working directory.

If the environment variable RFC_TRACE_DIR is not set, traces generated are saved under the folder <IID_ROOT><RAD_ROOT>.. This is the top-level folder of your IID installation. For example, C:\IBM\IID7.5

5. Optional: To enable bidirectional support for the adapter at run time:
 - a. In the **Bidi properties** area, select **Bidi transformation**.
 - b. Set the ordering schema, text direction, symmetric swapping, character shaping, and numeric shaping properties to control how bidirectional transformation is performed.
6. To configure Business Object name generated by the adapter for the BAPI Modules, expand **Business Object naming configuration** and select the required options from the following:
 - a. Check the **Generate business objects according to SAP naming conventions** checkbox to have the adapter to generate all the Business Object names same as that of the SAP XI standard. This option is applicable only for BAPI outbound processing.
 - b. By default, the **Enforce the same naming convention for business objects** checkbox is selected to have the adapter generate all the Business Object names without appending any hashcode. The hashcode is appended to the namespace of each Business Object instead of the name.
 - c. Do not check the **Enforce the same naming convention for business objects** checkbox to have the adapter append the hashcode to the subsequent Business Object that has the same name. This is done to avoid duplication.

For more information on naming conventions, refer to “Naming conventions for BAPI business objects” on page 315

7. Optional: To change the location of the log files for the wizard or the amount of information included in the logs, click **Change the logging properties for the wizard**, and then provide the following information:
 - a. In **Log file output location**, specify the location of the log file for the wizard.
 - b. In **Logging level**, specify the severity of errors that you want logged.

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

8. Click **Next**.

Results

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

What to do next

Specify search criteria that the external service wizard uses to discover functions or data on the SAP server.

Configuring the module for outbound processing

To configure a module to use the adapter for outbound processing, use the external service wizard in IBM Integration Designer to find and select business objects and services from the SAP server, and to generate the business object definitions and related artifacts.

Before you begin

The following table outlines the properties that are applicable for an interface:

Table 14.

Interface	SAP RFC Trace Configuration	Bidi properties	Business Objects naming convention	
			SAP naming convention	Enforce naming convention for business objects
BAPI	Yes	Yes	Yes	Yes
BAPI Work unit	Yes	Yes	No	Yes
BAPI Result Set	Yes	Yes	No	Yes
ALE	Yes	Yes	Yes	No
ALE Pass-through Idoc	Yes	Yes	No	No
Query Interface for SAP Software (QISS)	Yes	Yes	No	No
Advanced Event Process (AEP)	Yes	Yes	No	No

Authentication using connection specification properties

WebSphere Adapter for SAP Software uses connection properties either through Managed Connection Factory properties or a Java Authentication and Authorization Services (JAAS) alias. If you want to change the connection properties used for authentication with either one of these authentication methods, you can change the connection properties through the IBM Business Process Manager administrative console and restart the Java EE application or change the JAAS security settings.

In addition to the methods explained, the connection parameters can also be specified through the ConnectionSpec properties. The ConnectionSpec properties are used by an application component to pass connection-related properties.

You can specify the relevant ConnectionSpec properties for the outbound request. If you specify both ConnectionSpec properties and Managed Connection Factory properties during run time, the adapter uses the values specified in the ConnectionSpec properties to create a connection and ignores the values in the Managed Connection Factory properties.

The following table lists the ConnectionSpec properties for various authentication mechanisms:

Table 15. ConnectionSpec properties

Authentication mechanism	Properties
Basic	User Name, Password
Single Sign On (SSO)	ssoLogonTicket
X509 Certificate	x509cert

The following table lists the properties available to enable Secure Network Connection (SNC):

Table 16. Secure Network connection properties

Authentication mechanism	Properties
Secure Network Connection (SNC)	sncMyname, sncPartnername, sncQop, sncMode, sncLib

Refer to “Enable Secure Network Connection” on page 344, for more information.

Note: The SSO token / X509 authentication mechanism can also be accomplished by setting the username to "\$X509CERT\$" (for X509 certificate based authentication) and "\$MYSAPSSO2\$" (for SSO token based authentication) and setting the password to a base 64 encoded string, which comprises of a SSO token / X509 certificate.

To configure the adapter to create an SAP server connection, see “Passing the connection parameters dynamically.”

Passing the connection parameters dynamically

To pass the connection-related properties dynamically as part of the outbound request you must configure the connection specification class name and set the connection properties on the business graph.

Before you begin

1. WebSphere adapter for SAP import interface, for example, **SAPOutboundInterface**, must be created for the required outbound operations by running the external service wizard.
2. The input data type for each of the outbound operation must be configured to use the business graph of the business object. For example, the input data type of the operations can be SapBapiCustomerGetdetailBG or SapBapiCustomerCreateBG.

The business graph implementation has a child business object, 'properties' defined as an element in the business graph schema definition. The connection properties must be set in the dataobject 'properties' of the business graph.

About this task

To pass the connection-related properties dynamically as part of the outbound request, follow this procedure.

Procedure

1. Configure the ConnectionSpec class name in the SAP Import created.
 - a. Right-click the SAP adapter import in the assembly diagram and select **Show in > Properties view**.
 - b. In the Properties tab, select **Binding > End-point Configuration**.
 - c. In the Connection Spec properties tab, select ConnectionSpec class name as `com.ibm.j2ca.sap.SAPConnectionSpec`
2. Set the **Resource authentication** field in Security Attributes to Application.
 - a. Select **Security Attributes** from Binding properties.
 - b. Set the **Resource authentication** property to Application from Advanced properties. The default value is Container.

When the Resource Authentication property is set to Application, the Java EE component runs a programmatic sign-on to the SAP server. The application component passes security information, such as user name and password, through the ConnectionSpec instance.

3. Set the **Connection properties** in the BusinessGraph within the properties child business object.

For the adapter to accept the connection parameters dynamically during an outbound request, the application component must set the connection parameters on the business graph data object of the business object.

The connection properties set on the business graph are prefixed as "CS" to identify them as ConnectionSpec properties. For example, you can set the user name and password to 'CSusername' and 'CSpassword' in the properties element of the BusinessGraph to set the values of connection properties.

Note: The host name, protocol, or port number values are not accepted through the ConnectionSpec properties. The adapter accepts only authentication-related properties of the user, such as user name, password, SSO token, X509 certificate or SNC properties.

Results

The connection parameters are configured.

What to do next

Create an interface and a Java component, and then deploy the application onto the IBM Business Process Manager.

Configuring a module for the BAPI interface

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

Selecting business objects and services for BAPI outbound processing

To specify the BAPI function or functions that you want to call and the data you want to process, you provide information in the external service wizard.

Before you begin

Ensure that you have set the connection properties for the external service wizard.

About this task

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

Procedure

1. In the Find Objects in the Enterprise System window, indicate which BAPI or set of BAPIs you want to work with.

- a. Click **RFC** to enable the filter  icon.
- b. Click the filter button. The Filter Properties window is displayed.

Note: Instead of using the filter capability, you can expand **RFC** and select the function from the list, or you can expand **BOR**, expand the functional grouping (for example, **Cross-Application Components**), and select the BAPI. Then skip ahead to step 4.

2. From the Filter Properties window, specify information about the BAPI or BAPIs you want to discover:

- a. Select **Discover objects by name** or **Discover objects by description** from the **Object attribute to use for discovery** list.
- b. 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.

- c. Indicate the number of functions you want returned by changing the default value in the **Maximum number of objects to retrieve** field or by accepting the default value.
 - d. Click **OK**.
3. Select the BAPI or BAPIs.
 - a. Expand **RFC (filtered)** to show the objects that match the search criteria of BAPI_CUSTOMER*.
 - b. From the Discovered object list, select one or more BAPIs that you want to use.
 4. Click the arrow button to add the BAPIs to the **Selected objects** list.
 5. In the Configuration Properties window, perform the following steps for each BAPI to add it to the list of business objects to be imported:
 - a. Optionally, select the **Use SAP field names to generate attribute names of business objects** check box to generate business object attribute names

using the original SAP field names without casing. If you want to generate business object attribute names using the original SAP field name casing, check the **Use the original character case of SAP field names to generate business object attribute names** check box. By default (when the **Use SAP field names to generate attribute names of business objects** check box is not selected), field descriptions are used to generate properties.

Note: This option is not available when you select the **Generate business objects according to SAP naming conventions** option in the Specify the Discovery Properties window.

- b. Optionally, select the **Process Date fields as String** check box. This allows the adapter to retrieve SAP-specific date formats which are otherwise non-standard in Java by using the String type instead of the Date type format.

Note: To use this feature, ensure that the above check box is selected while running the external service wizard. As the date fields are converted to string type, ensure that all downstream dependencies (if any) are taken care of.

- c. If the BAPI has optional parameters associated with it, select the **Select optional parameters to include as child objects** check box, expand **Optional parameters**, and select the type of parameters (import, export, or table) that you want to work with.

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

If you are adding BAPI_CUSTOMER_GETDETAIL function module, you have the option of adding the following parameters:

Optional Import Parameters

PI_PASS_BUFFER
PI_DIVISION
PI_DISTR_CHAN

Optional Export Parameters:

PE_ADDRESS
RETURN

Refer to the SAP documentation for a list and description of the optional parameters.

- d. 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 the left arrow.

- 6. Click **Next**.

Results

The external service 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. The Specify Composite Properties window is displayed.

What to do next

Provide information about the business object (such as the name of the top-level object and associated operation).

Configuring simple BAPI objects

To configure simple BAPI business objects, you specify information about the object (such as the name of the object and the operation associated with the object). If you are using the version of the adapter with transaction support, you also select the type of remote function call you want to make (**Synchronous RFC**, **Asynchronous transactional RFC**, or **Asynchronous queued RFC**).

Before you begin

If you want to use the **Asynchronous Transactional RFC** or **Asynchronous Queued RFC** choice, you must have installed IBM WebSphere Adapter for SAP Software with transaction support (CWYAP_SAPAdapter_Tx).

If you are sending the function call to a queue on the SAP server (so that an application on SAP server can process the BAPIs SAP server in order), make sure you have configured the queue on the SAP server.

About this task

To configure the business object, use the following procedure.

Procedure

Note: For BAPI outbound processing, when you select the **Generate business objects in SAP XI standard** option in the Specify Discovery Properties window, the **Generate BAPI business objects within a wrapper** and **Business object namespace** options will not be displayed in the Specify Composite Properties window. The adapter will use the SAP XI standard business namespace in this case. Hence, go to Step 4 on page 116.

1. In the Configure Composite Properties window, select a name for the top-level business object.
2. If you do not select the **Generate BAPIs within Wrappers** check box, top-level business objects are automatically generated for each BAPI selected. For each top-level business object generated, the adapter internally assigns the **Execute** operation to it. There is no limit on the number of BAPI's that you can configure.

If you select the **Generate BAPIs within Wrappers** check box, top-level business objects are generated that contain a child business object for each BAPI selected. You can configure a maximum number of four BAPI's.

Perform one of the following sets of tasks if you have selected **Generate BAPIs within Wrappers** check box:

- If you are working with a single BAPI, click **Add**, select an operation (for example, **Retrieve**), and click **OK**.
You can select only one operation per BAPI.
- If you are working with multiple BAPIs, select, for each operation, the BAPI you want associated with that operation, as described in the following steps:
 - a. Click **Add**, select the operation (for example, **Create**) from the list, and click **OK**.

- b. From the **RFC function for selected operation** list, select a BAPI to associate with the operation you selected in the previous step.
- c. For the second BAPI, click **Add**, select an operation (for example, **Retrieve**) from the list, and click **OK**.
- d. From the **RFC function for selected operation** list, select a BAPI to associate with the operation you selected in the previous step.
- e. For any subsequent BAPIs, repeat the previous two steps.

You can select only one operation per BAPI.

3. In the **Business object 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 external service wizard), change the namespace value.

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

4. To indicate where the business object information should be stored, type the path to the location in the **Folder** field. This is an optional step.

Note: This field cannot be edited if you are modifying existing artifacts.

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

5. If you want the BAPI or BAPIs to be enclosed within a business graph, leave **Generate a business graph for each business object** selected. Otherwise, remove the check.
6. If you are using the version of the adapter with transaction support, you can select the type of remote function call you want to make.

Note: If you are using the version of the adapter without transaction support (CWYAP_SAPAdapter), this step does not apply. The BAPI or BAPIs are sent synchronously. Skip ahead to step 7 on page 117

If you are using the version of the adapter with transaction support (CWYAP_SAPAdapter_Tx) but do not select a type of remote function call, the default (**Synchronous RFC**) is used. In Synchronous RFC, the adapter calls the BAPI and then waits for the response from the SAP server.

- a. Select the arrow next to the **SAP Remote Function Call (RFC) type** list.
- b. Select one of the RFC types:
 - Select **Synchronous RFC** (the default) if you want the BAPI sent in a synchronous manner (the adapter calls the BAPI and then waits for the response from the SAP server). Note that the receiving system must be available when you use **Synchronous RFC**.
 - Select **Asynchronous Transactional RFC** if you want the call to succeed regardless of whether the receiving system (the SAP server) is available.
 - If the event is successful, the adapter sends the transaction ID to the client.

- If the event fails, the adapter returns an `AbapException` with the transaction ID to the adapter client. The adapter client can use this transaction ID to make the call again at a later time.

Note: When you use **Asynchronous Transactional RFC**, no data is returned to the client from the adapter.

- Select **Asynchronous Queued RFC** if you want the BAPI or BAPIs delivered to a predefined queue on the SAP server. After you select **Asynchronous Queued RFC**, select, from the list, the specific queue on the SAP server to which the BAPI or BAPIs will be delivered.

If no queues exist on the SAP server, you can type the name of a queue. You can then create the queue on the SAP server after configuration.

Note: If you do not select a queue, the adapter will configure the object to use the **Asynchronous Transactional RFC** type.

7. If you want to continue to process a BAPI even if the BAPI return object contains errors, select the **Ignore errors in BAPI Return object** check box.

Note: If you selected **Asynchronous Transactional RFC** or **Asynchronous Queued RFC**, this check box is not available.

8. To enable the adapter to wait until all time critical updates to SAP database are completed before calling commit on it, select the **Wait until commit call to SAP database is completed and returned** check box. This option is only available if you are using the `CWYAP_SAPAdapter.rar`.
9. Click **Finish**.

Results

You specified a name for the top-level business object, selected an operation for the BAPI or BAPIs, and indicated the type of remote function call. The Service Generation and Deployment Configuration window is displayed.

What to do next

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

Setting deployment properties and generating the service

To generate the module, which is the artifact that is deployed on IBM Business Process Manager or WebSphere Enterprise Service Bus, you create the module, associate the adapter with 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 Specify the Service Generation and Deployment Properties window should be displayed.

If you choose to use X509 certificate authentication (or) Single Sign On authentication (or) Secure Network Communication (SNC) to authenticate with SAP using the adapter, the target SAP Server has to be configured for the selected authentication mechanism. You must also obtain the required information for configuring the authentication mechanism from SAP (for example, SSO token, X509 certificate, SNC name, and SNC partner name).

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.

Procedure

1. 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**.
2. Indicate whether you will use an authentication alias (instead of typing a user ID and password) to establish a connection to the SAP server:
 - To specify an authentication alias, leave **Specify a Java Authentication and Authorization Services (JAAS) alias security credential** selected. Then, in the **J2C Authentication Data Entry** field, enter the name you specified in the Security section of the administrative console.
If you are not going to use an authentication alias, clear **Specify a Java Authentication and Authorization Services (JAAS) alias security credential**.
 - To use a user ID and password, select **Using security properties from the managed connection factory**. Type in the user name and password to specify your security credentials.
 - To use other security mechanisms native to the enterprise system or if security is not required, select **Other**
3. Select **With module for use by single application** to embed the adapter files in a module that is deployed to the application server, or select **On server for use by multiple applications** to install the adapter files as a stand-alone adapter.

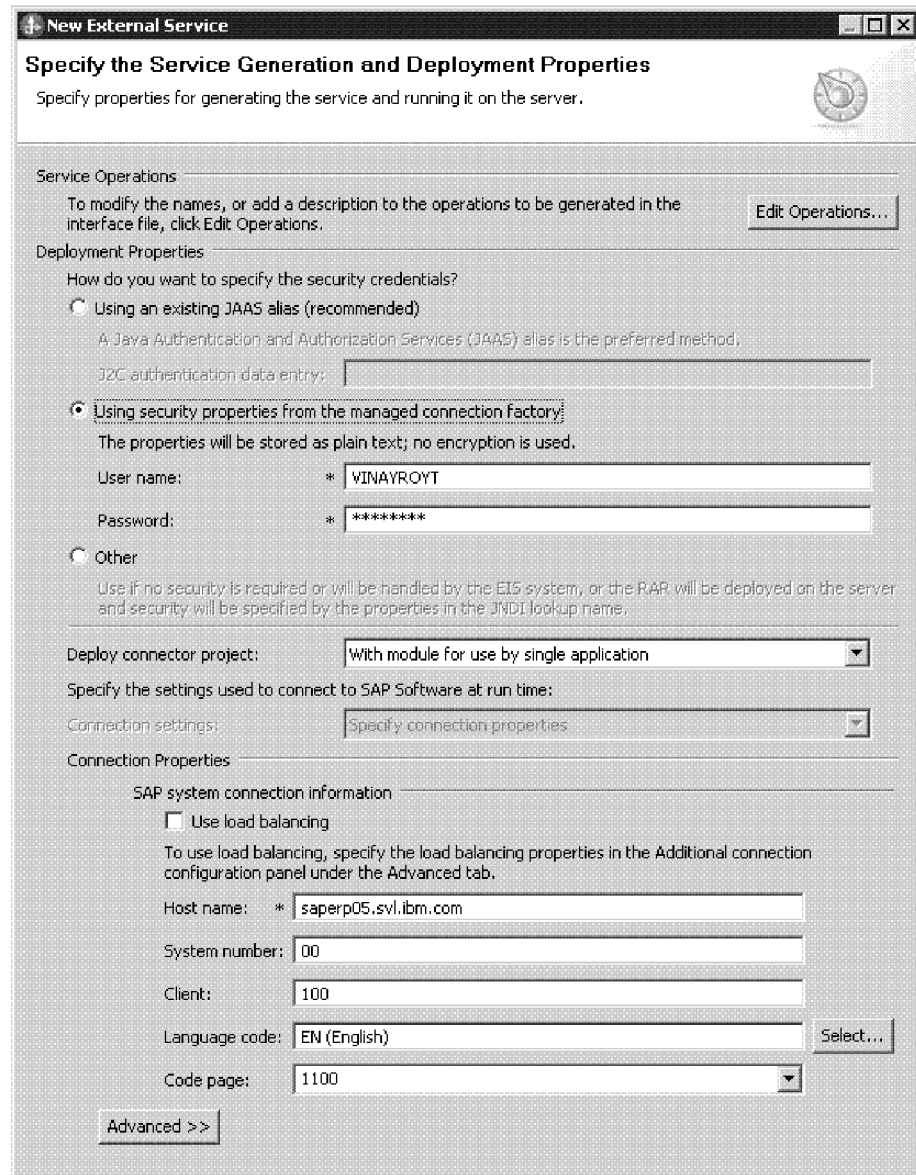


Figure 56. Connection properties

- **With module for use by single application:** With the adapter files embedded in the module, you can deploy the module to any application server. Use an embedded adapter when you have a single module using the adapter or if multiple modules need to run different versions of the adapter. Using an embedded adapter enables you to upgrade the adapter in a single module without the risk of destabilizing other modules by changing their adapter version.
- **On server for use by multiple applications:** If you do not include the adapter files in a module, you must install them as a stand-alone adapter on each application server where you want to run the module. Use a stand-alone adapter when multiple modules can use the same version of the adapter and you want to administer the adapter in a central location. A stand-alone adapter can also reduce the resources required by running a single adapter instance for multiple modules.

4. If you selected **On server for use by multiple applications** in the previous step, the **Connection properties** list becomes active. Make one of the following selections:

- Select **Specify connection properties** if you want to provide configuration information now. Then continue with step 5.
- Select **Use predefined connection properties** if you want to use a connection factory configuration that already exists.

If you decide to use predefined connection properties, you must ensure that your resource adapter name matches the name of the installed adapter, because this is how the adapter instance is associated with these properties. If you want to change the resource adapter name in the import or export, use the assembly editor in IBM Integration Designer to change the value in the import or export.

When you select **Use predefined connection properties**, the **JNDI Lookup Name** field is displayed in place of the properties.

- a. Type a value for **JNDI Lookup Name**.
- b. Click **Next**.
- c. Go to step 7 on page 122.

5. In the Connection properties section, set or change any connection properties that apply to your configuration.

Notice that some of the values are already filled in. For example, the values that you used in the Specify the Discovery Properties window (such as the **Host name**) are filled in.

Select the **Use load balancing** check box to use load balancing to connect to the SAP system. Load balancing properties which are **Message server host**, **Logon group name** and **SAP system ID** have to be specified in the Additional connection configuration panel under the Advanced tab.

See “Managed connection factory properties” on page 334 for more information about these properties.

Properties marked with an asterisk (*) are required.

6. To set additional properties, click **Advanced**.

a. Expand **Processing methods for illegal XML characters**

1) From the **Illegal XML character processing** drop-down list, select one of the options that correspond to your requirement:

- **Do not validate illegal XML characters**, if you want to work with the default adapter behavior.
- **Throw exception if contents in the BO contains illegal XML characters**, if you want to proceed after you get an exception message, and log the illegal XML characters in the trace file, at runtime.
- **Discard all illegal XML characters and related logs**, if you want the adapter to discard the illegal XML characters, and log the characters in the trace file, at runtime.

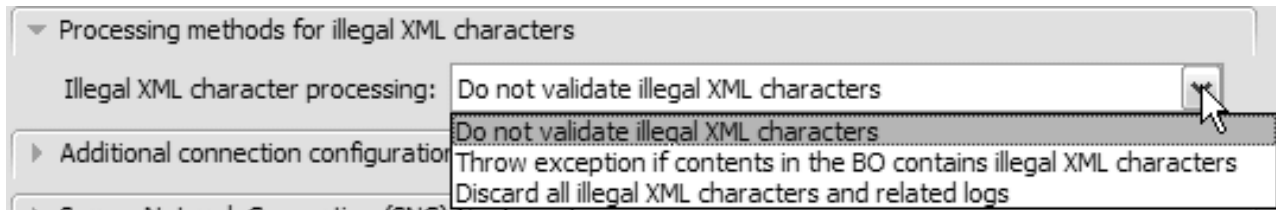


Figure 57. Processing methods for illegal XML characters

- b. Optionally expand **Additional connection configuration** and provide values (or change the default values) for the fields in this section of the window. For example, if your SAP configuration uses load balancing, provide values for the **Message server host** field, **Logon group name**, and **SAP system ID**. Select the **Reset the JCO client after closing connection handle** check box if during an outbound transaction, you want the adapter to make sure changes on the SAP EIS are reflected in the client. To enable the adapter to wait until all time critical updates to SAP database are completed before calling commit on it, select the **Wait until commit call to SAP database is completed and returned** check box. This option is only available if you are using the CWYAP_SAPAdapter_Tx.rar.
- c. Optionally, expand **Advanced Security Configuration** to specify your security mechanism. You can enable both **Secure Network Connection (SNC)** and **Single Sign On (SSO)**.
 - If you are using Secure Network Connection, select **Enable Secure Network Connection**. Enter information in the associated fields (name, partner, security level, and library path). If you select X509 certificate based authentication, the X509 certificate name has to be entered. Secure Network Connection has to be enabled to use X509 certificate based authentication.
 - If you are using Single Sign On authentication, select **Enable Single Sign On (SSO)**. Enter the SSO Logon Ticket in the field.
- d. Optionally to set RFC and JCo tracing properties, perform the following steps:
 - 1) Expand **SAP RFC trace configuration** and select **RFC trace on**.
 - 2) Select the **Write JCo traces into adapter logs** checkbox to enable JCo API traces to be generated in the broker trace file. These traces will then be interleaved with the adapter traces. Each JCo API trace logged into the broker trace file will have the string [JCoAPI]. All JCo API traces are logged at the INFO message log level.
 - 3) Select a tracing level from the **RFC trace level** list.
 - 4) Click **Browse** and select a location to which the RFC trace files will be saved.
- e. Optionally expand **Logging and tracing** and specify the following information:
 - If you have multiple instances of the adapter, set **Adapter ID** to a value that is unique for this instance.
 - If you want to mask sensitive information in log and trace files (for example, if you want to avoid making customer information visible in these files), select **Disguise user data as "XXX" in log and trace files** check box.

See “Managed connection factory properties” on page 334 for more information about these optional properties.

7. Click **Next**. The Specify the Location Properties window opens.
8. Create a module.
 - a. In the Specify the Location Properties window, click **New** in the **Module** field.
 - b. In the Integration Project window, click **Create a module project** or **Create a mediation module project** and click **Next**.
9. In the Module window, perform the following tasks:
 - a. Type a name for the module.
As you type the name, it is added to the workplace specified in the **Location** field.
This is the default location. If you want to specify a different location, remove the check from **Use default location** and type a new location or click **Browse** and select the location.
 - b. Specify whether you want to open the module in the assembly diagram (for module projects) or whether you want to create a mediation flow component (for mediation module projects). By default, these choices are selected.
 - c. Click **Finish**.
10. In the Specify the Location Properties window, perform the following steps:
 - a. If you want to change the default namespace, clear the **Use default namespace** check box and type a new path in the **Namespace** field.
 - b. 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. This is an optional step.
 - c. Optionally change the name of the interface.
The default name is SAPOutboundInterface. You can change it to a more descriptive title if you prefer.
 - d. If you want to save the business objects so that they can be used by another application, select **Save business objects to a library** and then select a library from the list or click **New** to create a new library.
 - e. Optionally type a description of the module.
11. Click **Finish**.

Results

The new module is added to the Business Integration perspective.

What to do next

Export the module as an EAR file for deployment.

Configuring a module for the BAPI work unit interface

To configure a module to use the adapter for BAPI work unit processing, you use the external service wizard in IBM Integration Designer to find a set of BAPIs. You then configure the business objects that are generated and create a deployable module.

Selecting business objects and services for BAPI work unit processing

To specify which BAPI functions you want to call and which data you want to process, you provide information in the external service wizard.

Before you begin

Make sure you have set the connection properties for the external service wizard.


About this task

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

To specify the search criteria and select the BAPI functions for the work unit, use the following procedure.

Procedure

1. In the Specify the Discovery Properties window, indicate which BAPI you want to work with.

- a. Click **RFC** to enable the filter  button.
- b. Click the filter button.

Note: Instead of using the filter capability, you can expand **RFC** and select the function from the list, or you can expand **BOR**, expand the functional grouping (for example, **Cross-Application Components**), and select the BAPI. Then skip ahead to step 4.

2. From the Filter Properties window, specify information about the BAPIs you want to discover:

- a. Select **Discover objects by name** or **Discover objects by description** from the **Object attribute to use for discovery** list.
- b. 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.

- c. Indicate the number of functions you want returned by changing the value in the **Maximum number of objects to retrieve** field or by accepting the default value.
 - d. Click **OK**.
3. Select the BAPIs.
 - a. Expand **RFC (filtered)**.
 - b. From the Discovered object list, select one or more BAPIs that you want to use.
 4. Click the arrow button to add the BAPIs to the **Selected objects** list.
 5. In the Configuration Properties window, perform the following steps for each BAPI to add it to the list of business objects to be imported:
 - a. Optionally, select the **Use the original character case of SAP field names to generate business object attribute names** check box. By default (when the check box is not selected), the field descriptions are used to generate properties.
 - b. Optionally, select the **Process Date fields as String** check box. This allows the adapter to retrieve SAP-specific date formats which are otherwise non-standard in Java by using the String type instead of the Date type format.

Note: To use this feature, ensure that the above check box is selected while running the external service wizard. As the date fields are converted to string type, ensure that all downstream dependencies (if any) are taken care of.

- c. If the BAPI has optional parameters associated with it, select the **Select optional parameters to include as child objects** check box, expand **Optional parameters**, and select the type of parameters (import, export, or table) that you want to work with.

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

If you are adding BAPI_CUSTOMER_GETDETAIL function module, you have the option of adding the following parameters:

Optional Import Parameters:

PI_PASS_BUFFER
PI_DIVISION
PI_DISTR_CHAN

Optional Export Parameters:

PE_ADDRESS
RETURN

Refer to the SAP documentation for a list and description of the optional parameters.

- d. 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 the left arrow.

6. Click **Next**.

Results

The external service wizard has returned the functions that match the search criteria, and you have selected the functions you want to work with. The Specify Composite Properties window is displayed.

What to do next

Provide information about the business objects (such as the name of the top-level object and associated operation).

Configuring BAPI work unit objects

To configure a BAPI work unit business object, you specify information about the object (such as the name of the object, the operations associated with the BAPIs in the work unit, and the sequence in which you want the BAPIs to be processed).

Before you begin

Make sure you have selected and imported the BAPI functions.

About this task

To configure the business object, use the following procedure.

Procedure

1. In the Specify Composite Properties window, select a name for the top-level business object.
2. Associate an operation with each BAPI and specify the order in which the BAPIs should be processed:
 - a. Click **Add**, select an operation (for example, **Create**), and click **OK**.
 - b. In the **Sequence of RFC functions for the selected operation** section of the window, indicate the order in which the BAPIs should be processed by clicking **Add**, selecting the BAPI that should be processed first, and clicking **OK**.
 - c. For each subsequent BAPI in the transaction, click **Add**, select the BAPI, and click **OK**.
 - d. After you have added all the BAPIs, click **Add**, select **COMMIT**, and click **OK**.

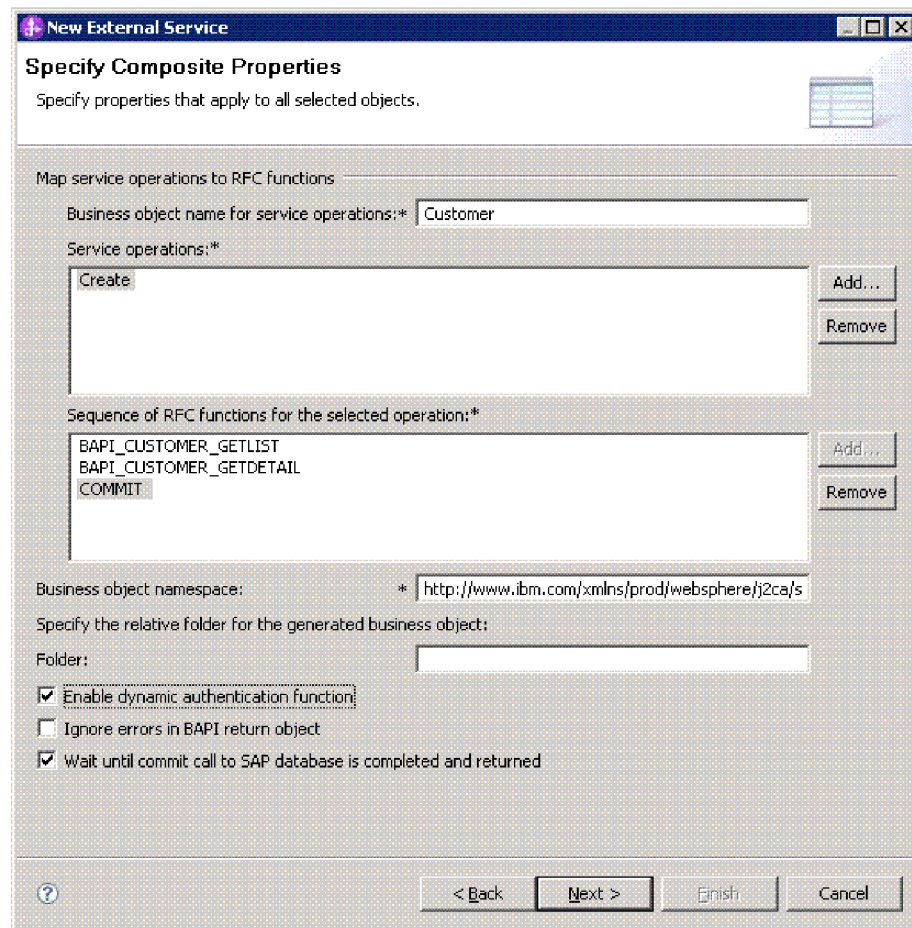


Figure 58. The Specify Composite Properties window after you have selected the BAPIs and the COMMIT operation

3. In the **Business object 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 external service wizard), change the namespace value.

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

4. To indicate where the business object information should be stored, type the path to the location in the **Folder** field. This is an optional step.
5. If you want the BAPI or BAPIs to be enclosed within a business graph, leave **Generate a business graph for each business object** selected. Otherwise, remove the check.
6. If you want to continue to process a BAPI even if the BAPI return object contains errors, select the **Ignore errors in BAPI Return object** check box.
7. Adapter uses BAPI_TRANSACTION_COMMIT functional module to call the commit on to SAP. If you want BAPI_TRANSACTION_COMMIT function call to wait until all the time-critical (V1) updates are completed, select the **Use wait parameter before calling BAPI commit** check box.
8. Click **Finish**.

Results

You specified a name for the top-level business object and selected an operation for the BAPIs. You also established the order of processing for the BAPIs. The Service Generation and Deployment Configuration window is displayed.

What to do next

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

Setting deployment properties and generating the service

To generate the module, which is the artifact that is deployed on IBM Business Process Manager or WebSphere Enterprise Service Bus, you create the module, associate the adapter with 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 Specify the Service Generation and Deployment Properties window should be displayed.

If you choose to use X509 certificate authentication (or) Single Sign On authentication (or) Secure Network Communication (SNC) to authenticate with SAP using the adapter, the target SAP Server has to be configured for the selected authentication mechanism. You must also obtain the required information for configuring the authentication mechanism from SAP (for example, SSO token, X509 certificate, SNC name, and SNC partner name).

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.

Procedure

1. 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**.

2. Indicate whether you will use an authentication alias (instead of typing a user ID and password) to establish a connection to the SAP server:
 - To specify an authentication alias, leave **Specify a Java Authentication and Authorization Services (JAAS) alias security credential** selected. Then, in the **J2C Authentication Data Entry** field, enter the name you specified in the Security section of the administrative console.
If you are not going to use an authentication alias, clear **Specify a Java Authentication and Authorization Services (JAAS) alias security credential**.
 - To use a user ID and password, select **Using security properties from the managed connection factory**. Type in the user name and password to specify your security credentials.
 - To use other security mechanisms native to the enterprise system or if security is not required, select **Other**
3. Select **With module for use by single application** to embed the adapter files in a module that is deployed to the application server, or select **On server for use by multiple applications** to install the adapter files as a stand-alone adapter.

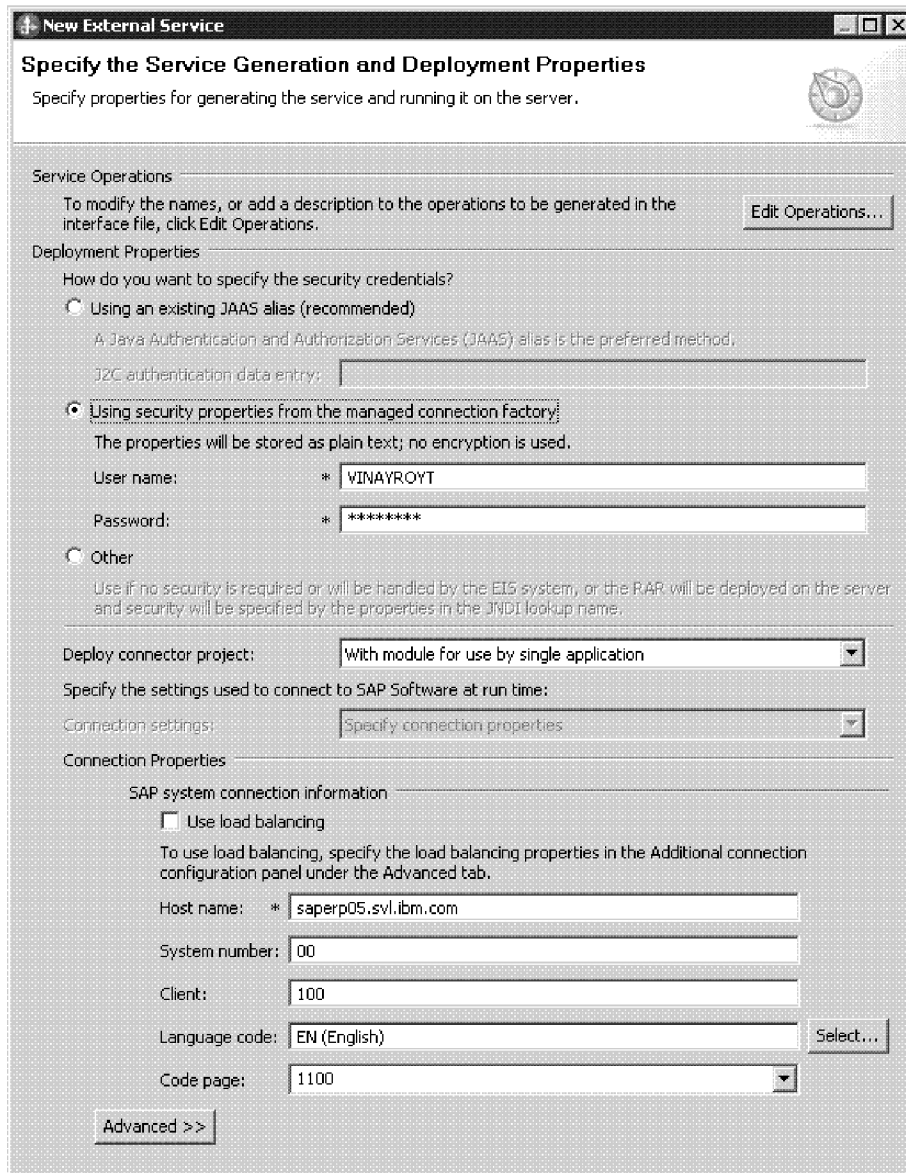


Figure 59. Connection properties

- **With module for use by single application:** With the adapter files embedded in the module, you can deploy the module to any application server. Use an embedded adapter when you have a single module using the adapter or if multiple modules need to run different versions of the adapter. Using an embedded adapter enables you to upgrade the adapter in a single module without the risk of destabilizing other modules by changing their adapter version.
- **On server for use by multiple applications:** If you do not include the adapter files in a module, you must install them as a stand-alone adapter on each application server where you want to run the module. Use a stand-alone adapter when multiple modules can use the same version of the adapter and you want to administer the adapter in a central location. A stand-alone adapter can also reduce the resources required by running a single adapter instance for multiple modules.

4. If you selected **On server for use by multiple applications** in the previous step, the **Connection properties** list becomes active. Make one of the following selections:

- Select **Specify connection properties** if you want to provide configuration information now. Then continue with step 5.
- Select **Use predefined connection properties** if you want to use a connection factory configuration that already exists.

If you decide to use predefined connection properties, you must ensure that your resource adapter name matches the name of the installed adapter, because this is how the adapter instance is associated with these properties. If you want to change the resource adapter name in the import or export, use the assembly editor in IBM Integration Designer to change the value in the import or export.

When you select **Use predefined connection properties**, the **JNDI Lookup Name** field is displayed in place of the properties.

- a. Type a value for **JNDI Lookup Name**.
- b. Click **Next**.
- c. Go to step 7 on page 131.

5. In the Connection properties section, set or change any connection properties that apply to your configuration.

Notice that some of the values are already filled in. For example, the values that you used in the Specify the Discovery Properties window (such as the **Host name**) are filled in.

Select the **Use load balancing** check box to use load balancing to connect to the SAP system. Load balancing properties which are **Message server host**, **Logon group name** and **SAP system ID** have to be specified in the Additional connection configuration panel under the Advanced tab.

See “Managed connection factory properties” on page 334 for more information about these properties.

Properties marked with an asterisk (*) are required.

6. To set additional properties, click **Advanced**.

a. Expand **Processing methods for illegal XML characters**

1) From the **Illegal XML character processing** drop-down list, select one of the options that correspond to your requirement:

- **Do not validate illegal XML characters**, if you want to work with the default adapter behavior.
- **Throw exception if contents in the BO contains illegal XML characters**, if you want to proceed after you get an exception message, and log the illegal XML characters in the trace file, at runtime.
- **Discard all illegal XML characters and related logs**, if you want the adapter to discard the illegal XML characters, and log the characters in the trace file, at runtime.

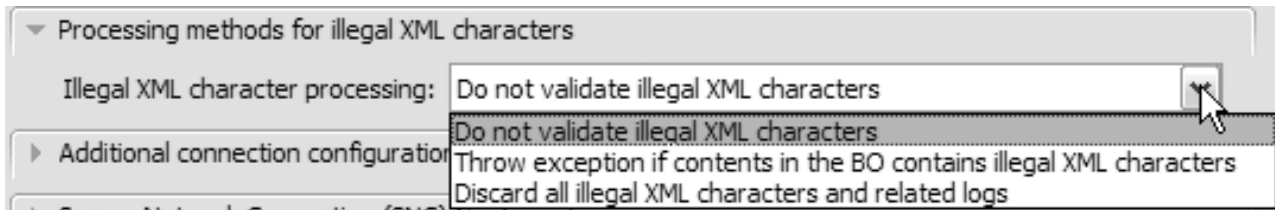


Figure 60. Processing methods for illegal XML characters

- b. Optionally expand **Additional connection configuration** and provide values (or change the default values) for the fields in this section of the window. For example, if your SAP configuration uses load balancing, provide values for the **Message server host** field, **Logon group name**, and **SAP system ID**. Select the **Reset the JCO client after closing connection handle** check box if during an outbound transaction, you want the adapter to make sure changes on the SAP EIS are reflected in the client. To enable the adapter to wait until all time critical updates to SAP database are completed before calling commit on it, select the **Wait until commit call to SAP database is completed and returned** check box. This option is only available if you are using the CWYAP_SAPAdapter_Tx.rar.
- c. Optionally, expand **Advanced Security Configuration** to specify your security mechanism. You can enable both **Secure Network Connection (SNC)** and **Single Sign On (SSO)**.
 - If you are using Secure Network Connection, select **Enable Secure Network Connection**. Enter information in the associated fields (name, partner, security level, and library path). If you select X509 certificate based authentication, the X509 certificate name has to be entered. Secure Network Connection has to be enabled to use X509 certificate based authentication.
 - If you are using Single Sign On authentication, select **Enable Single Sign On (SSO)**. Enter the SSO Logon Ticket in the field.
- d. Optionally to set RFC and JCo tracing properties, perform the following steps:
 - 1) Expand **SAP RFC trace configuration** and select **RFC trace on**.
 - 2) Select the **Write JCo traces into adapter logs** checkbox to enable JCo API traces to be generated in the broker trace file. These traces will then be interleaved with the adapter traces. Each JCo API trace logged into the broker trace file will have the string [JCoAPI]. All JCo API traces are logged at the INFO message log level.
 - 3) Select a tracing level from the **RFC trace level** list.
 - 4) Click **Browse** and select a location to which the RFC trace files will be saved.
- e. Optionally expand **Logging and tracing** and specify the following information:
 - If you have multiple instances of the adapter, set **Adapter ID** to a value that is unique for this instance.
 - If you want to mask sensitive information in log and trace files (for example, if you want to avoid making customer information visible in these files), select **Disguise user data as "XXX" in log and trace files** check box.

See “Managed connection factory properties” on page 334 for more information about these optional properties.

7. Click **Next**. The Specify the Location Properties window opens.
8. Create a module.
 - a. In the Specify the Location Properties window, click **New** in the **Module** field.
 - b. In the Integration Project window, click **Create a module project** or **Create a mediation module project** and click **Next**.
9. In the Module window, perform the following tasks:
 - a. Type a name for the module.
As you type the name, it is added to the workplace specified in the **Location** field.
This is the default location. If you want to specify a different location, remove the check from **Use default location** and type a new location or click **Browse** and select the location.
 - b. Specify whether you want to open the module in the assembly diagram (for module projects) or whether you want to create a mediation flow component (for mediation module projects). By default, these choices are selected.
 - c. Click **Finish**.
10. In the Specify the Location Properties window, perform the following steps:
 - a. If you want to change the default namespace, clear the **Use default namespace** check box and type a new path in the **Namespace** field.
 - b. 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. This is an optional step.
 - c. Optionally change the name of the interface.
The default name is SAPOutboundInterface. You can change it to a more descriptive title if you prefer.
 - d. If you want to save the business objects so that they can be used by another application, select **Save business objects to a library** and then select a library from the list or click **New** to create a new library.
 - e. Optionally type a description of the module.
11. Click **Finish**.

Results

The new module is added to the Business Integration perspective.

What to do next

Export the module as an EAR file for deployment.

Configuring a module for the BAPI result set interface

To configure a module to use the adapter for BAPI result set processing, you use the external service wizard in IBM Integration Designer to discover the BAPIs used to create the result set. You then configure the business objects that are generated and create a deployable module.

Selecting business objects and services for BAPI result set processing

To specify which BAPI functions you want to use and which data you want to process, you provide information in the external service wizard.

Before you begin

Make sure you have set the connection properties for the external service wizard.


About this task

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

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

Procedure

1. In the Find Objects in the Enterprise System window, indicate the BAPIs you want to work with.

- a. Click **RFC** to enable the filter  button.
- b. Click the filter button.

Note: Instead of using the filter capability, you can expand **RFC** and select the function from the list, or you can expand **BOR**, expand the functional grouping (for example, **Cross-Application Components**), and select the BAPI. Then skip ahead to step 4.

2. From the Filter Properties window, specify information about the BAPIs:
 - a. Select **Discover objects by name** or **Discover objects by description** from the **Object attribute to use for discovery** list.
 - b. 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.
 - c. Indicate the number of functions you want returned by changing the value in the **Maximum number of objects to retrieve** field or by accepting the default value.
 - d. Click **OK**.
3. Select the BAPIs.
 - a. Expand **RFC (filtered)**.
 - b. Select two BAPIs—GetList and GetDetail. One BAPI represents the query and one representing the results.

The following figure shows the **Discovered objects** list if you typed BAPI_CUSTOMER_GET* as the filter:

4. Click the arrow button to add the BAPIs to the **Selected objects** list.
5. In the Configuration Properties window, perform the following steps for each BAPI to add it to the list of business objects to be imported:
 - a. Optionally, select the **Use the original character case of SAP field names to generate business object attribute names** check box. By default (when the check box is not selected), the field descriptions are used to generate properties.

- b. Optionally, select the **Process Date fields as String** check box. This allows the adapter to retrieve SAP-specific date formats which are otherwise non-standard in Java by using the String type instead of the Date type format.

Note: To use this feature, ensure that the above check box is selected while running the external service wizard. As the date fields are converted to string type, ensure that all downstream dependencies (if any) are taken care of.

- c. If the BAPI has optional parameters associated with it, select the **Select optional parameters to include as child objects** check box, expand **Optional parameters**, and select the type of parameters (import, export, or table) that you want to work with.

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

If you are adding BAPI_CUSTOMER_GETDETAIL function module, you have the option of adding the following parameters:

Optional Import Parameters:

PI_PASS_BUFFER
PI_DIVISION
PI_DISTR_CHAN

Optional Export Parameters:

PE_ADDRESS
RETURN

Refer to the SAP documentation for a list and description of the optional parameters.

- d. 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 the left arrow.

6. Click **Next**.

Results

The external service wizard has returned the functions that match the search criteria, and you have selected the functions you want to work with. The Specify Composite Properties window is displayed.

What to do next

Provide information about the business object (such as the name of the top-level object and associated operation).

Configuring BAPI result set selected objects

To configure a BAPI result set business object, you specify information about the object (such as the name of the object and an indication of which BAPI is used as the query).

Before you begin

Make sure you have selected and imported the BAPI functions.

About this task

To configure the business object, use the following procedure.

Procedure

1. In the Specify Composite Properties window, select a name for the top-level business object.
2. Specify which BAPI is used as the query and select a property that forms the parent-child relationship between BAPIs:
 - a. Confirm that the correct BAPI is listed in the **Query BAPI** field. If it is not, select the other BAPI from the list.
 - b. Click **Add**.
 - c. To display all the properties associated with the first BAPI, click **Select**.
 - d. Select the property that you will use to form the parent-child relationship and click **OK**.

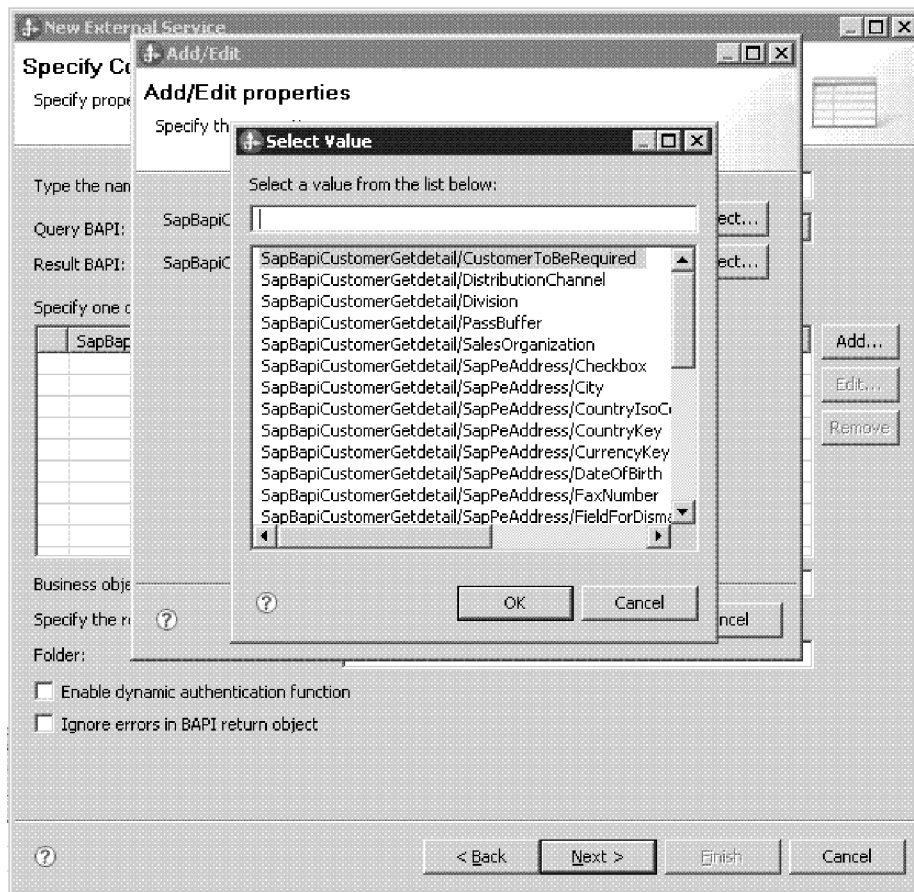


Figure 61. The list of properties for the selected BAPI

- e. To display all the properties associated with the second BAPI, click **Select**.
 - f. Select the property that you will use to form the parent-child relationship and click **OK**.
3. In the **Business object 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 external service wizard), change the namespace value.

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

4. To indicate where the business object information should be stored, type the path to the location in the **Folder** field. This is an optional step.
5. If you want the BAPI or BAPIs to be enclosed within a business graph, leave **Generate a business graph for each business object** selected. Otherwise, remove the check.
6. If you want to continue to process a BAPI even if the BAPI return object contains errors, select the **Ignore errors in BAPI Return object** check box.
7. Click **Finish**.

Results

You specified a name for the top-level business object and established the relationship between the BAPIs. The Service Generation and Deployment Configuration window is displayed.

What to do next

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

Setting deployment properties and generating the service

To generate the module, which is the artifact that is deployed on IBM Business Process Manager or WebSphere Enterprise Service Bus, you create the module, associate the adapter with 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 Specify the Service Generation and Deployment Properties window should be displayed.

If you choose to use X509 certificate authentication (or) Single Sign On authentication (or) Secure Network Communication (SNC) to authenticate with SAP using the adapter, the target SAP Server has to be configured for the selected authentication mechanism. You must also obtain the required information for configuring the authentication mechanism from SAP (for example, SSO token, X509 certificate, SNC name, and SNC partner name).

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.

Procedure

1. 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**.
2. Indicate whether you will use an authentication alias (instead of typing a user ID and password) to establish a connection to the SAP server:

- To specify an authentication alias, leave **Specify a Java Authentication and Authorization Services (JAAS) alias security credential** selected. Then, in the **J2C Authentication Data Entry** field, enter the name you specified in the Security section of the administrative console.
If you are not going to use an authentication alias, clear **Specify a Java Authentication and Authorization Services (JAAS) alias security credential**.
 - To use a user ID and password, select **Using security properties from the managed connection factory**. Type in the user name and password to specify your security credentials.
 - To use other security mechanisms native to the enterprise system or if security is not required, select **Other**
3. Select **With module for use by single application** to embed the adapter files in a module that is deployed to the application server, or select **On server for use by multiple applications** to install the adapter files as a stand-alone adapter.

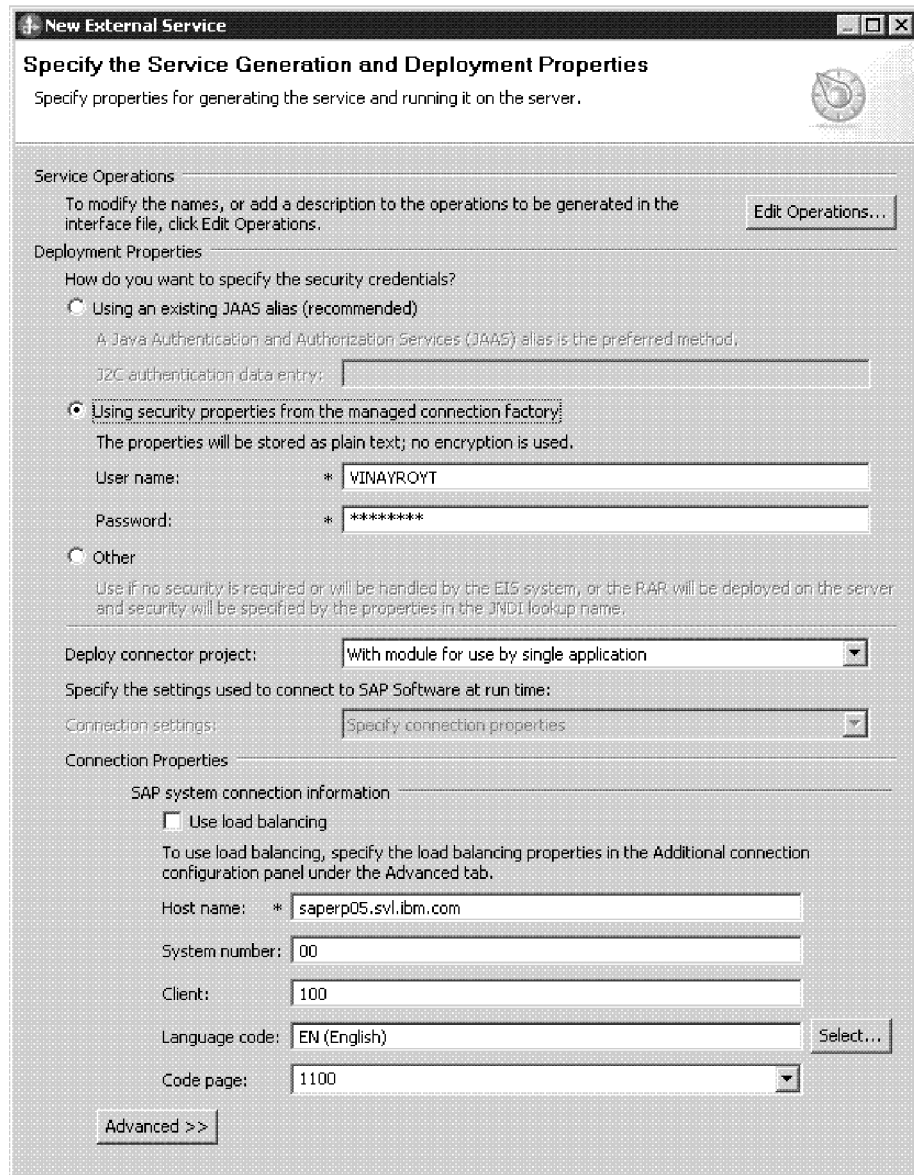


Figure 62. Connection properties

- **With module for use by single application:** With the adapter files embedded in the module, you can deploy the module to any application server. Use an embedded adapter when you have a single module using the adapter or if multiple modules need to run different versions of the adapter. Using an embedded adapter enables you to upgrade the adapter in a single module without the risk of destabilizing other modules by changing their adapter version.
- **On server for use by multiple applications:** If you do not include the adapter files in a module, you must install them as a stand-alone adapter on each application server where you want to run the module. Use a stand-alone adapter when multiple modules can use the same version of the adapter and you want to administer the adapter in a central location. A stand-alone adapter can also reduce the resources required by running a single adapter instance for multiple modules.

4. If you selected **On server for use by multiple applications** in the previous step, the **Connection properties** list becomes active. Make one of the following selections:

- Select **Specify connection properties** if you want to provide configuration information now. Then continue with step 5.
- Select **Use predefined connection properties** if you want to use a connection factory configuration that already exists.

If you decide to use predefined connection properties, you must ensure that your resource adapter name matches the name of the installed adapter, because this is how the adapter instance is associated with these properties. If you want to change the resource adapter name in the import or export, use the assembly editor in IBM Integration Designer to change the value in the import or export.

When you select **Use predefined connection properties**, the **JNDI Lookup Name** field is displayed in place of the properties.

- a. Type a value for **JNDI Lookup Name**.
 - b. Click **Next**.
 - c. Go to step 7 on page 140.
5. In the Connection properties section, set or change any connection properties that apply to your configuration.

Notice that some of the values are already filled in. For example, the values that you used in the Specify the Discovery Properties window (such as the **Host name**) are filled in.

Select the **Use load balancing** check box to use load balancing to connect to the SAP system. Load balancing properties which are **Message server host**, **Logon group name** and **SAP system ID** have to be specified in the Additional connection configuration panel under the Advanced tab.

See “Managed connection factory properties” on page 334 for more information about these properties.

Properties marked with an asterisk (*) are required.

6. To set additional properties, click **Advanced**.
 - a. Expand **Processing methods for illegal XML characters**
 - 1) From the **Illegal XML character processing** drop-down list, select one of the options that correspond to your requirement:
 - **Do not validate illegal XML characters**, if you want to work with the default adapter behavior.
 - **Throw exception if contents in the BO contains illegal XML characters**, if you want to proceed after you get an exception message, and log the illegal XML characters in the trace file, at runtime.
 - **Discard all illegal XML characters and related logs**, if you want the adapter to discard the illegal XML characters, and log the characters in the trace file, at runtime.

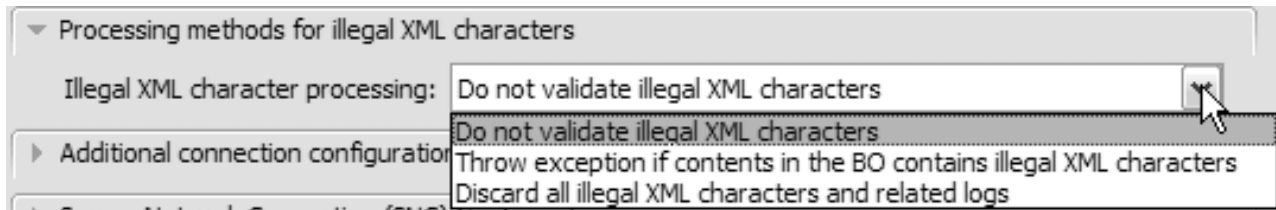


Figure 63. Processing methods for illegal XML characters

- b. Optionally expand **Additional connection configuration** and provide values (or change the default values) for the fields in this section of the window. For example, if your SAP configuration uses load balancing, provide values for the **Message server host** field, **Logon group name**, and **SAP system ID**. Select the **Reset the JCO client after closing connection handle** check box if during an outbound transaction, you want the adapter to make sure changes on the SAP EIS are reflected in the client. To enable the adapter to wait until all time critical updates to SAP database are completed before calling commit on it, select the **Wait until commit call to SAP database is completed and returned** check box. This option is only available if you are using the CWYAP_SAPAdapter_Tx.rar.
- c. Optionally, expand **Advanced Security Configuration** to specify your security mechanism. You can enable both **Secure Network Connection (SNC)** and **Single Sign On (SSO)**.
 - If you are using Secure Network Connection, select **Enable Secure Network Connection**. Enter information in the associated fields (name, partner, security level, and library path). If you select X509 certificate based authentication, the X509 certificate name has to be entered. Secure Network Connection has to be enabled to use X509 certificate based authentication.
 - If you are using Single Sign On authentication, select **Enable Single Sign On (SSO)**. Enter the SSO Logon Ticket in the field.
- d. Optionally to set RFC and JCo tracing properties, perform the following steps:
 - 1) Expand **SAP RFC trace configuration** and select **RFC trace on**.
 - 2) Select the **Write JCo traces into adapter logs** checkbox to enable JCo API traces to be generated in the broker trace file. These traces will then be interleaved with the adapter traces. Each JCo API trace logged into the broker trace file will have the string [JCoAPI]. All JCo API traces are logged at the INFO message log level.
 - 3) Select a tracing level from the **RFC trace level** list.
 - 4) Click **Browse** and select a location to which the RFC trace files will be saved.
- e. Optionally expand **Logging and tracing** and specify the following information:
 - If you have multiple instances of the adapter, set **Adapter ID** to a value that is unique for this instance.
 - If you want to mask sensitive information in log and trace files (for example, if you want to avoid making customer information visible in these files), select **Disguise user data as "XXX" in log and trace files** check box.

See “Managed connection factory properties” on page 334 for more information about these optional properties.

7. Click **Next**. The Specify the Location Properties window opens.
8. Create a module.
 - a. In the Specify the Location Properties window, click **New** in the **Module** field.
 - b. In the Integration Project window, click **Create a module project** or **Create a mediation module project** and click **Next**.
9. In the Module window, perform the following tasks:
 - a. Type a name for the module.
As you type the name, it is added to the workplace specified in the **Location** field.
This is the default location. If you want to specify a different location, remove the check from **Use default location** and type a new location or click **Browse** and select the location.
 - b. Specify whether you want to open the module in the assembly diagram (for module projects) or whether you want to create a mediation flow component (for mediation module projects). By default, these choices are selected.
 - c. Click **Finish**.
10. In the Specify the Location Properties window, perform the following steps:
 - a. If you want to change the default namespace, clear the **Use default namespace** check box and type a new path in the **Namespace** field.
 - b. 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. This is an optional step.
 - c. Optionally change the name of the interface.
The default name is SAPOutboundInterface. You can change it to a more descriptive title if you prefer.
 - d. If you want to save the business objects so that they can be used by another application, select **Save business objects to a library** and then select a library from the list or click **New** to create a new library.
 - e. Optionally type a description of the module.
11. Click **Finish**.

Results

The new module is added to the Business Integration perspective.

What to do next

Export the module as an EAR file for deployment.

Configuring a module for ALE outbound processing

To configure a module to use the adapter for ALE outbound processing, you use the external service wizard in IBM Integration Designer to find an IDoc or set of IDocs. You then configure the business objects that are generated and create a deployable module.

Selecting business objects and services for ALE outbound processing

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

About this task

For the ALE interface, you can select IDocs in one of two ways:

- You can specify an IDoc or a set of IDocs by entering search criteria (such as the name of the IDoc) and having the external service wizard search the SAP system.
- You can enter an IDoc definition file name with the complete path to its location on the file system.

If you choose to discover IDocs from a file, you must first configure the file. The file is generated from information on the SAP server and is then saved to your local file system.

Whichever method you choose, you can also specify the queue on the SAP server to which IDocs should be delivered.

Discovering IDocs from the system:

Use the **Discover IDocs from System** option to have the external service wizard search for IDocs based on the criteria you specify.

Before you begin


Make sure you have set the connection properties for the external service wizard.

About this task

Specify search criteria that the external service wizard uses to discover IDocs on the SAP server.

Note: The **Discover IDoc From System** choice applies to both the ALE interface and the ALE pass-through IDoc interface.

Procedure

1. In the Find Objects in the Enterprise System window, indicate which IDoc you want to work with.
 - a. Expand **ALE**.
 - b. Click **Discover IDoc From System** to enable the filter  button.
 - c. Click the filter button.

Note: Instead of using the filter button, you can expand **Discover IDoc From System** and select the IDoc from the list. You then skip ahead to step 4 on page 142.

2. From the Filter Properties window, specify information about the IDoc or IDocs:
 - a. Select **Discover objects by name** or **Discover objects by description** from the **Object attribute to use for discovery** list.
 - b. Type a search string (for example, ALEREQ*) representing the IDoc you want to call.

This is the name of the IDoc in SAP plus an asterisk as a wild card character to indicate that you want a list of all IDocs that start with ALEREQ.

- c. Select **Basic IDocs** or **Extension IDocs** from the **IDoc type to use for discovery** field.
 - d. Indicate the number of functions you want returned by changing the value in the **Maximum number of objects to retrieve** field or by accepting the default value.
 - e. Click **OK**.
3. Select the IDoc or IDocs.
 - a. Expand **Discover IDoc From System (filtered)**.
 - b. From the Discovered object list, click the IDoc you want to use. If you are working with multiple IDocs, click the names of all the IDocs.
 4. Click the arrow button to add the IDoc or IDocs to the **Selected objects** list.
 5. In the Configuration Parameters window, perform the following tasks to add the IDoc to the list of business objects to be imported.
 - a. Optionally select the **Use SAP field names to generate attribute names**. By default, when the check box is not selected, field descriptions are used to generate properties. If you choose to use SAP field names to generate attribute names, two more check boxes are enabled:
 - b. Optionally select the **Use SAP field names to generate attribute names** check box. By default (when the first check box is not selected), field descriptions are used to generate properties.
 - c. Select the **Use SAP-original casing for Control Record Business object attribute names** check box to generate attribute names in SAP original casing. If unchecked, attribute names are generated using field names from SAP in camel-casing.
 - d. Select the **Use SAP-original casing for Data Record Business object attribute names** check box to generate attribute names in SAP original casing. If unchecked, attribute names are generated using field Name from SAP in camel-casing.

The following are the different possible combinations:

Table 17.

Scenario	Use SAP field names to generate attribute names (check-box)	Use SAP-original casing Control Record business object attribute names (check-box)	Use SAP-original casing Data Record business object attribute names (check-box)	Control Record	Data Record
1	Checked	Checked	Checked	Attribute names are generated using Field Name from SAP (Original SAP casing)	Attribute names are generated using Field Name from SAP (Original SAP casing)

Table 17. (continued)

Scenario	Use SAP field names to generate attribute names (check-box)	Use SAP-original casing Control Record business object attribute names (check-box)	Use SAP-original casing Data Record business object attribute names (check-box)	Control Record	Data Record
2	Checked	Checked	Not checked	Attribute names are generated using Field Name from SAP (Original SAP casing)	Attribute names are generated using Field Name from SAP (Camel-casing)
3	Checked	Not checked	Checked	Attribute names are generated using Field Name from SAP (Camel-casing)	Attribute names are generated using Field Name from SAP (Original SAP casing)
4	Checked	Not checked	Not checked	Attribute names are generated using Field Name from SAP (Camel-casing)	Attribute names are generated using Field Name from SAP (Camel-casing)
5	Not checked	Not checked	Not checked	Attribute names are generated using Field Description from SAP (Camel-casing)	Attribute names are generated using Field Description from SAP (Camel-casing)

- e. If you want to have IDocs sent to a queue on the SAP server, click **Use qRFC to serialize outbound data using a queue**, and then select the queue from the **Select the queue name** list.
- f. In the **IDoc release version** field, specify the SAP release number to identify the IDoc type you want the external service wizard to use for creating business objects. If the selected IDoc has unreleased segments, the **IDoc release version** property becomes required. It is recommended that you select the default value **Unreleased** if the IDoc you are working with has unreleased segments. If **Unreleased** is selected, the adapter generates the business objects for segments using the unreleased segment definition.
- g. In the **IDoc release version** field, specify the SAP release number to identify the IDoc type you want the external service wizard to use for creating business objects. The version of the SAP system is set as default. If the

selected IDoc has unreleased segments, the **IDoc release version** property becomes required. It is recommended that you select **Unreleased** if the IDoc you are working with has unreleased segments. If **Unreleased** is selected, the adapter generates the business objects for segments using the unreleased segment definition.

- h. Optional: In the node area, you can either select or deselect the optional segments for which you want to generate the artifacts that will be part of the Data record.
 - i. Click **OK**.
6. Click **Next**.

Results

The external service wizard has returned an IDoc or a list of IDocs, and you have selected the ones you want to work with. You see the Specify Composite Properties window.

What to do next

Optionally specify a namespace and directory to which the generated business object will be stored and indicate whether you want a business graph generated.

Discovering IDocs from a file:

To select IDocs from a file, you must first configure an IDoc definition file based on information on the SAP server. You then specify, in the external service wizard, the path to the file on your local system.

Before you begin

You must have created an IDoc definition file.

Note: If you are using **Discover IDoc From System**, do not complete the following steps. The IDoc definition file is needed only if you are using **Discover IDoc From File**.

About this task

Specify the IDoc definition file that the external service wizard uses to discover the IDoc.

Procedure

1. In the Find Objects in the Enterprise System window, indicate which IDoc you want to work with.
 - a. Expand **ALE**.
 - b. Click **Discover IDoc From File** to enable the filter



button.

- c. Click the filter button.

Note: Instead of using the filter button, you can expand **Discover IDoc From File** and select the IDoc definition file. You then skip ahead to step 4 on page 146.

2. From the Filter Properties window, specify the location of the IDoc definition file.
 - a. Click **Browse** to navigate to the IDoc definition file, or type the path to the file.

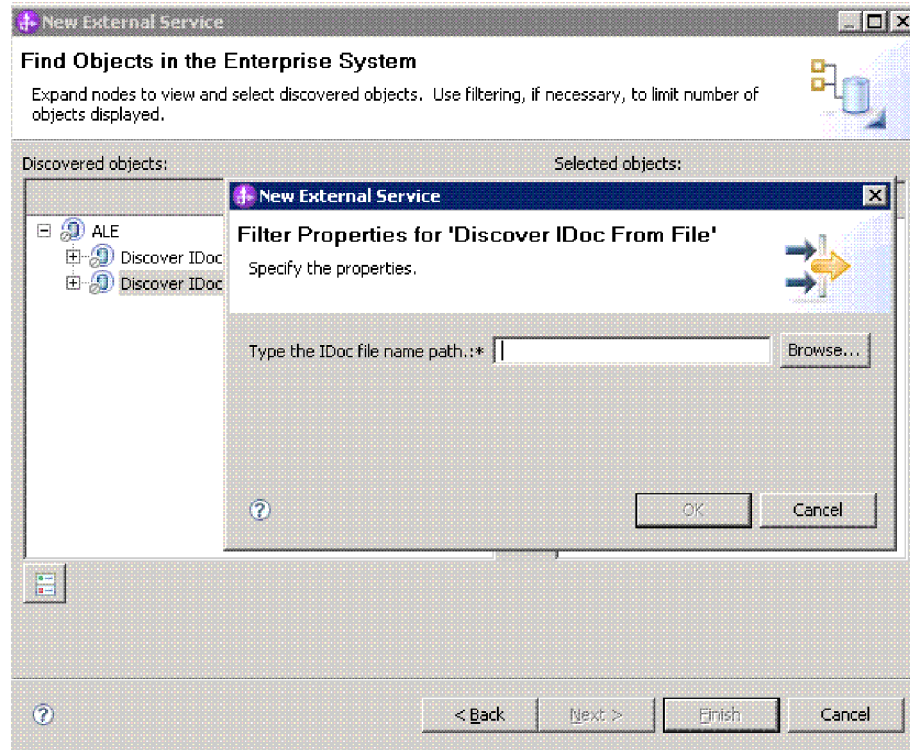


Figure 64. The Filter Properties for Discover IDoc From File window

- b. After you type or select the file, click **OK**.
3. Select the IDoc or IDocs.
 - a. Expand **Discover IDoc From File (filtered)**.
The IDoc definition file is displayed.
 - b. Click the IDoc definition file.

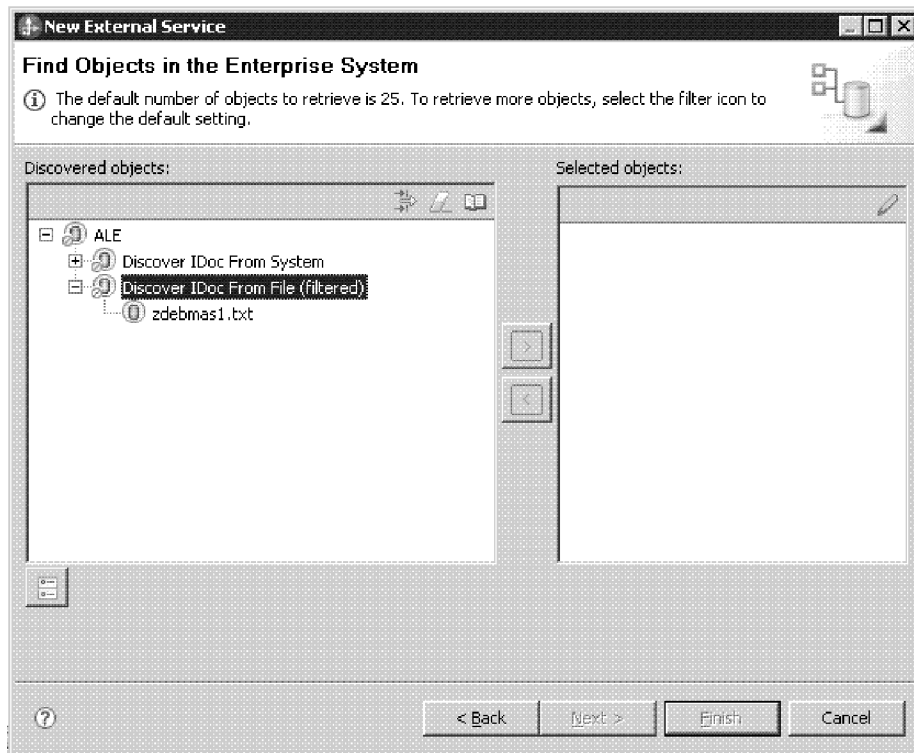


Figure 65. The Find Objects in the Enterprise System window

4. Click the arrow button to add it to the **Selected objects** list.
5. In the Configuration Parameters window, perform the following tasks:
 - a. Optionally select the **Use SAP field names to generate attribute names**. By default, when the check box is not selected, field descriptions are used to generate properties. If you choose to use SAP field names to generate attribute names, two more check boxes are enabled:
 - b. Select the **Use SAP-original casing for Control Record Business object attribute names** check box to generate attribute names in SAP original casing. If unchecked, attribute names are generated using field names from SAP in camel-casing.
 - c. Select the **Use SAP-original casing for Data Record Business object attribute names** check box to generate attribute names in SAP original casing. If unchecked, attribute names are generated using field Name from SAP in camel-casing.

The following are the different possible combinations:

Table 18.

Scenario	Use SAP field names to generate attribute names (check-box)	Use SAP-original casing Control Record business object attribute names (check-box)	Use SAP-original casing Data Record business object attribute names (check-box)	Control Record	Data Record
1	Checked	Checked	Checked	Attribute names are generated using Field Name from SAP (Original SAP casing)	Attribute names are generated using Field Name from SAP (Original SAP casing)
2	Checked	Checked	Not checked	Attribute names are generated using Field Name from SAP (Original SAP casing)	Attribute names are generated using Field Name from SAP (Camel-casing)
3	Checked	Not checked	Checked	Attribute names are generated using Field Name from SAP (Camel-casing)	Attribute names are generated using Field Name from SAP (Original SAP casing)
4	Checked	Not checked	Not checked	Attribute names are generated using Field Name from SAP (Camel-casing)	Attribute names are generated using Field Name from SAP (Camel-casing)
5	Not checked	Not checked	Not checked	Attribute names are generated using Field Description from SAP (Camel-casing)	Attribute names are generated using Field Description from SAP (Camel-casing)

- d. If you want to have IDocs sent to a queue on the SAP server, click **Use qRFC to serialize outbound data with a queue**, and then select the queue from the **Select the queue name** list.
- e. In the **IDoc release version** field, specify the SAP release number to identify the IDoc type you want the external service wizard to use for creating business objects.

- f. Click **OK**.
6. Click **Next**.

Results

The external service wizard has returned an IDoc or a list of IDocs associated with the IDoc definition file. You see the Configure Composite Properties window.

What to do next

Optionally specify a namespace and directory to which the generated business object will be stored and indicate whether you want a business graph generated

Configuring the selected objects

To configure the business object, you specify information about the object (such as the name of the folder where the object will be stored).

Before you begin

Make sure you have selected and imported the ALE IDoc.

About this task

Note: This task does not apply to business objects generated with the ALE pass-through IDoc interface.

To configure the business object, use the following procedure.

Procedure

1. In the **Business object 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 external service wizard), change the namespace value.
For example, you could change the namespace to <http://www.ibm.com/xmlns/prod/websphere/j2ca/sap1>.
2. To indicate where the business object information should be stored, type the path to the location in the **Folder** field. This is an optional step.

Note: This field cannot be edited if you are modifying existing artifacts.

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

3. If you want the IDoc or IDocs to be enclosed within a business graph, leave **Generate a business graph for each business object** selected. Otherwise, remove the check.
4. Click **Next**.

Results

You have optionally specified a location where the object is stored and changed the namespace. The Service Generation and Deployment Configuration window is displayed.

What to do next

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

Setting deployment properties and generating the service

To generate the module, which is the artifact that is deployed on IBM Business Process Manager 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 Specify the Service Generation and Deployment Properties window should be displayed.

If you choose to use X509 certificate authentication (or) Single Sign On authentication (or) Secure Network Communication (SNC) to authenticate with SAP using the adapter, the target SAP Server has to be configured for the selected authentication mechanism. You must also obtain the required information for configuring the authentication mechanism from SAP (for example, SSO token, X509 certificate, SNC name, and SNC partner name).

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.

Procedure

1. 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**.
2. Indicate whether you will use an authentication alias (instead of typing a user ID and password) to establish a connection to the SAP server:
 - To specify an authentication alias, leave **Specify a Java Authentication and Authorization Services (JAAS) alias security credential** selected. Then, in the **J2C Authentication Data Entry** field, enter the name you specified in the Security section of the administrative console.
If you are not going to use an authentication alias, clear **Specify a Java Authentication and Authorization Services (JAAS) alias security credential**.
 - To use a user ID and password, select **Using security properties from the managed connection factory**. Type in the user name and password to specify your security credentials.
 - To use other security mechanisms native to the enterprise system or if security is not required, select **Other**

3. Select **With module for use by single application** to embed the adapter files in a module that is deployed to the application server, or select **On server for use by multiple applications** to install the adapter files as a stand-alone adapter.

New External Service

Specify the Service Generation and Deployment Properties

Specify properties for generating the service and running it on the server.

Service Operations

To modify the names, or add a description to the operations to be generated in the interface file, click **Edit Operations...**

Deployment Properties

How do you want to specify the security credentials?

Using an existing JAAS alias (recommended)

A Java Authentication and Authorization Services (JAAS) alias is the preferred method.

J2C authentication data entry: _____

Using security properties from the managed connection factory

The properties will be stored as plain text; no encryption is used.

User name: * VINAYROYT

Password: * *****

Other

Use if no security is required or will be handled by the EIS system, or the RAR will be deployed on the server and security will be specified by the properties in the JNDI lookup name.

Deploy connector project: With module for use by single application

Specify the settings used to connect to SAP Software at run time:

Connection settings: Specify connection properties

Connection Properties

SAP system connection information

Use load balancing

To use load balancing, specify the load balancing properties in the Additional connection configuration panel under the Advanced tab.

Host name: * saperp05.svl.ibm.com

System number: 00

Client: 100

Language code: EN (English) **Select...**

Code page: 1100

Advanced >>

Figure 66. Connection properties

- **With module for use by single application:** With the adapter files embedded in the module, you can deploy the module to any application server. Use an embedded adapter when you have a single module using the adapter or if multiple modules need to run different versions of the adapter. Using an embedded adapter enables you to upgrade the adapter in a single module without the risk of destabilizing other modules by changing their adapter version.
- **On server for use by multiple applications:** If you do not include the adapter files in a module, you must install them as a stand-alone adapter on each application server where you want to run the module. Use a stand-alone adapter when multiple modules can use the same version of the

adapter and you want to administer the adapter in a central location. A stand-alone adapter can also reduce the resources required by running a single adapter instance for multiple modules.

4. If you selected **On server for use by multiple applications** in the previous step, the **Connection properties** list becomes active. Make one of the following selections:

- Select **Specify connection properties** if you want to provide configuration information now. Then continue with step 5.
- Select **Use predefined connection properties** if you want to use a connection factory configuration that already exists.

If you decide to use predefined connection properties, you must ensure that your resource adapter name matches the name of the installed adapter, because this is how the adapter instance is associated with these properties. If you want to change the resource adapter name in the import or export, use the assembly editor in IBM Integration Designer to change the value in the import or export.

When you select **Use predefined connection properties**, the **JNDI Lookup Name** field is displayed in place of the properties.

- a. Type a value for **JNDI Lookup Name**.
- b. Click **Next**.
- c. Go to step 8 on page 152.

5. In the Connection properties section, set or change any connection properties that apply to your configuration.

Notice that some of the values are already filled in. For example, the values that you used in the Specify the Discovery Properties window (such as the **Host name**) are filled in.

Select the **Use load balancing** check box to use load balancing to connect to the SAP system. Load balancing properties which are **Message server host**, **Logon group name** and **SAP system ID** have to be specified in the Additional connection configuration panel under the Advanced tab.

See “Managed connection factory properties” on page 334 for more information about these properties.

Properties marked with an asterisk (*) are required.

6. To set additional properties, click **Advanced**.

a. Expand **Processing methods for illegal XML characters**

1) From the **Illegal XML character processing** drop-down list, select one of the options that correspond to your requirement:

- **Do not validate illegal XML characters**, if you want to work with the default adapter behavior.
- **Throw exception if contents in the BO contains illegal XML characters**, if you want to proceed after you get an exception message, and log the illegal XML characters in the trace file, at runtime.
- **Discard all illegal XML characters and related logs**, if you want the adapter to discard the illegal XML characters, and log the characters in the trace file, at runtime.

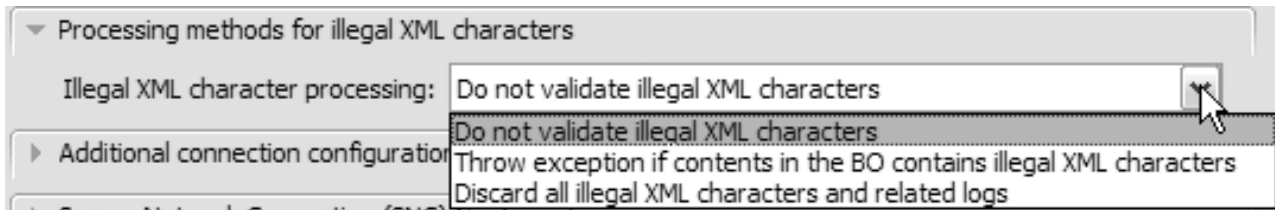


Figure 67. Processing methods for illegal XML characters

- b. Optionally expand **Additional connection configuration** and provide values (or change the default values) for the fields in this section of the window. For example, if your SAP configuration uses load balancing, provide values for the **Message server host** field, **Logon group name** and **SAP system ID**. Select the **Reset the JCO client after closing connection handle** check box if during an outbound transaction, you want the adapter to make sure changes on the SAP EIS are reflected in the client. For ALE transactions the property does not have a functionality impact. It is recommended that you do not select the check box.
- c. Optionally, expand **Advanced Security Configuration** to specify your security mechanism. You can enable both **Secure Network Connection (SNC)** and **Single Sign On (SSO)**.
 - If you are using Secure Network Connection, select **Enable Secure Network Connection**. Enter information in the associated fields (name, partner, security level, and library path). If you select X509 certificate based authentication, the X509 certificate name has to be entered. Secure Network Connection has to be enabled to use X509 certificate based authentication.
 - If you are using Single Sign On authentication, select **Enable Single Sign On (SSO)**. Enter the SSO Logon Ticket in the field.
- d. Optionally to set RFC and JCo tracing properties, perform the following steps:
 - 1) Expand **SAP RFC trace configuration** and select **RFC trace on**.
 - 2) Select the **Write JCo traces into adapter logs** checkbox to enable JCo API traces to be generated in the broker trace file. These traces will then be interleaved with the adapter traces. Each JCo API trace logged into the broker trace file will have the string [JCoAPI]. All JCo API traces are logged at the INFO message log level.
 - 3) Select a tracing level from the **RFC trace level** list.
 - 4) Click **Browse** and select a location to which the RFC trace files will be saved.
- e. Optionally expand **Logging and tracing** and specify the following information:
 - If you have multiple instances of the adapter, set **Adapter ID** to a value that is unique for this instance.
 - If you want to mask sensitive information in log and trace files (for example, if you want to avoid making customer information visible in these files), select **Disguise user data as "XXX" in log and trace files**.

See “Managed connection factory properties” on page 334 for more information about these optional properties.
7. Click **Next**. The Specify the Location Properties window opens.
8. Create a module.

- a. In the Specify the Location Properties window, click **New** in the **Module** field.
 - b. In the Integration Project window, click **Create a module project** or **Create a mediation module project** and click **Next**.
9. In the New Module window, perform the following tasks:
- a. Type a name for the module.
As you type the name, it is added to the workplace specified in the **Location** field.
This is the default location. If you want to specify a different location, remove the check from **Use default location** and type a new location or click **Browse** and select the location.
 - b. Specify whether you want to open the module in the assembly diagram (for module projects) or whether you want to create a mediation flow component (for mediation module projects). By default, these choices are selected.
 - c. Click **Finish**.
10. In the Specify the Location Properties window, perform the following tasks:
- a. If you want to change the default namespace, clear the **Use default namespace** check box and type a new path in the **Namespace** field.
 - b. 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. This is an optional step.
 - c. Optionally change the name of the interface.
The default name is `SAPOutboundInterface`. You can change it to a more descriptive title if you prefer.
 - d. If you want to save the business objects so that they can be used by another application, click **Save business objects to a library** and then select a library from the list or click **New** to create a new library.
 - e. Optionally type a description of the module.
11. Click **Finish**.

Results

The new module is added to the Business Integration perspective.

What to do next

Export the module as an EAR file for deployment.

Configuring a module for ALE pass-through IDoc outbound processing

To configure a module to use the adapter for ALE outbound processing, you use the external service wizard in IBM Integration Designer to find an IDoc or set of IDocs. You then configure the business objects that are generated and create a deployable module.

Selecting business objects and services for ALE pass-through IDoc outbound processing

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

About this task

For the ALE pass-through IDoc interface, you can specify IDocs from a system or from a file, but the most likely reason for using the pass-through IDoc interface is to use a generic IDoc.

- When you select a generic IDoc, you create one business-object definition that can apply to any IDoc at run time. This selection is helpful if you are processing many IDocs and do not want to create a separate business-object definition for each one.
- If specify an IDoc from a system or file, you select a specific IDoc (for example, ORDERS05) during configuration. However, you can use a different IDoc during run time when you send the request to the SAP server.

Whichever method you choose, you can also specify the queue on the SAP server to which IDocs should be delivered.

Procedure

1. In the Find Objects in the Enterprise System window, indicate that you want to select a generic IDoc.
 - a. Expand **ALE**.
 - b. Click **Generic IDoc** .
2. Click the arrow button to add the generic IDoc to the **Selected objects** list.
3. When the Configuration Parameters window is displayed, indicate whether you want to have IDocs sent to a queue on the SAP server:
 - If you do not want to send IDocs to a queue, click **Cancel**.
 - If you want to send IDocs to a queue, perform the following steps:
 - a. Click **Use qRFC to serialize outbound data using a queue**.
 - b. Select a queue from the **Select the queue name** list.
 - If your Data Record size to be less than the standard length (1063 characters) specified by SAP adapter, use a delimiter value in the **Delimiter to be used for splitting multiple IDocs** field. Any valid string without escape characters or either of the two escape characters, \n or \r\n can be used as a delimiter value. For more information on delimiters refer to “Outbound processing for the ALE pass-through IDoc interface” on page 67.

If your hexbinary content has non-unicode characters, set the partner character set to the specific character set in the following steps: 5 on page 151
4. Click **OK**.
5. Click **Next**.

Results

You have selected a generic IDoc.

What to do next

Set deployment properties and generate a module.

Setting deployment properties and generating the service

To generate the module, which is the artifact that is deployed on IBM Business Process Manager 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 Specify the Service Generation and Deployment Properties window should be displayed.

If you choose to use X509 certificate authentication (or) Single Sign On authentication (or) Secure Network Communication (SNC) to authenticate with SAP using the adapter, the target SAP Server has to be configured for the selected authentication mechanism. You must also obtain the required information for configuring the authentication mechanism from SAP (for example, SSO token, X509 certificate, SNC name, and SNC partner name).

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.

Procedure

1. 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**.
2. Indicate whether you will use an authentication alias (instead of typing a user ID and password) to establish a connection to the SAP server:
 - To specify an authentication alias, leave **Specify a Java Authentication and Authorization Services (JAAS) alias security credential** selected. Then, in the **J2C Authentication Data Entry** field, enter the name you specified in the Security section of the administrative console.
If you are not going to use an authentication alias, clear **Specify a Java Authentication and Authorization Services (JAAS) alias security credential**.
 - To use a user ID and password, select **Using security properties from the managed connection factory**. Type in the user name and password to specify your security credentials.
 - To use other security mechanisms native to the enterprise system or if security is not required, select **Other**
3. Select **With module for use by single application** to embed the adapter files in a module that is deployed to the application server, or select **On server for use by multiple applications** to install the adapter files as a stand-alone adapter.

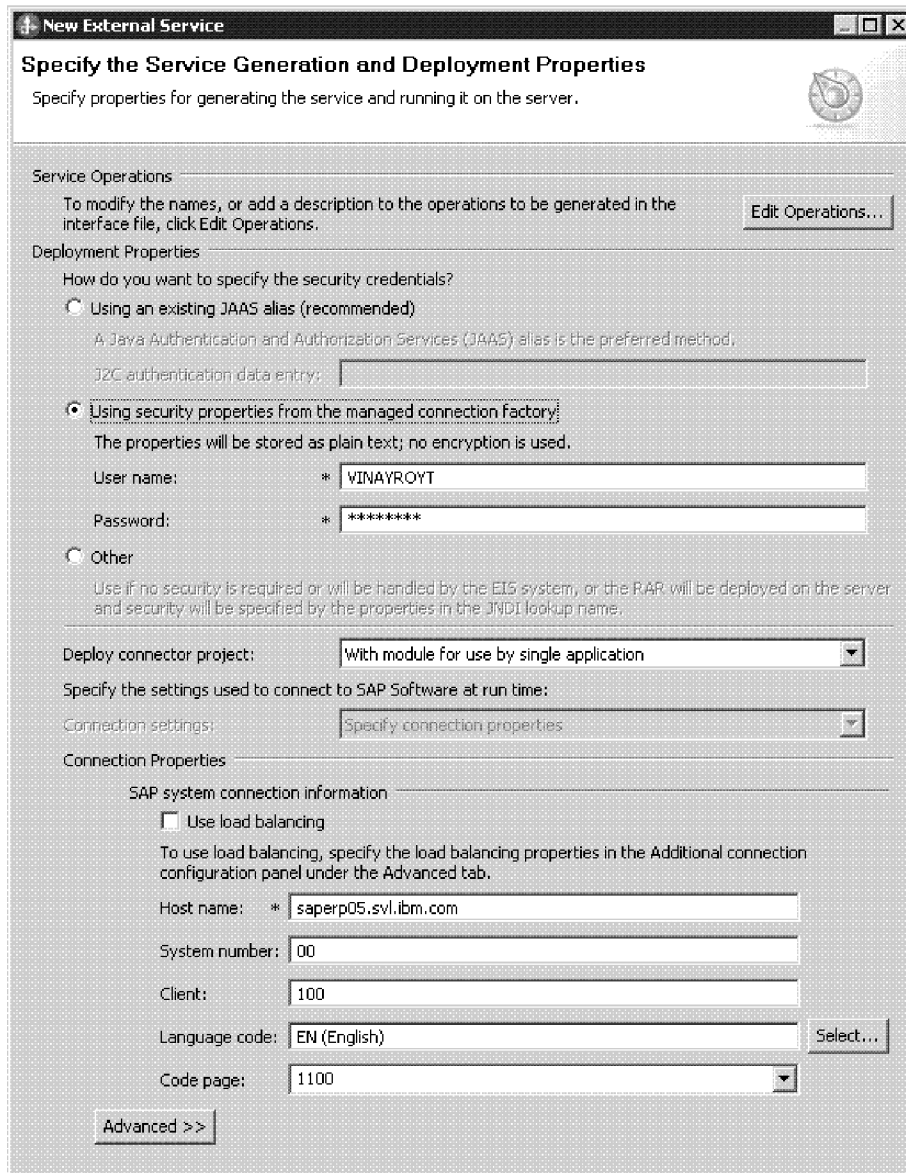


Figure 68. Connection properties

- **With module for use by single application:** With the adapter files embedded in the module, you can deploy the module to any application server. Use an embedded adapter when you have a single module using the adapter or if multiple modules need to run different versions of the adapter. Using an embedded adapter enables you to upgrade the adapter in a single module without the risk of destabilizing other modules by changing their adapter version.
- **On server for use by multiple applications:** If you do not include the adapter files in a module, you must install them as a stand-alone adapter on each application server where you want to run the module. Use a stand-alone adapter when multiple modules can use the same version of the adapter and you want to administer the adapter in a central location. A stand-alone adapter can also reduce the resources required by running a single adapter instance for multiple modules.

4. If you selected **On server for use by multiple applications** in the previous step, the **Connection properties** list becomes active. Make one of the following selections:

- Select **Specify connection properties** if you want to provide configuration information now. Then continue with step 5.
- Select **Use predefined connection properties** if you want to use a connection factory configuration that already exists.

If you decide to use predefined connection properties, you must ensure that your resource adapter name matches the name of the installed adapter, because this is how the adapter instance is associated with these properties. If you want to change the resource adapter name in the import or export, use the assembly editor in IBM Integration Designer to change the value in the import or export.

When you select **Use predefined connection properties**, the **JNDI Lookup Name** field is displayed in place of the properties.

- a. Type a value for **JNDI Lookup Name**.
- b. Click **Next**.
- c. Go to step 8 on page 158.

5. In the Connection properties section, set or change any connection properties that apply to your configuration.

Notice that some of the values are already filled in. For example, the values that you used in the Specify the Discovery Properties window (such as the **Host name**) are filled in.

Select the **Use load balancing** check box to use load balancing to connect to the SAP system. Load balancing properties which are **Message server host**, **Logon group name** and **SAP system ID** have to be specified in the Additional connection configuration panel under the Advanced tab.

See “Managed connection factory properties” on page 334 for more information about these properties.

Properties marked with an asterisk (*) are required.

6. To set additional properties, click **Advanced**.

a. Expand **Processing methods for illegal XML characters**

1) From the **Illegal XML character processing** drop-down list, select one of the options that correspond to your requirement:

- **Do not validate illegal XML characters**, if you want to work with the default adapter behavior.
- **Throw exception if contents in the BO contains illegal XML characters**, if you want to proceed after you get an exception message, and log the illegal XML characters in the trace file, at runtime.
- **Discard all illegal XML characters and related logs**, if you want the adapter to discard the illegal XML characters, and log the characters in the trace file, at runtime.

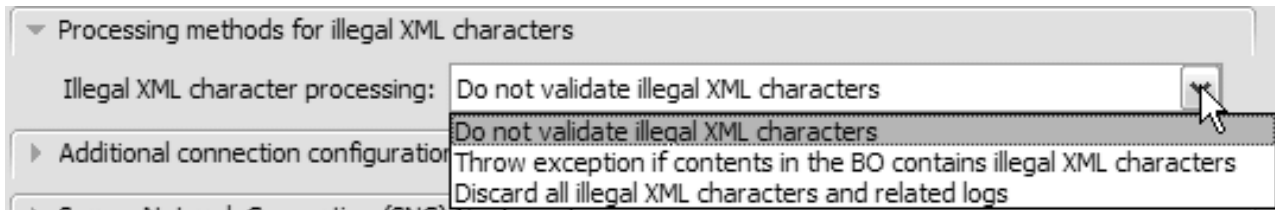


Figure 69. Processing methods for illegal XML characters

- b. Optionally expand **Additional connection configuration** and provide values (or change the default values) for the fields in this section of the window. For example, if your SAP configuration uses load balancing, provide values for the **Message server host** field, **Logon group name** and **SAP system ID**. Select the **Reset the JCO client after closing connection handle** check box if during an outbound transaction, you want the adapter to make sure changes on the SAP EIS are reflected in the client. For ALE transactions the property does not have a functionality impact. It is recommended that you do not select the check box.
- c. Optionally, expand **Advanced Security Configuration** to specify your security mechanism. You can enable both **Secure Network Connection (SNC)** and **Single Sign On (SSO)**.
 - If you are using Secure Network Connection, select **Enable Secure Network Connection**. Enter information in the associated fields (name, partner, security level, and library path). If you select X509 certificate based authentication, the X509 certificate name has to be entered. Secure Network Connection has to be enabled to use X509 certificate based authentication.
 - If you are using Single Sign On authentication, select **Enable Single Sign On (SSO)**. Enter the SSO Logon Ticket in the field.
- d. Optionally to set RFC and JCo tracing properties, perform the following steps:
 - 1) Expand **SAP RFC trace configuration** and select **RFC trace on**.
 - 2) Select the **Write JCo traces into adapter logs** checkbox to enable JCo API traces to be generated in the broker trace file. These traces will then be interleaved with the adapter traces. Each JCo API trace logged into the broker trace file will have the string [JCoAPI]. All JCo API traces are logged at the INFO message log level.
 - 3) Select a tracing level from the **RFC trace level** list.
 - 4) Click **Browse** and select a location to which the RFC trace files will be saved.
- e. Optionally expand **Logging and tracing** and specify the following information:
 - If you have multiple instances of the adapter, set **Adapter ID** to a value that is unique for this instance.
 - If you want to mask sensitive information in log and trace files (for example, if you want to avoid making customer information visible in these files), select **Disguise user data as "XXX" in log and trace files**.

See "Managed connection factory properties" on page 334 for more information about these optional properties.
7. Click **Next**. The Specify the Location Properties window opens.
8. Create a module.

- a. In the Specify the Location Properties window, click **New** in the **Module** field.
 - b. In the Integration Project window, click **Create a module project** or **Create a mediation module project** and click **Next**.
9. In the New Module window, perform the following tasks:
- a. Type a name for the module.
As you type the name, it is added to the workplace specified in the **Location** field.
This is the default location. If you want to specify a different location, remove the check from **Use default location** and type a new location or click **Browse** and select the location.
 - b. Specify whether you want to open the module in the assembly diagram (for module projects) or whether you want to create a mediation flow component (for mediation module projects). By default, these choices are selected.
 - c. Click **Finish**.
10. In the Specify the Location Properties window, perform the following tasks:
- a. If you want to change the default namespace, clear the **Use default namespace** check box and type a new path in the **Namespace** field.
 - b. 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. This is an optional step.
 - c. Optionally change the name of the interface.
The default name is `SAPOutboundInterface`. You can change it to a more descriptive title if you prefer.
 - d. If you want to save the business objects so that they can be used by another application, click **Save business objects to a library** and then select a library from the list or click **New** to create a new library.
 - e. Optionally type a description of the module.
11. Click **Finish**.

Results

The new module is added to the Business Integration perspective.

What to do next

Export the module as an EAR file for deployment.

Configuring a module for Query interface for SAP Software processing

To configure a module to use the adapter for Query interface for SAP Software outbound processing, you use the external service wizard in IBM Integration Designer to find data in an SAP table or a set of tables. You then configure the business objects that are generated and create a deployable module.

Selecting business objects and services

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

Before you begin

Make sure you have set the connection properties for the external service wizard.

About this task

Specify search criteria that the external service wizard uses to query data on the SAP server. The external service wizard returns the data that meets the search criteria.

You can use the discovered tables to generate individual objects (objects that have no relationship to each other) or to generate objects that have a hierarchical structure.

- If you are generating individual objects, you can import one or more objects from the list of discovered tables at the same time.
- If you are generating hierarchical objects, you must import the parent tables first and then import the child tables.

When you configure the child tables for import, you can select the parent table you imported earlier as its parent. Repeat this process to add more tables to the hierarchical structure. A hierarchical object with three levels, for example, requires three separate imports to establish the parent-child relationship.

To specify the search criteria, use the following procedure.

Procedure

1. In the Specify the Discovery Properties window, indicate which table or tables you want to work with.
 - a. Click **QISS** to enable the filter



button.

- b. Click the filter button.

Note: Instead of using the filter capability, you can expand **QISS** and select the table from the list. Then skip ahead to step 4.

2. From the Filter Properties window, specify information about the table.
 - a. Select **Discover objects by name** or **Discover objects by description** from the **Object attribute to use for discovery** list.
 - b. Type a search string (for example, KN*) representing the table.
This is the name of the table 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 KN.
 - c. Indicate the number of objects you want returned by changing the value in the **Maximum number of objects to retrieve** field or by accepting the default value.
 - d. Click **OK**.
3. Select the table objects.
 - a. Expand **QISS (filtered)**.
 - b. Click the table object you want to use.
4. Click the arrow button to add the table object to the **Selected objects** list.

5. In the Specify the Configuration Properties for 'object' *table* window, provide information about the table:
 - a. The **Add a WHERE clause** field specifies the primary key to the table. A default value is provided. Change this value if you want to use a different primary key.

In the example of the KNA1 table, shown in the following figure, the default value is KUNNR = /CustomerNumber1. The KUNNR field is one of the primary keys in the KNA1 table. The WHERE query will return information based on the customer number provided in the query.
 - b. Optionally select the **Use SAP field names to generate attribute names** check box. By default (when the check box is not selected), the field descriptions are used to generate properties.
 - c. Indicate which columns you want included in the query.

Note: There are many columns and, by default, all the columns are selected. You can clear the check from those columns you do not want included, or if you want to select only a few columns, you can use the **Select or unselect all columns** check box.

For example, if you want only two columns, clear **Select or unselect all columns** to remove the check from all columns, and then select the two columns you want.

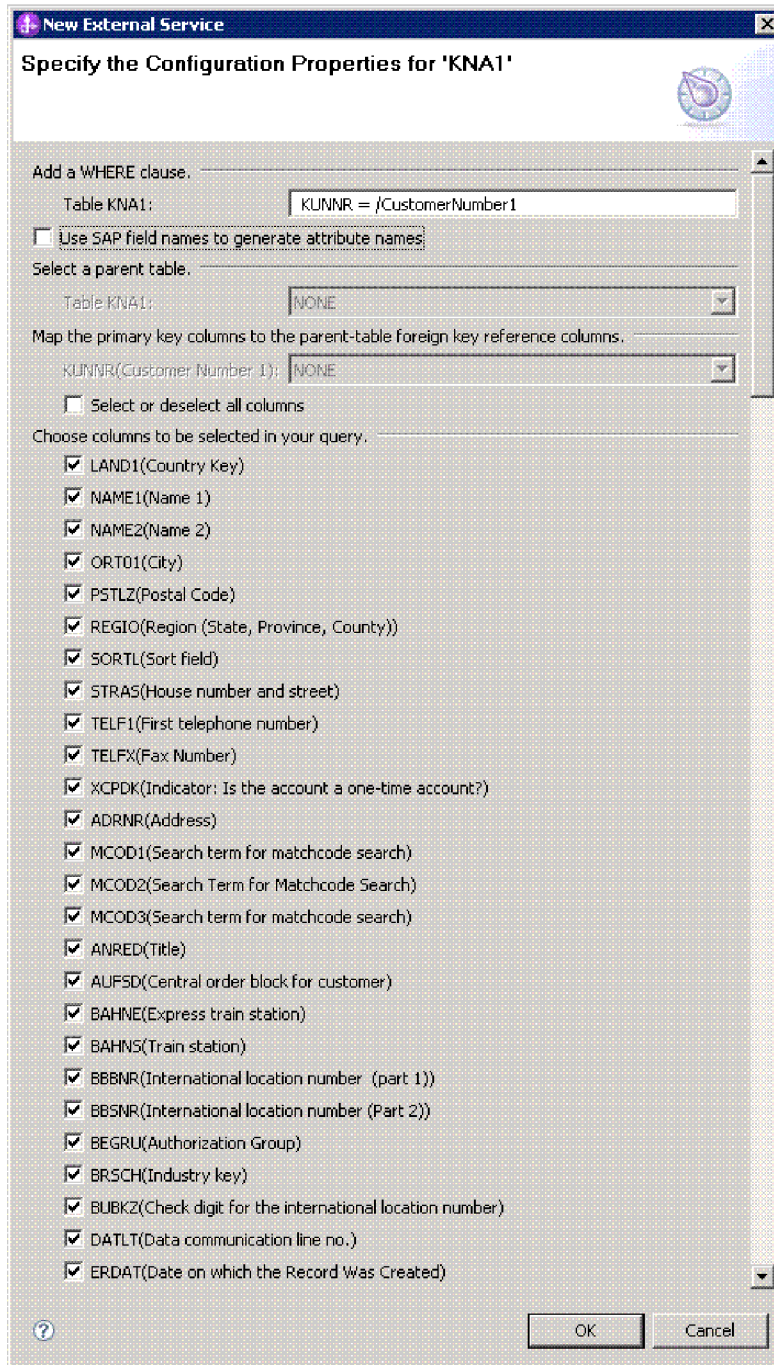


Figure 70. The Specify the Configuration Properties for 'object' KNA1 window

- d. Click **OK**
6. To include another table in the query, perform the following tasks:
 - a. Click **QISS** to enable the filter



button.

- b. Click the filter button.

Note: Instead of using the filter capability, you can expand **QISS** and select the table from the list.

7. From the Filter Properties window, specify information about the table.
 - a. Select **Discover objects by name** or **Discover objects by description** from the **Object attribute to use for discovery** list.
 - b. Type a search string (for example, ADRC) representing the table.
 - c. Indicate the number of functions you want returned by changing the value in the **Maximum number of objects to retrieve** field or by accepting the default value.
 - d. Click **OK**.
8. Select the table objects.
 - a. Expand **QISS (filtered)**.
 - b. Click the second table object.
 - c. Click the arrow button to add the table object to the **Selected objects** list.
9. In the Specify the Configuration Properties for 'object' *table* window, provide information about the table:
 - a. The **Add a WHERE clause** field specifies the primary key to the table. A default value is provided. Change this value if you want to use a different primary key.
 - b. Optionally select the **Use SAP field names to generate attribute names** check box. By default (when the check box is not selected), the field descriptions are used to generate properties.
 - c. Associate this table with the one you previously added (KNA1 in the example) by selecting that table from the **Select a parent table** section of the window.
 - d. Under **Map the primary key columns to the parent-table foreign key reference columns**, select a value to link the tables.
For example, you might select **ADRNR** for **ADDRNUMBER**.

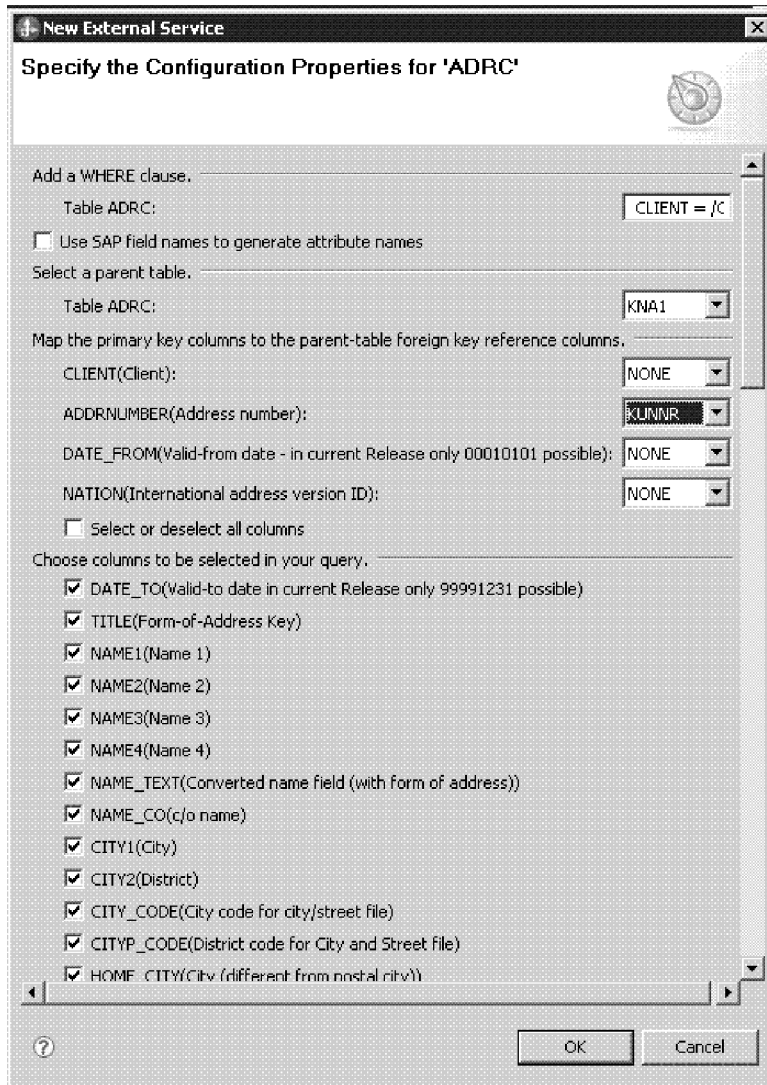


Figure 71. The Specify the Configuration Properties for 'object'ADRC window

- e. Indicate which columns you want included in the query.
 - f. Click **OK**
10. Click **Next**.

Results

The external service wizard returns the data that matches the search criteria.

What to do next

From the Specify Composite Properties window, optionally specify a namespace and directory to which the generated business object will be stored and indicate whether you want a business graph generated.

Configuring the selected objects

To configure the object, you specify information about where the object should be stored.

Before you begin

Make sure you have selected and imported the business object.

About this task

To configure the business object, use the following procedure.

Procedure

1. In the **Business object 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 external service wizard), change the namespace value.

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

2. To indicate where the business object information should be stored, type the path to the location in the **Folder** field. This is an optional step.

Note: This field cannot be edited if you are modifying existing artifacts.

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

3. If you want the business object to be enclosed within a business graph, leave **Generate a business graph for each business object** selected. Otherwise, remove the check.
4. In the **Custom retrieve function name** field, type the name of the function you created on the EIS to avoid the improper data truncation with delimiter set.
5. Click **Next** to continue to the Service Generation and Deployment Configuration window.

Results

You have made changes to the default settings (for example, changing the namespace), or you have accepted all the default settings. The Service Generation and Deployment Configuration window is displayed.

What to do next

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

Setting deployment properties and generating the service

To generate the module, which is the artifact that is deployed on IBM Business Process Manager or WebSphere Enterprise Service Bus, you create the module, associate the adapter with 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 Specify the Service Generation and Deployment Properties window should be displayed.

If you choose to use X509 certificate authentication (or) Single Sign On authentication (or) Secure Network Communication (SNC) to authenticate with SAP using the adapter, the target SAP Server has to be configured for the selected authentication mechanism. You must also obtain the required information for configuring the authentication mechanism from SAP (for example, SSO token, X509 certificate, SNC name, and SNC partner name).

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.

Procedure

1. 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**.
2. Indicate whether you will use an authentication alias (instead of typing a user ID and password) to establish a connection to the SAP server:
 - To specify an authentication alias, leave **Specify a Java Authentication and Authorization Services (JAAS) alias security credential** selected. Then, in the **J2C Authentication Data Entry** field, enter the name you specified in the Security section of the administrative console.
If you are not going to use an authentication alias, clear **Specify a Java Authentication and Authorization Services (JAAS) alias security credential**.
 - To use a user ID and password, select **Using security properties from the managed connection factory**. Type in the user name and password to specify your security credentials.
 - To use other security mechanisms native to the enterprise system or if security is not required, select **Other**
3. Select **With module for use by single application** to embed the adapter files in a module that is deployed to the application server, or select **On server for use by multiple applications** to install the adapter files as a stand-alone adapter.

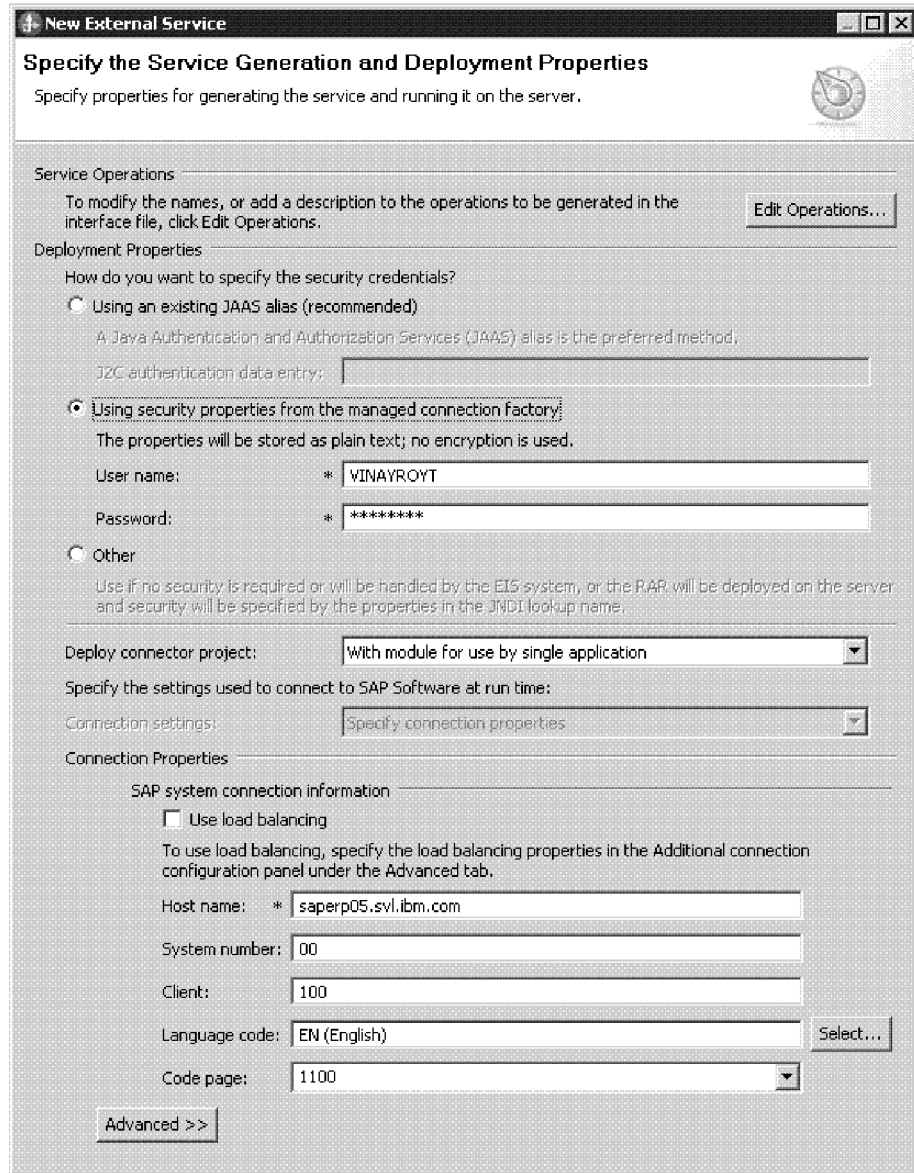


Figure 72. Connection properties

- **With module for use by single application:** With the adapter files embedded in the module, you can deploy the module to any application server. Use an embedded adapter when you have a single module using the adapter or if multiple modules need to run different versions of the adapter. Using an embedded adapter enables you to upgrade the adapter in a single module without the risk of destabilizing other modules by changing their adapter version.
- **On server for use by multiple applications:** If you do not include the adapter files in a module, you must install them as a stand-alone adapter on each application server where you want to run the module. Use a stand-alone adapter when multiple modules can use the same version of the adapter and you want to administer the adapter in a central location. A stand-alone adapter can also reduce the resources required by running a single adapter instance for multiple modules.

4. If you selected **On server for use by multiple applications** in the previous step, the **Connection properties** list becomes active. Make one of the following selections:

- Select **Specify connection properties** if you want to provide configuration information now. Then continue with step 5.
- Select **Use predefined connection properties** if you want to use a connection factory configuration that already exists.

If you decide to use predefined connection properties, you must ensure that your resource adapter name matches the name of the installed adapter, because this is how the adapter instance is associated with these properties. If you want to change the resource adapter name in the import or export, use the assembly editor in IBM Integration Designer to change the value in the import or export.

When you select **Use predefined connection properties**, the **JNDI Lookup Name** field is displayed in place of the properties.

- a. Type a value for **JNDI Lookup Name**.
- b. Click **Next**.
- c. Go to step 8 on page 169.

5. In the Connection properties section, set or change any connection properties that apply to your configuration.

Notice that some of the values are already filled in. For example, the values that you used in the Specify the Discovery Properties window (such as the **Host name**) are filled in.

Select the **Use load balancing** check box to use load balancing to connect to the SAP system. Load balancing properties which are **Message server host**, **Logon group name** and **SAP system ID** have to be specified in the Additional connection configuration panel under the Advanced tab.

See “Managed connection factory properties” on page 334 for more information about these properties.

Properties marked with an asterisk (*) are required.

6. To set additional properties, click **Advanced**.

a. Expand **Processing methods for illegal XML characters**

1) From the **Illegal XML character processing** drop-down list, select one of the options that correspond to your requirement:

- **Do not validate illegal XML characters**, if you want to work with the default adapter behavior.
- **Throw exception if contents in the BO contains illegal XML characters**, if you want to proceed after you get an exception message, and log the illegal XML characters in the trace file, at runtime.
- **Discard all illegal XML characters and related logs**, if you want the adapter to discard the illegal XML characters, and log the characters in the trace file, at runtime.

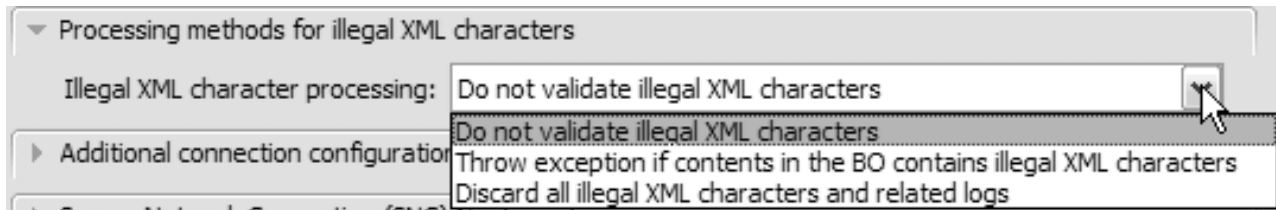


Figure 73. Processing methods for illegal XML characters

- b. Optionally expand **Additional connection configuration** and provide values (or change the default values) for the fields in this section of the window. For example, if your SAP configuration uses load balancing, provide values for the **Message server host** field, **Logon group name** and **SAP system ID**. Select the **Reset the JCO client after closing connection handle** check box if during an outbound transaction, you want the adapter to make sure changes on the SAP EIS are reflected in the client. For Query interface transactions, the property does not have a functionality impact. It is recommended that you do not select the check box.
 - c. Optionally, expand **Advanced Security Configuration** to specify your security mechanism. You can enable both **Secure Network Connection (SNC)** and **Single Sign On (SSO)**.
 - If you are using Secure Network Connection, select **Enable Secure Network Connection**. Enter information in the associated fields (name, partner, security level, and library path). If you select X509 certificate based authentication, the X509 certificate name has to be entered. Secure Network Connection has to be enabled to use X509 certificate based authentication.
 - If you are using Single Sign On authentication, select **Enable Single Sign On (SSO)**. Enter the SSO Logon Ticket in the field.
 - d. Optionally to set RFC and JCo tracing properties, perform the following steps:
 - 1) Expand **SAP RFC trace configuration** and select **RFC trace on**.
 - 2) Select the **Write JCo traces into adapter logs** checkbox to enable JCo API traces to be generated in the broker trace file. These traces will then be interleaved with the adapter traces. Each JCo API trace logged into the broker trace file will have the string [JCoAPI]. All JCo API traces are logged at the INFO message log level.
 - 3) Select a tracing level from the **RFC trace level** list.
 - 4) Click **Browse** and select a location to which the RFC trace files will be saved.
 - e. Optionally expand **Logging and tracing** and specify the following information:
 - If you have multiple instances of the adapter, set **Adapter ID** to a value that is unique for this instance.
 - If you want to mask sensitive information in log and trace files (for example, if you want to avoid making customer information visible in these files), select **Disguise user data as "XXX" in log and trace files**.

See "Managed connection factory properties" on page 334 for information about these optional properties.
7. Click **Next**. The Service Location Properties window opens.
 8. Create a module.

- a. In the Specify the Location Properties window, click **New** in the **Module** field.
 - b. In the Integration Project window, click **Create a module project** or **Create a mediation module project** and click **Next**.
9. In the New Module window, perform the following tasks:
- a. Type a name for the module.
As you type the name, it is added to the workplace specified in the **Location** field.
This is the default location. If you want to specify a different location, remove the check from **Use default location** and type a new location or click **Browse** and select the location.
 - b. Specify whether you want to open the module in the assembly diagram (for module projects) or whether you want to create a mediation flow component (for mediation module projects). By default, these choices are selected.
 - c. Click **Finish**.
10. In the Specify the Location Properties window, perform the following steps:
- a. If you want to change the default namespace, clear the **Use default namespace** check box and type a new path in the **Namespace** field.
 - b. 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.
This is an optional step.
 - c. Optionally change the name of the interface.
The default name is `SAPOutboundInterface`. You can change it to a more descriptive title if you prefer.
 - d. If you want to save the business objects so that they can be used by another application, click **Save business objects to a library** and then select a library from the list or click **New** to create a new library.
 - e. Optionally type a description of the module.
11. Click **Finish**.

Results

The new module is added to the Business Integration perspective.

What to do next

Export the module as an EAR file for deployment.

Configuring a module for Advanced event processing - outbound

To configure a module to use the adapter for Advanced event processing, you use the external service wizard in IBM Integration Designer to discover IDocs on the SAP server. You then configure the business objects that are generated and create a deployable module. To use the Advanced event processing interface, you must first add the adapter-supplied transport files to the SAP server.

Selecting business objects and services for Advanced event (outbound) processing

To specify which function you want to process, you provide information in the external service wizard.

Before you begin

Make sure you have set the connection properties for the external service wizard.

About this task

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

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

Procedure

1. In the Find Objects in the Enterprise System window, indicate which IDoc you want to work with.
 - a. Expand **AEP**.
 - b. Click **Discover IDoc From System** to enable the filter



button.

- c. Click the filter button.
- Note:** Instead of using the filter button, you can expand **Discover IDoc From System** and select the IDoc from the list. Then skip ahead to step 4.
2. From the Filter Properties window, specify information about the IDoc or IDocs:
 - a. Select **Discover objects by name** or **Discover objects by description** from the **Object attribute to use for discovery** list.
 - b. Type a search string representing the IDoc you want to call.
 - c. Select **Basic IDocs** or **Extension IDocs** from the **IDoc type to use for discovery** field.
 - d. Indicate the number of functions you want returned by changing the value in the **Maximum number of objects to retrieve** field or by accepting the default value.
 - e. Click **OK**.
 3. Select the IDoc or IDocs.
 - a. Expand **Discover IDoc From System (filtered)**.
 - b. From the Discovered object list, click the IDoc you want to use. If you are working with multiple IDocs, click the names of all the IDocs.
 4. Click the arrow button to add the IDoc or IDocs to the **Selected objects** list.
 5. In the Configuration Parameters window, perform the following steps to add the IDoc to the list of business objects to be imported.
 - a. Optionally, select the **Use SAP field names to generate attribute names** check box. By default (when the check box is not selected), the field descriptions are used to generate properties.
 - b. In the **IDoc release version** field, specify the SAP release number to identify the IDoc type you want the external service wizard to use for creating business objects.
 - c. Expand the IDoc name and select one or more nodes to be used as the primary key, or leave the default values selected.

- d. Click **OK**.
6. Click **Next**.

Results

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

What to do next

From the Specify Composite Properties window, select an operation for the IDoc and an ABAP function module for that operation. Optionally specify a namespace and directory to which the generated business object will be stored and indicate whether you want a business graph generated.

Configuring the selected objects

To configure the object, you associate an operation with the IDoc and associate an ABAP function module with the selected operation.

Before you begin

Make sure you have selected and imported the function.

About this task

To configure the business object, use the following procedure.

Procedure

1. In the Configure Composite Properties window, click an IDoc from the **IDoc to configure** list.
If you are configuring only one IDoc, this step is not necessary.
2. Click **Add** in the Service operations for selected IDoc section of the window.
3. Select an operation (for example, **Retrieve**), and click **OK**.
4. In the **ABAP function module name for selected operations** field, type the name of the ABAP function module to associate with this operation.

Note: The ABAP function module must have been created and must exist on the SAP server.

5. If you are working with multiple IDocs, repeat the previous four steps for each IDoc.
6. In the **Business object 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 external service wizard), change the namespace value.
For example, you could change the namespace to <http://www.ibm.com/xmlns/prod/websphere/j2ca/sap1>.
7. To indicate where the business object information should be stored, type the path to the location in the **Folder** field. This is an optional step.

Note: This field cannot be edited if you are modifying existing artifacts.

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

8. If you want the IDoc or IDocs to be enclosed within a business graph, leave **Generate a business graph for each business object** selected. Otherwise, remove the check.
9. Click **Finish**.

Results

You have associated an operation with each IDoc and have associated an ABAP function module with each operation. The Service Generation and Deployment Configuration window is displayed.

What to do next

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

Setting deployment properties and generating the service

To generate the module, which is the artifact that is deployed on IBM Business Process Manager 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 Specify the Service Generation and Deployment Properties window should be displayed.

If you choose to use X509 certificate authentication (or) Single Sign On authentication (or) Secure Network Communication (SNC) to authenticate with SAP using the adapter, the target SAP Server has to be configured for the selected authentication mechanism. You must also obtain the required information for configuring the authentication mechanism from SAP (for example, SSO token, X509 certificate, SNC name, and SNC partner name).

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.

Procedure

1. 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**.
2. Indicate whether you will use an authentication alias (instead of typing a user ID and password) to establish a connection to the SAP server:
 - To specify an authentication alias, leave **Specify a Java Authentication and Authorization Services (JAAS) alias security credential** selected. Then, in

the **J2C Authentication Data Entry** field, enter the name you specified in the Security section of the administrative console.

If you are not going to use an authentication alias, clear **Specify a Java Authentication and Authorization Services (JAAS) alias security credential**.

- To use a user ID and password, select **Using security properties from the managed connection factory**. Type in the user name and password to specify your security credentials.
 - To use other security mechanisms native to the enterprise system or if security is not required, select **Other**
3. Select **With module for use by single application** to embed the adapter files in a module that is deployed to the application server, or select **On server for use by multiple applications** to install the adapter files as a stand-alone adapter.

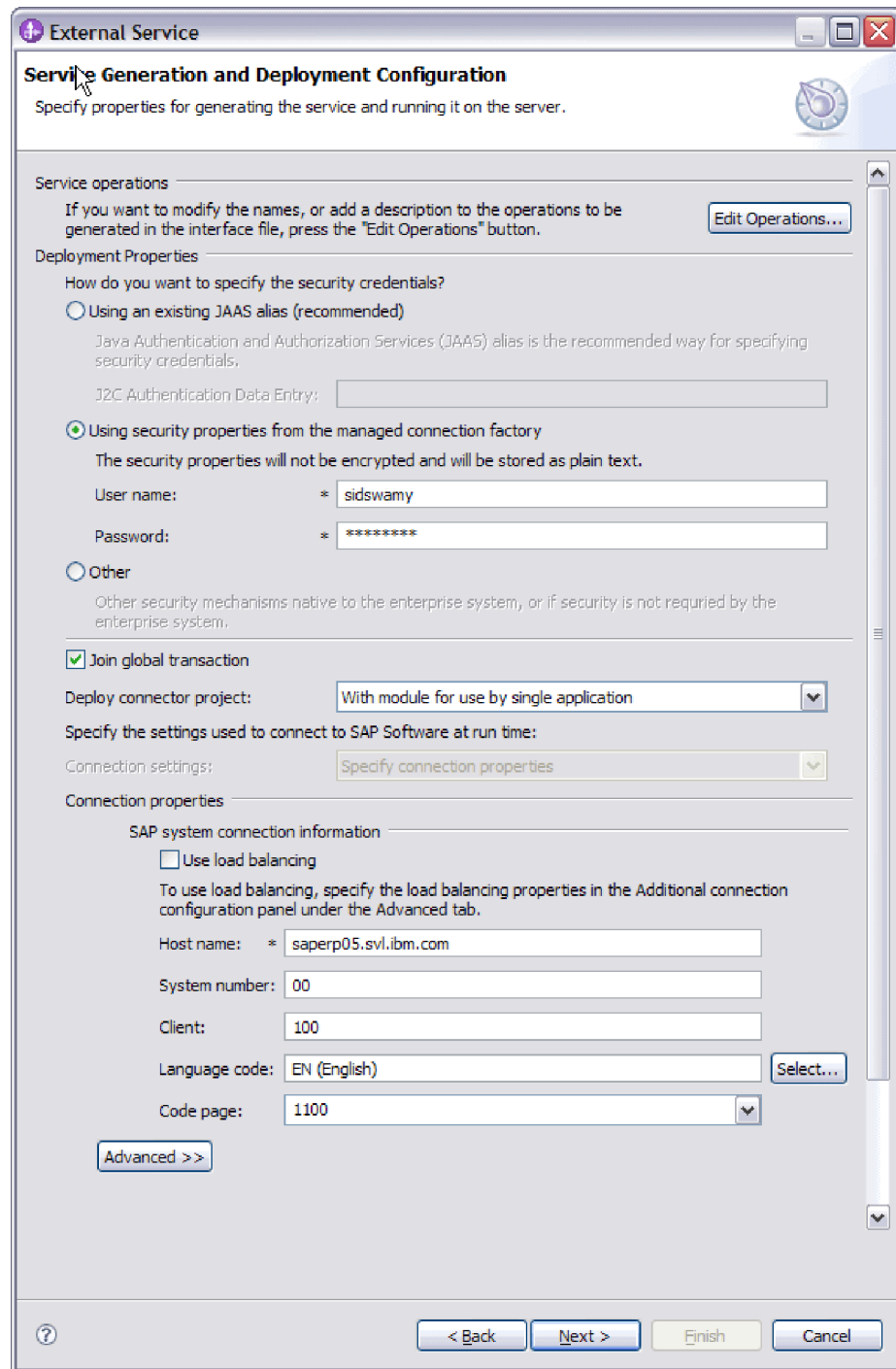


Figure 74. Connection properties

- **With module for use by single application:** With the adapter files embedded in the module, you can deploy the module to any application server. Use an embedded adapter when you have a single module using the adapter or if multiple modules need to run different versions of the adapter. Using an embedded adapter enables you to upgrade the adapter in a single module without the risk of destabilizing other modules by changing their adapter version.
- **On server for use by multiple applications:** If you do not include the adapter files in a module, you must install them as a stand-alone adapter

on each application server where you want to run the module. Use a stand-alone adapter when multiple modules can use the same version of the adapter and you want to administer the adapter in a central location. A stand-alone adapter can also reduce the resources required by running a single adapter instance for multiple modules.

4. If you selected **On server for use by multiple applications** in the previous step, the **Connection properties** list becomes active. Make one of the following selections:

- Select **Specify connection properties** if you want to provide configuration information now. Then continue with step 5.
- Select **Use predefined connection properties** if you want to use a connection factory configuration that already exists.

If you decide to use predefined connection properties, you must ensure that your resource adapter name matches the name of the installed adapter, because this is how the adapter instance is associated with these properties. If you want to change the resource adapter name in the import or export, use the assembly editor in IBM Integration Designer to change the value in the import or export.

When you select **Use predefined connection properties**, the **JNDI Lookup Name** field is displayed in place of the properties.

- a. Type a value for **JNDI Lookup Name**.
- b. Click **Next**.
- c. Go to step 8 on page 178.

5. In the Connection properties section, set or change any connection properties that apply to your configuration.

Notice that some of the values are already filled in. For example, the values that you used in the Specify the Discovery Properties window (such as the **Host name**) are filled in.

Select the **Use load balancing** check box to use load balancing to connect to the SAP system. Load balancing properties which are **Message server host**, **Logon group name** and **SAP system ID** have to be specified in the Additional connection configuration panel under the Advanced tab.

See “Managed connection factory properties” on page 334 for more information about these properties.

Properties marked with an asterisk (*) are required.

6. To set additional properties, click **Advanced**.

- a. Expand **Processing methods for illegal XML characters**

- 1) From the **Illegal XML character processing** drop-down list, select one of the options that correspond to your requirement:

- **Do not validate illegal XML characters**, if you want to work with the default adapter behavior.
- **Throw exception if contents in the BO contains illegal XML characters**, if you want to proceed after you get an exception message, and log the illegal XML characters in the trace file, at runtime.
- **Discard all illegal XML characters and related logs**, if you want the adapter to discard the illegal XML characters, and log the characters in the trace file, at runtime.

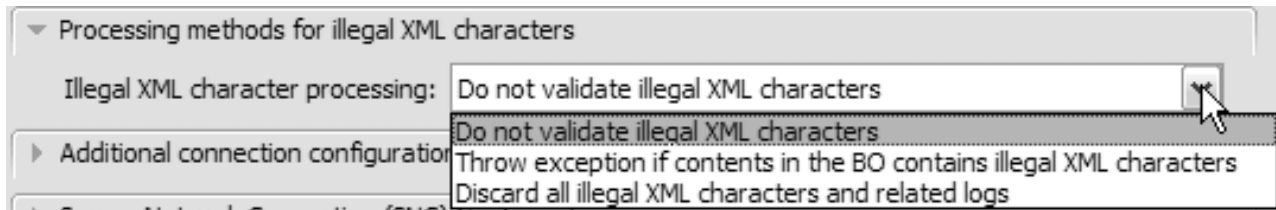


Figure 75. Processing methods for illegal XML characters

- b. Optionally expand **Advanced connection configuration** and provide values (or change the default values) for the fields in this section of the window. For example, if your SAP configuration uses load balancing, provide values for the **Message server host** field, **Logon group name** and **SAP system ID**.
- c. Optionally expand **Additional connection configuration** and provide values (or change the default values) for the fields in this section of the window. For example, if your SAP configuration uses load balancing, provide values for the **Message server host** field, **Logon group name** and **SAP system ID**. Select the **Reset the JCO client after closing connection handle** check box if during an outbound transaction, you want the adapter to make sure changes on the SAP EIS are reflected in the client. For AEP transactions the property does not have a functionality impact. It is recommended that you do not select the check box.
- d. Optionally, expand **Advanced Security Configuration** to specify your security mechanism. You can enable both **Secure Network Connection (SNC)** and **Single Sign On (SSO)**.
 - If you are using Secure Network Connection, select **Enable Secure Network Connection**. Enter information in the associated fields (name, partner, security level, and library path). If you select X509 certificate based authentication, the X509 certificate name has to be entered. Secure Network Connection has to be enabled to use X509 certificate based authentication.
 - If you are using Single Sign On authentication, select **Enable Single Sign On (SSO)**. Enter the SSO Logon Ticket in the field.
- e. Optionally to set RFC and JCo tracing properties, perform the following steps:
 - 1) Expand **SAP RFC trace configuration** and select **RFC trace on**.
 - 2) Select the **Write JCo traces into adapter logs** checkbox to enable JCo API traces to be generated in the broker trace file. These traces will then be interleaved with the adapter traces. Each JCo API trace logged into the broker trace file will have the string [JCoAPI]. All JCo API traces are logged at the INFO message log level.
 - 3) Select a tracing level from the **RFC trace level** list.
 - 4) Click **Browse** and select a location to which the RFC trace files will be saved.
- f. Optionally expand **Logging and tracing** and specify the following information:
 - If you have multiple instances of the adapter, set **Adapter ID** to a value that is unique for this instance.
 - If you want to mask sensitive information in log and trace files (for example, if you want to avoid making customer information visible in these files), select **Disguise user data as "XXX" in log and trace files**.

See “Managed connection factory properties” on page 334 for more information about these optional properties.

7. Click **Next**. The Specify the Location Properties window opens.
8. Create a module.
 - a. In the Specify the Location Properties window, click **New** in the **Module** field.
 - b. In the Integration Project window, click **Create a module project** or **Create a mediation module project** and click **Next**.
9. In the New Module window, perform the following tasks:
 - a. Type a name for the module.

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

This is the default location. If you want to specify a different location, remove the check from **Use default location** and type a new location or click **Browse** and select the location.
 - b. Specify whether you want to open the module in the assembly diagram (for module projects) or whether you want to create a mediation flow component (for mediation module projects). By default, these choices are selected.
 - c. Click **Finish**.
10. In the Specify the Location Properties window, perform the following tasks:
 - a. If you want to change the default namespace, clear the **Use default namespace** check box and type a new path in the **Namespace** field.
 - b. 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. This is an optional step.
 - c. Optionally change the name of the interface.

The default name is `SAPOutboundInterface`. You can change it to a more descriptive title if you prefer.
 - d. If you want to save the business objects so that they can be used by another application, click **Save business objects to a library** and then select a library from the list or click **New** to create a new library.
 - e. Optionally type a description of the module.
11. Click **Finish**.

Results

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 module for inbound processing

To configure a module to use the adapter for inbound processing, use the external service wizard in IBM Integration Designer to find and select business objects and services from the SAP server, and to generate business object definitions and related artifacts.

Before you begin

The following table outlines the properties that are applicable for an interface:

Table 19.

Interface	SAP RFC Trace Configuration	Bidi properties	Business Objects naming convention	
			SAP naming convention	Enforce naming convention for business objects
BAPI	Yes	Yes	Yes	Yes
ALE	Yes	Yes	Yes	No
ALE Pass-through Idoc	Yes	Yes	No	No
Advanced Event Process (AEP)	Yes	Yes	No	No

Configuring a module for BAPI inbound processing

To configure a module to use the adapter for BAPI inbound processing, you use the external service wizard in IBM Integration Designer to find RFC-enabled functions. You then configure the business objects that are generated and create a deployable module.

Selecting business objects and services for BAPI inbound processing

To specify which function you want to process, you provide information in the external service wizard.

Before you begin

Make sure you have set the connection properties for the external service wizard.

About this task

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

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

Procedure

1. In the Find Objects in the Enterprise System window, indicate which BAPI or set of BAPIs you want to work with.
 - a. Click **RFC** to enable the filter



button.

- b. Click the filter button.

Note: Instead of using the filter capability, you can expand **RFC** and select the function from the list, or you can expand **BOR**, expand the functional grouping (for example, **Cross-Application Components**), and select the BAPI. You can now also filter BOR child objects. Then skip ahead to step 4.

2. From the Filter Properties window, specify information about the BAPI or BAPIs you want to discover:
 - a. Select **Discover objects by name** or **Discover objects by description** from the **Object attribute to use for discovery** list.
 - b. 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.
 - c. Indicate the number of functions you want returned by changing the value in the **Maximum number of objects to retrieve** field or by accepting the default value.
 - d. Click **OK**.
3. Select the BAPI or BAPIs.
 - a. Expand **RFC (filtered)**.
 - b. From the Discovered object list, select one or more BAPIs that you want to use.
4. Click the arrow button to add the BAPI or BAPIs to the **Selected objects** list.
5. In the Configuration Parameters window, perform the following steps for each BAPI to add it to the list of business objects to be imported:
 - a. Optionally, select the **Use SAP field names to generate attribute names of business objects** check box to generate business object attribute names using the original SAP field names without casing. If you want to generate business object attribute names using the original SAP field name casing, check the **Use the original character case of SAP field names to generate business object attribute names** check box. By default (when the **Use SAP field names to generate attribute names of business objects** check box is not selected), field descriptions are used to generate properties.

Note: This option is not available when you select the **Generate business objects according to SAP naming conventions** option in the Specify the Discovery Properties window.

- b. Optionally, select the **Process Date fields as String** check box. This allows the adapter to retrieve SAP-specific date formats which are otherwise non-standard in Java by using the String type instead of the Date type format.

Note: To use this feature, ensure that the above check box is selected while running the external service wizard. As the date fields are converted to string type, ensure that all downstream dependencies (if any) are taken care of.

- c. If the BAPI has optional parameters associated with it, select the **Select optional parameters to include as child objects** check box, expand **Optional parameters**, and select the type of parameters (import, export, or table) that you want to work with.

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

If you are adding BAPI_CUSTOMER_GETDETAIL function module, you have the option of adding the following parameters:

Optional Import Parameters

PI_PASS_BUFFER

PI_DIVISION

PI_DISTR_CHAN

Optional Export Parameters

PE_ADDRESS

RETURN

Refer to the SAP documentation for a list and description of the optional parameters.

d. 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 the left arrow.

6. Click **Next**

Results

The external service wizard has returned the function or list of functions that match the search criteria, and you have selected the function or functions you want to work with. The Specify Composite Properties window is displayed.

What to do next

Provide information about the business object (such as the operation associated with the object and the SAP remote function call type).

Configuring the selected objects

To configure the object, you specify information about the object (such as the operation associated with the object and the type of remote function call).

Before you begin

If you are sending the function call from a queue on the SAP server (which ensures the order in which BAPIs are delivered), make sure you have configured an outbound queue on the SAP server. You also need an ABAP program on the SAP server that delivers the BAPI events to the outbound queue.

About this task

During configuration of the object, you select which type of remote function call you want to make. You can select **Synchronous RFC** (the default) or **Asynchronous Transactional/Queued RFC**.

- Use **Synchronous RFC** when you want to wait for a response from the endpoint. The endpoint must be available when you send the function call from the SAP server to the adapter.
- Use **Asynchronous Transactional/Queued RFC** in the following circumstances:

- When you are sending a function call from a queue on the SAP server to the adapter
- When you want the function call to succeed regardless of whether the endpoint is available at the time of the call.

To configure the business object, use the following procedure.

Procedure

1. In the Specify Composite Properties window, select an operation for each BAPI you selected in the previous task.
 - If you are working with one BAPI, select an operation for that BAPI from the **Operations** list.
 - If you are working with multiple BAPIs, select an operation for each BAPI from the list next to the name of the BAPI. Make sure you select one operation for each BAPI.

2. In the **Business object 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 external service wizard), change the namespace value.

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

3. To indicate where the business object information should be stored, type the path to the location in the **Folder** field. This is an optional step.

Note: The above two fields cannot be edited if you are modifying existing artifacts.

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

4. If you want the BAPI or BAPIs to be enclosed within a business graph, leave **Generate a business graph for each business object** selected. Otherwise, remove the check.
5. Select the type of remote function call you want to make.

Note: If you do not select a type of remote function call, the default (**Synchronous RFC**) is used. In Synchronous RFC, the SAP server sends the BAPI and then waits for the response from the endpoint.

- a. Select the arrow next to the **SAP Remote Function Call (RFC) type** list.

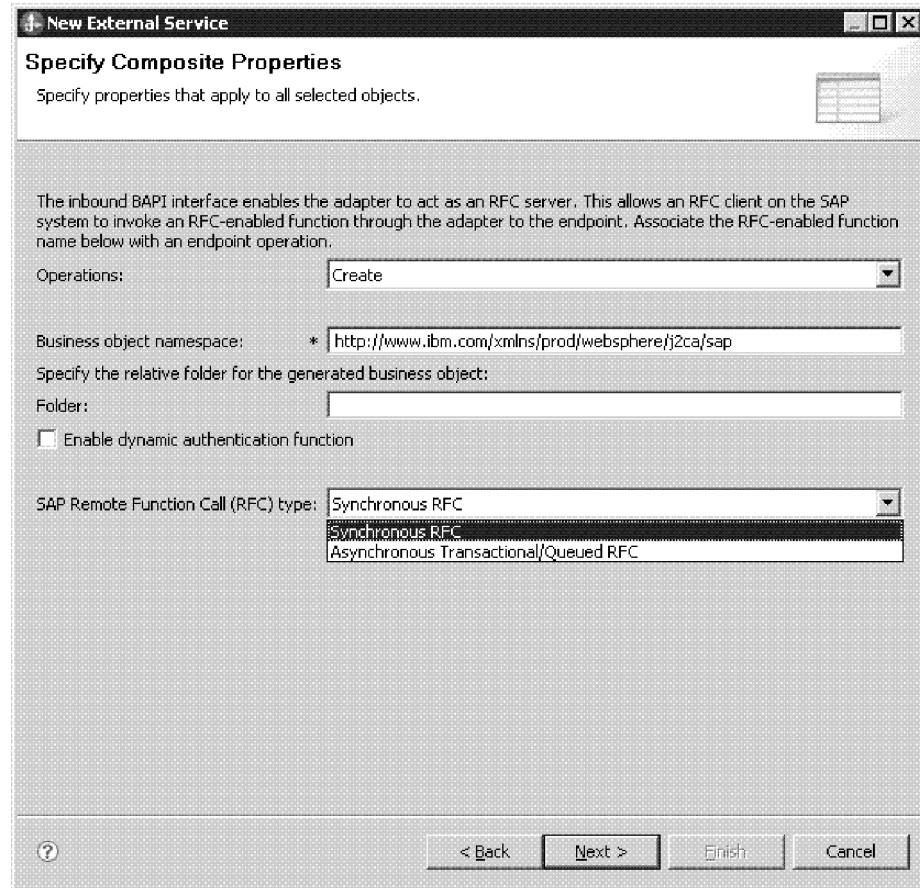


Figure 76. SAP Remote Function Call (RFC) type list

b. Select one of the RFC types:

- Select **Asynchronous Transactional/Queued RFC** when you are sending the function call from a queue on the SAP server or if you want the call to succeed regardless of whether the receiving system (the endpoint) is available.
 - If the adapter is available, the call succeeds.
 - If the adapter is not available, the SAP server continues to attempt to make the call until the adapter is available. The SAP system ensures that the call is invoked only once. A transaction ID (TID) is associated with the BAPI.
- Select **Synchronous RFC** (the default) if you want the BAPI sent in a synchronous manner (the SAP server sends the BAPI and then waits for the response from the endpoint). Note that the endpoint must be available when you use **Synchronous RFC**.

Note: This field cannot be edited if you are modifying existing artifacts.

6. Click **Next**.

Results

You selected an operation for each BAPI. The Specify the Service Generation and Deployment Properties window is displayed.

What to do next

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

Setting deployment properties and generating the service

To generate the module, which is the artifact that is deployed on IBM Business Process Manager 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 Specify the Service Generation and Deployment Properties 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.

Procedure

1. 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**.
2. Indicate whether you will use an authentication alias (instead of typing a user ID and password) to establish a connection to the SAP server:
 - To specify an authentication alias, leave **Specify a Java Authentication and Authorization Services (JAAS) alias security credential** selected. Then, in the **J2C Authentication Data Entry** field, enter the name you specified in the Security section of the administrative console.
If you are not going to use an authentication alias, clear **Specify a Java Authentication and Authorization Services (JAAS) alias security credential**.
 - To use a user ID and password, select **Using security properties from the activation specification**. Type in the user name and password to specify your security credentials.
 - To use other security mechanisms native to the enterprise system or if security is not required, select **Other**
3. Select **With module for use by single application** to embed the adapter files in a module that is deployed to the application server, or select **On server for use by multiple applications** to install the adapter files as a stand-alone adapter.

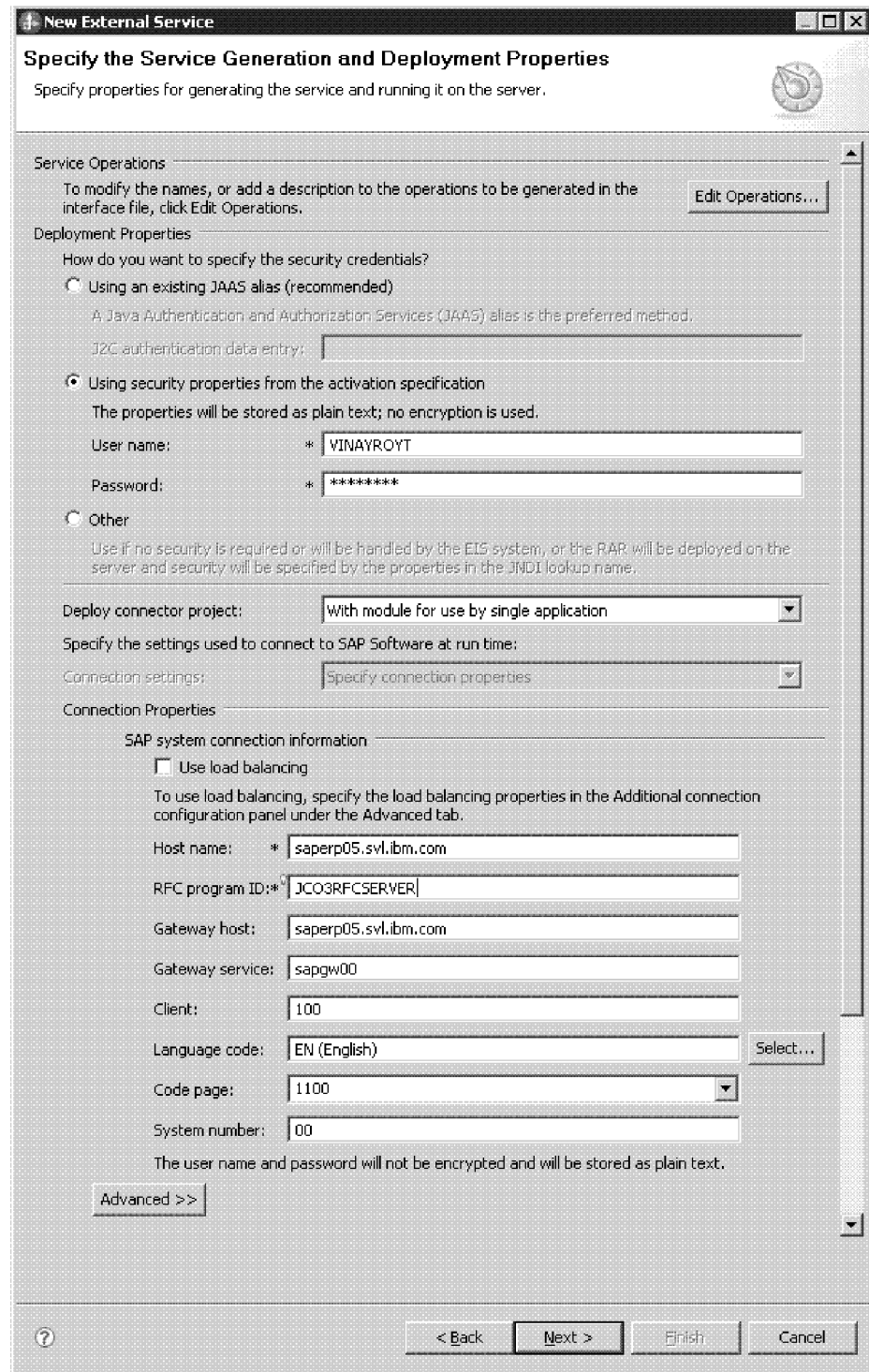


Figure 77. Specify the Service Generation and Deployment Properties window

- With module for use by single application:** With the adapter files embedded in the module, you can deploy the module to any application server. Use an embedded adapter when you have a single module using the adapter or if multiple modules need to run different versions of the adapter. Using an embedded adapter enables you to upgrade the adapter in a single module without the risk of destabilizing other modules by changing their adapter version.

- **On server for use by multiple applications:** If you do not include the adapter files in a module, you must install them as a stand-alone adapter on each application server where you want to run the module. Use a stand-alone adapter when multiple modules can use the same version of the adapter and you want to administer the adapter in a central location. A stand-alone adapter can also reduce the resources required by running a single adapter instance for multiple modules.
4. If you selected **On server for use by multiple applications** in the previous step, the **Connection properties** list becomes active. Make one of the following selections:
 - Select **Specify connection properties** if you want to provide configuration information now. Then continue with step 5.
 - Select **Use predefined connection properties** if you want to use an activation specification configuration that already exists.

If you decide to use predefined connection properties, you must ensure that your resource adapter name matches the name of the installed adapter, because this is how the adapter instance is associated with these properties. If you want to change the resource adapter name in the import or export, use the assembly editor in IBM Integration Designer to change the value in the import or export.

When you select **Use predefined connection properties**, the **JNDI Lookup Name** field is displayed in place of the properties.

 - a. Type a value for **JNDI Lookup Name**.
 - b. Click **Next**.
 - c. Go to step 7 on page 188.
 5. In the Connection properties section, set or change any connection properties that apply to your configuration.

Notice that some of the values are already filled in. For example, the values that you used in the Specify the Discovery Properties window (such as the **Host name**) are filled in.

 - a. Select the **Use load balancing** check box to use load balancing to connect to the SAP system. Load balancing properties which are **Message server host**, **Logon group name** and **SAP system ID** have to be specified in the Additional connection configuration panel under the Advanced tab.
 - b. Change the **Host name** field if you are planning to send events from a different SAP server than the one you are using to create the adapter module.
 - c. In the **RFC Program ID** field, type the name of the program ID you registered with the SAP server.
 - d. The **Gateway host** is filled in, by default, with the value from the **Host name** field.
 - e. A default value of **sapgw00** is filled in for **Gateway service**. If you have more than one gateway server in your SAP configuration, change **sapgw00** to the correct value.
 - f. The remaining values in the SAP system connection information section are filled in with the values you entered in the Specify the Discovery Properties window. Change these values, if necessary.

See “Activation specification properties for BAPI inbound processing” on page 366 for more information about these properties.

Properties marked with an asterisk (*) are required.
 6. To set additional properties, click **Advanced**.

Properties marked with an asterisk (*) are required.

a. Expand **Processing methods for illegal XML characters**

1) From the **Illegal XML character processing** drop-down list, select one of the options that correspond to your requirement:

- **Do not validate illegal XML characters**, if you want to work with the default adapter behavior.
- **Throw exception if contents in the BO contains illegal XML characters**, if you want to proceed after you get an exception message, and log the illegal XML characters in the trace file, at runtime.
- **Discard all illegal XML characters and related logs**, if you want the adapter to discard the illegal XML characters, and log the characters in the trace file, at runtime.

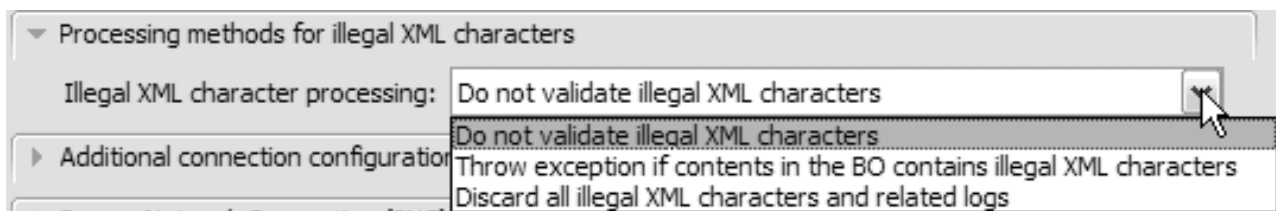


Figure 78. Processing methods for illegal XML characters

- b. Optionally expand **Additional connection configuration** and provide values (or change the default values) for the fields in this section of the window. For example, if your SAP configuration uses load balancing, provide values for the **Message server host** field, **Logon group name** and **SAP system ID**.
- c. You can expand **Event polling configuration** and specify values indicating how events should be polled on the SAP server. For example, you can enter a list of event types in the **Event types to process** field if you want to limit the events that the adapter should process. You can select **Retry EIS connection on startup** if you want the adapter to retry a failed connection when starting. For more information, see "Retry EIS connection on startup" on page 377.
- d. You can expand **Event delivery configuration** if you want to change the default values for the way events are delivered. Then enter a value (or change the default value) in one or more fields. For example, you can change the number of times the SAP server attempts to deliver a failed event.
See "Activation specification properties for BAPI inbound processing" on page 366 for more information about these properties.
- e. If you are using Secure Network Connection, expand **Secure Network Connection (SNC) Configuration** and select **Enable Secure Network Connection**. Then enter information in the associated fields (name, partner, security level, and library path). Optionally type the name of an X509 certificate.
- f. Optionally set a comma-separated list of Interceptor classes in the **List of Interceptors** field. These Interceptor classes act as User Exits and must implement the

com.ibm.j2ca.extension.monitoring.interceptors.Interceptor Java interface. Refer to User Exits through Interceptors for more details on Interceptors.

- g. Optionally to set RFC and JCo tracing properties, perform the following steps:
 - 1) Expand **SAP RFC trace configuration** and select **RFC trace on**.
 - 2) Select the **Write JCo traces into adapter logs** checkbox to enable JCo API traces to be generated in the broker trace file. These traces will then be interleaved with the adapter traces. Each JCo API trace logged into the broker trace file will have the string [JCoAPI]. All JCo API traces are logged at the INFO message log level.
 - 3) Select a tracing level from the **RFC trace level** list.
 - 4) Click **Browse** and select a location to which the RFC trace files will be saved.
 - h. Optionally expand **Logging and tracing** and specify the following information:
 - If you have multiple instances of the adapter, set **Adapter ID**, to a value that is unique for this instance.
 - If you want to mask sensitive information in log and trace files (for example, if you want to avoid making customer information visible in these files), select **Disguise user data as "XXX" in log and trace files**.
7. Create a module.
- a. In the Specify the Location Properties window, click **New** in the **Module** field.
 - b. In the Integration Project window, click **Create a module project** or **Create a mediation module project** and click **Next**.
8. In the New Module window, perform the following tasks:
- a. Type a name for the module.

As you type the name, it is added to the workspace specified in the **Location** field.

This is the default location. If you want to specify a different location, remove the check from **Use default location** and type a new location or click **Browse** and select the location.
 - b. Specify whether you want to open the module in the assembly diagram (for module projects) or whether you want to create a mediation flow component (for mediation module projects). By default, these choices are selected.
 - c. Click **Finish**.
9. In the Specify the Location Properties window, perform the following tasks:
- a. If you want to change the default namespace, clear the **Use default namespace** check box and type a new path in the **Namespace** field.
 - b. 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. This is an optional step.
 - c. Optionally change the name of the interface.

The default name is `SAPInboundInterface`. You can change it to a more descriptive title if you prefer.
 - d. If you want to save the business objects so that they can be used by another application, click **Save business objects to a library** and then select a library from the list or click **New** to create a new library.

- e. Optionally type a description of the module.
10. Click **Finish**.

Results

The new module is added to the Business Integration perspective.

What to do next

Export the module as an EAR file for deployment.

Configuring a module for ALE inbound processing

To configure a module to use the adapter for ALE inbound processing, you use the external service wizard in IBM Integration Designer to find an IDoc or set of IDocs, configure the business objects that are generated, and create a deployable module. If you are going to set up an event recovery table to persist inbound events (to ensure once-only delivery of events), you must also set up a data source.

Selecting business objects and services for ALE inbound processing

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

About this task

For the ALE interface, you can select IDocs in one of two ways:

- You can specify an IDoc or a set of IDocs by entering search criteria (such as the name of the IDoc) and having the external service wizard search the SAP system.
- You can enter an IDoc definition file name with the complete path to its location on the file system.

If you choose to discover IDocs from a file, you must first configure the file. The file is generated from information on the SAP server and is then saved to your local file system.

For the ALE pass-through IDoc interface, you can specify IDocs from a system or from a file, as described in the previous section. Additionally, you can select a generic IDoc.

When you select a generic IDoc, you create one business-object definition that can apply to any IDoc at run time. This selection is helpful if you are processing many IDocs and do not want to create a separate business-object definition for each one.

Discovering IDocs from the system:

Use the **Discover IDocs from System** option to have the external service wizard search for IDocs based on the criteria you specify.

Before you begin

Make sure you have set the connection properties for the external service wizard.

About this task

Specify search criteria that the external service wizard uses to discover IDocs on the SAP server.

Note: The **Discover IDoc From System** choice applies to both the ALE interface and the ALE pass-through IDoc interface.

Procedure

1. In the Find Objects in the Enterprise System window, indicate which IDoc you want to work with.
 - a. Expand **ALE**.
 - b. Click **Discover IDoc From System** to enable the filter



button.

- c. Click the filter button.

Note: Instead of using the filter button, you can expand **Discover IDoc From System** and select the IDoc from the list. You then skip ahead to step 4.

2. From the Filter Properties window, specify information about the IDoc or IDocs:
 - a. Select **Discover objects by name** or **Discover objects by description** from the **Object attribute to use for discovery** list.
 - b. Type a search string (for example, ALEREQ*) representing the IDoc you want to call.

This is the name of the IDoc in SAP plus an asterisk as a wild card character to indicate that you want a list of all IDocs that start with ALEREQ.
 - c. Select **Basic IDocs** or **Extension IDocs** from the **IDoc type to use for discovery** field.
 - d. Indicate the number of functions you want returned by changing the value in the **Maximum number of objects to retrieve** field or by accepting the default value.
 - e. Click **OK**.
3. Select the IDoc or IDocs.
 - a. Expand **Discover IDoc From System (filtered)**.
 - b. From the Discovered object list, click the IDoc you want to use. If you are working with multiple IDocs, click the names of all the IDocs.
4. Click the arrow button to add the IDoc or IDocs to the **Selected objects** list.
5. In the Configuration Parameters window, perform the following tasks to add the IDoc to the list of business objects to be imported.

Note: If you are using the ALE pass-through IDoc interface, only the **Send an IDoc Packet as one business object** configuration property is available.

- a. Optionally select the **Use SAP field names to generate attribute names**. By default, when the check box is not selected, field descriptions are used to generate properties. If you choose to use SAP field names to generate attribute names, two more check boxes are enabled:

- b. Select the **Use SAP-original casing for Control Record Business object attribute names** check box to generate attribute names in SAP original casing. If unchecked, attribute names are generated using field names from SAP in camel-casing.
- c. Select the **Use SAP-original casing for Data Record Business object attribute names** check box to generate attribute names in SAP original casing. If unchecked, attribute names are generated using field Name from SAP in camel-casing.

The following are the different possible combinations:

Table 20.

Scenario	Use SAP field names to generate attribute names (check-box)	Use SAP-original casing Control Record business object attribute names (check-box)	Use SAP-original casing Data Record business object attribute names (check-box)	Control Record	Data Record
1	Checked	Checked	Checked	Attribute names are generated using Field Name from SAP (Original SAP casing)	Attribute names are generated using Field Name from SAP (Original SAP casing)
2	Checked	Checked	Not checked	Attribute names are generated using Field Name from SAP (Original SAP casing)	Attribute names are generated using Field Name from SAP (Camel-casing)
3	Checked	Not checked	Checked	Attribute names are generated using Field Name from SAP (Camel-casing)	Attribute names are generated using Field Name from SAP (Original SAP casing)
4	Checked	Not checked	Not checked	Attribute names are generated using Field Name from SAP (Camel-casing)	Attribute names are generated using Field Name from SAP (Camel-casing)

Table 20. (continued)

Scenario	Use SAP field names to generate attribute names (check-box)	Use SAP-original casing Control Record business object attribute names (check-box)	Use SAP-original casing Data Record business object attribute names (check-box)	Control Record	Data Record
5	Not checked	Not checked	Not checked	Attribute names are generated using Field Description from SAP (Camel-casing)	Attribute names are generated using Field Description from SAP (Camel-casing)

- d. If you are working with an IDoc packet and want to specify that the packet not be split, select the **Send an IDoc Packet as one business object** check box.
- e. 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 an IDoc packet as unparsed data** check box.

Note: Refer to Resolving data record (hexbinary) format issues for ALE inbound interfaces for more information on resolving data record format issues for ALE inbound interfaces.

- f. In the **IDoc release version** field, specify the SAP release number to identify the IDoc type you want the external service wizard to use for creating business objects. If the selected IDoc has unreleased segments, the **IDoc release version** property becomes required. It is recommended that you select the default value **Unreleased** if the IDoc you are working with has unreleased segments. If **Unreleased** is selected, the adapter generates the business objects for segments using the unreleased segment definition.
 - g. In the **IDoc release version** field, specify the SAP release number to identify the IDoc type you want the external service wizard to use for creating business objects. The version of the SAP system is set as default. If the selected IDoc has unreleased segments, the **IDoc release version** property becomes required. It is recommended that you select **Unreleased** if the IDoc you are working with has unreleased segments. If **Unreleased** is selected, the adapter generates the business objects for segments using the unreleased segment definition.
 - h. Click **OK**.
6. Click **Next**.

Results

The external service wizard has returned an IDoc or a list of IDocs, and you have selected the ones you want to work with. You see the Configure Composite Properties window (if you are using the ALE interface) or the Service Generation and Deployment Configuration (if you are using the ALE pass-through IDoc interface).

What to do next

- If you are using the ALE interface, you can optionally specify a namespace and directory to which the generated business object will be stored and indicate whether you want a business graph generated, as described in Configuring the selected objects.
- If you are using the ALE pass-through IDoc interface, you generate a deployable module that includes the adapter and the business objects, as described in “Setting deployment properties and generating the service” on page 200.

Discovering IDocs from a file:

To select IDocs from a file, you must first configure an IDoc definition file based on information on the SAP server. You then specify, in the external service wizard, the path to the file on your local system.

Before you begin

You must have created an IDoc definition file.

Note: If you are using **Discover IDoc From System**, do not complete the following steps. The IDoc definition file is needed only if you are using **Discover IDoc From File**.

About this task

Specify the IDoc definition file that the external service wizard uses to discover the IDoc.

Note: The **Discover IDoc From File** choice applies to both the ALE interface and the ALE pass-through IDoc interface.

Procedure

1. In the Find Objects in the Enterprise System window, indicate which IDoc you want to work with.
 - a. Expand **ALE**.
 - b. Click **Discover IDoc From File** to enable the filter button. Find Objects in the Enterprise System window shows the window as it appears in the ALE interface. If you are using the ALE pass-through interface, you also see a **Generic IDoc** choice.

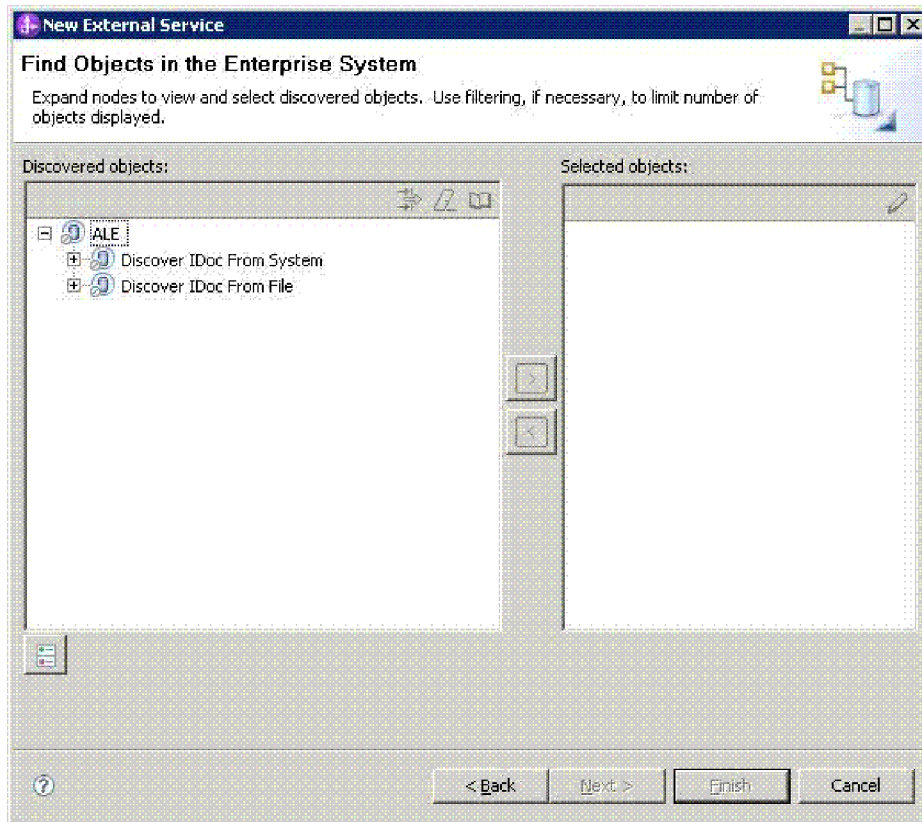


Figure 79. The Find Objects in the Enterprise System window

- c. Click the filter button.

Note: Instead of using the filter button, you can expand **Discover IDoc From File** and select the IDoc definition file. You then skip ahead to step 4 on page 196.

2. From the Filter Properties window, specify the location of the IDoc definition file.
 - a. Click **Browse** to navigate to the IDoc definition file, or type the path to the file.

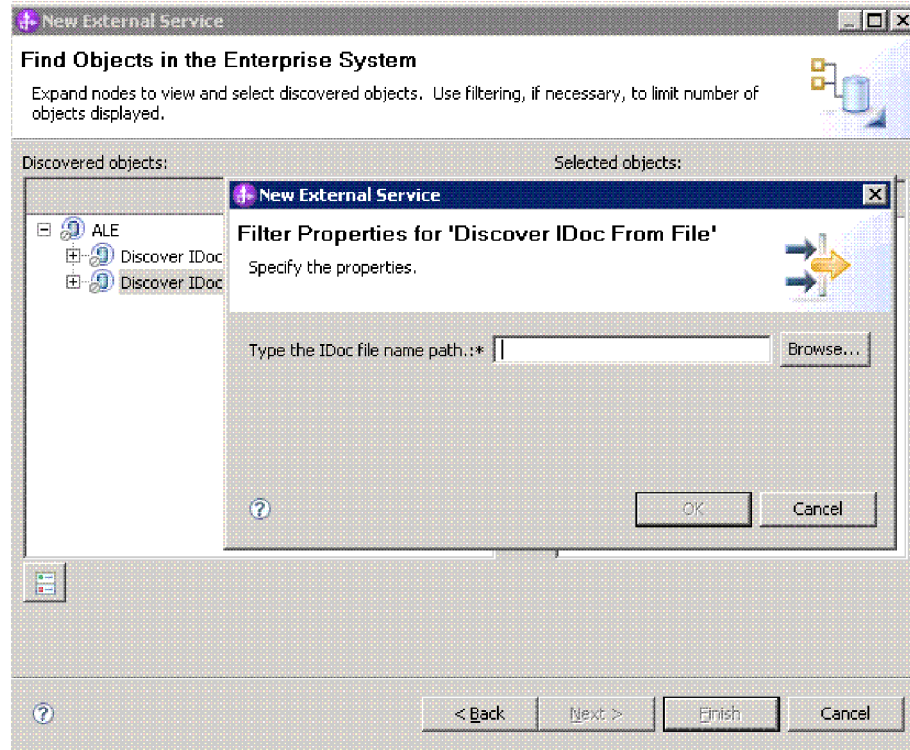


Figure 80. The Filter Properties for Discover IDoc From File window

- b. After you type or select the file, click **OK**.
3. Select the IDoc or IDocs.
 - a. Expand **Discover IDoc From File (filtered)**.
The IDoc definition file is displayed.
 - b. Click the IDoc definition file.

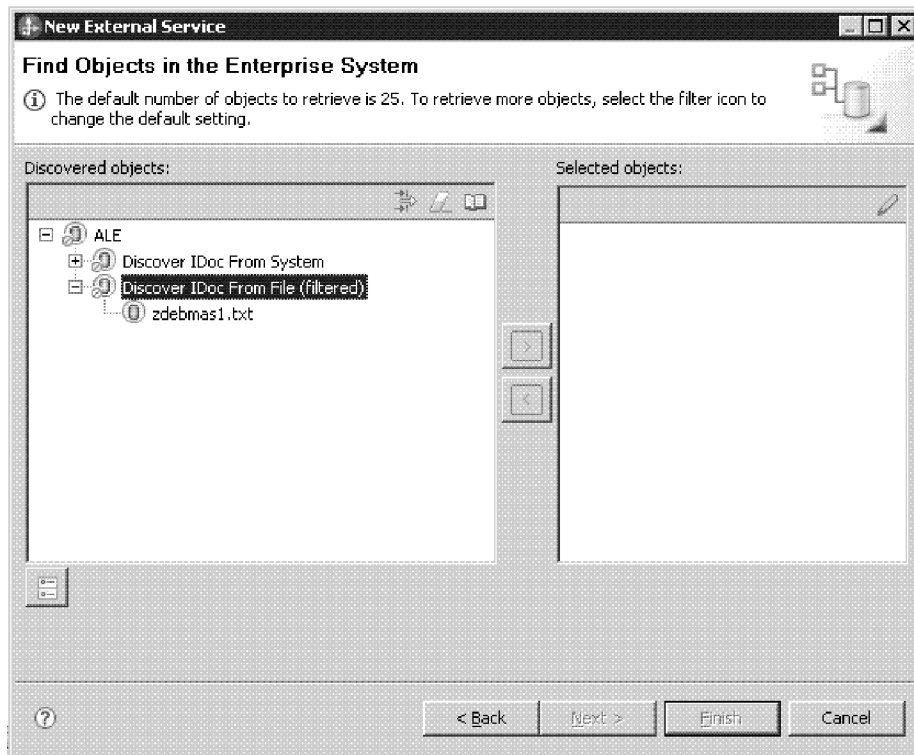


Figure 81. The Find Objects in the Enterprise System window

4. Click the arrow button to add it to the **Selected objects** list.
5. In the Specify the Configuration Properties for 'object' window, perform the following tasks:

Note: If you are using the ALE pass-through IDoc interface, only the **Send an IDoc Packet as one business object** configuration property is available.

- a. Optionally select the **Use SAP field names to generate attribute names**. By default, when the check box is not selected, field descriptions are used to generate properties. If you choose to use SAP field names to generate attribute names, two more check boxes are enabled:
- b. Select the **Use SAP-original casing for Control Record Business object attribute names** check box to generate attribute names in SAP original casing. If unchecked, attribute names are generated using field names from SAP in camel-casing.
- c. Select the **Use SAP-original casing for Data Record Business object attribute names** check box to generate attribute names in SAP original casing. If unchecked, attribute names are generated using field Name from SAP in camel-casing.

The following are the different possible combinations:

Table 21.

Scenario	Use SAP field names to generate attribute names (check-box)	Use SAP-original casing Control Record business object attribute names (check-box)	Use SAP-original casing Data Record business object attribute names (check-box)	Control Record	Data Record
1	Checked	Checked	Checked	Attribute names are generated using Field Name from SAP (Original SAP casing)	Attribute names are generated using Field Name from SAP (Original SAP casing)
2	Checked	Checked	Not checked	Attribute names are generated using Field Name from SAP (Original SAP casing)	Attribute names are generated using Field Name from SAP (Camel-casing)
3	Checked	Not checked	Checked	Attribute names are generated using Field Name from SAP (Camel-casing)	Attribute names are generated using Field Name from SAP (Original SAP casing)
4	Checked	Not checked	Not checked	Attribute names are generated using Field Name from SAP (Camel-casing)	Attribute names are generated using Field Name from SAP (Camel-casing)
5	Not checked	Not checked	Not checked	Attribute names are generated using Field Description from SAP (Camel-casing)	Attribute names are generated using Field Description from SAP (Camel-casing)

- d. If you are working with an IDoc packet and want to specify that the packet not be split, select the **Send an IDoc Packet as one business object** check box.
- e. 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 an IDoc packet as unparsed data** check box.

Note: Refer to Resolving data record (hexbinary) format issues for ALE inbound interfaces for more information on resolving data record format issues for ALE inbound interfaces.

- f. In the **IDoc release version** field, specify the SAP release number to identify the IDoc type you want the external service wizard to use for creating business objects.
 - g. Click **OK**.
6. Click **Next**.

Results

The external service wizard has returned an IDoc or a list of IDocs associated with the IDoc definition file. You see the Specify Composite Properties window (if you are using the ALE interface) or the Specify the Service Generation and Deployment Properties (if you are using the ALE pass-through IDoc interface).

What to do next

- If you are using the ALE interface, you can optionally specify a namespace and directory to which the generated business object will be stored and indicate whether you want a business graph generated, as described in “Configuring the selected objects” on page 148.
- If you are using the ALE pass-through IDoc interface, you generate a deployable module that includes the adapter and the business objects, as described in “Setting deployment properties and generating the service” on page 200.

Configuring the selected objects

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

Before you begin

Make sure you have selected and imported the ALE IDoc.

About this task

To configure the business object, use the following procedure.

Procedure

1. Select an IDoc to configure from the **IDocs selected** pane. You can select multiple IDocs to configure.

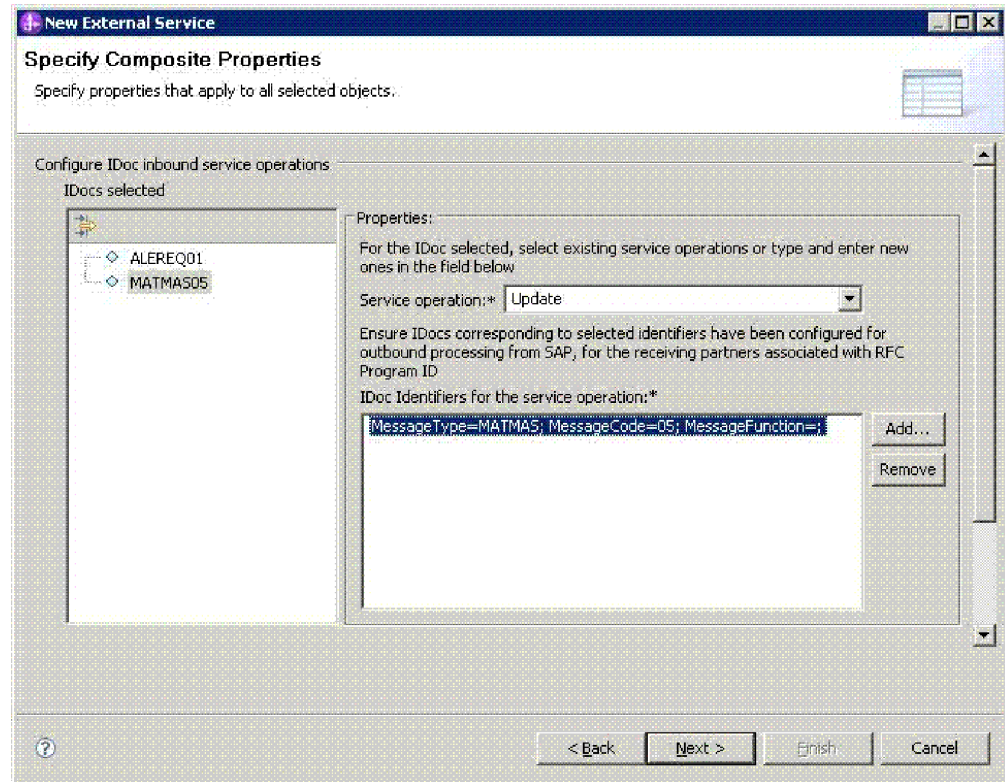


Figure 82. The Specify Composite Properties window to configure the inbound service operation.

2. Select an operation (for example, **Create**) from the **Service operation** drop-down list.

For each selected IDoc and receiving partner, you can configure the service operation. In addition to the default existing service operations which are Create, Update and Delete, you can specify a new operation by entering an operation name in the **Service operation** field.

3. Click **Add** to add the identifiers you want to associate with the operation. From the **IDoc Identifiers for the service operation:** list, select a set of identifiers to associate the Receiving partner, IDoc message type, message code, and message function values with the selected service operation. At run time, the adapter uses these values to identify the service operation at the endpoint for invocation. You can associate multiple identifiers for a single service operation.

Note: New operation will be added to the list only if you associate an identifier with the operation.

4. In the **Business object 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 external service wizard), change the namespace value.

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

5. To indicate where the business object information should be stored, type the path to the location in the **Folder** field. This is an optional step.

Note: The above two fields cannot be edited if you are modifying existing artifacts.

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

6. If you want the IDoc or IDocs to be enclosed within a business graph, leave **Generate a business graph for each business object** selected. Otherwise, remove the check.
7. Click **Next**.

Results

You have associated an operation with an identifier. The Specify the Service Generation and Deployment Properties window is displayed.

What to do next

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

Setting deployment properties and generating the service

To generate the module, which is the artifact that is deployed on IBM Business Process Manager 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 all the information needed to complete the Specify the Service Generation and Deployment Properties window and that you have performed any prerequisite tasks. For example, a program ID must be registered on the SAP server. If you want to assure once-only delivery of inbound events by persisting events to an event recovery table, make sure a data source has been created to hold the table.

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.

Procedure

1. 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**.
2. Indicate whether you will use an authentication alias or type in a user ID and password to establish a connection to the SAP server:
 - To specify an authentication alias, leave **Specify a Java Authentication and Authorization Services (JAAS) alias security credential** selected. Then, in the **J2C Authentication Data Entry** field, enter the name you specified in the Security section of the administrative console.

If you are not going to use an authentication alias, clear **Specify a Java Authentication and Authorization Services (JAAS) alias security credential**.

- To use a user ID and password, select **Using security properties from the activation specification**. Type in the user name and password to specify your security credentials.
 - To use other security mechanisms native to the enterprise system or if security is not required, select **Other**
3. Select **With module for use by single application** to embed the adapter files in a module that is deployed to the application server, or select **On server for use by multiple applications** to install the adapter files as a stand-alone adapter.

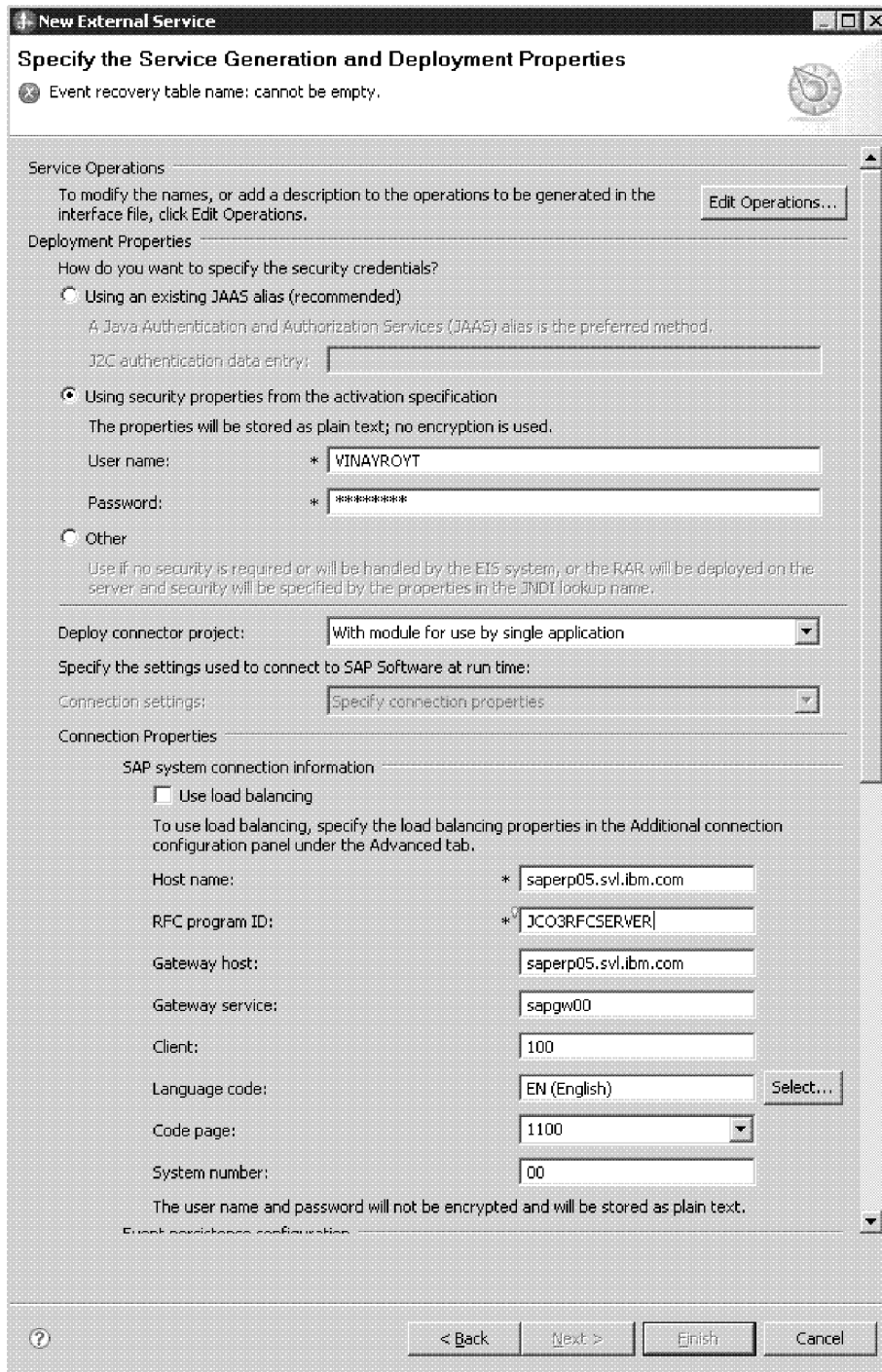


Figure 83. Specify the Service Generation and Deployment Properties window

- **With module for use by single application:** With the adapter files embedded in the module, you can deploy the module to any application server. Use an embedded adapter when you have a single module using the adapter or if multiple modules need to run different versions of the adapter. Using an embedded adapter enables you to upgrade the adapter in a single module without the risk of destabilizing other modules by changing their adapter version.

- **On server for use by multiple applications:** If you do not include the adapter files in a module, you must install them as a stand-alone adapter on each application server where you want to run the module. Use a stand-alone adapter when multiple modules can use the same version of the adapter and you want to administer the adapter in a central location. A stand-alone adapter can also reduce the resources required by running a single adapter instance for multiple modules.
4. If you selected **On server for use by multiple applications** in the previous step, the **Connection properties** list becomes active. Make one of the following selections:
 - Select **Specify connection properties** if you want to provide configuration information now. Then continue with step 5.
 - Select **Use predefined connection properties** if you want to use an activation specification configuration that already exists.

If you decide to use predefined connection properties, you must ensure that your resource adapter name matches the name of the installed adapter, because this is how the adapter instance is associated with these properties. If you want to change the resource adapter name in the import or export, use the assembly editor in IBM Integration Designer to change the value in the import or export.

When you select **Use predefined connection properties**, the **JNDI Lookup Name** field is displayed in place of the properties.

 - a. Type a value for **JNDI Lookup Name**.
 - b. Click **Next**.
 - c. Go to step 9 on page 205.
 5. In the Connection properties section, set or change any connection properties that apply to your configuration.

Notice that some of the values are already filled in. For example, the values that you used in the Specify the Discovery Properties window (such as the **Host name**) are filled in.

 - a. Select the **Use load balancing** check box to use load balancing to connect to the SAP system. Load balancing properties which are **Message server host**, **Logon group name** and **SAP system ID** have to be specified in the Additional connection configuration panel under the Advanced tab.
 - b. Change the **Host name** field if you are planning to send events from a different SAP server than the one you are using to create the adapter module.
 - c. In the **RFC Program ID** field, type the name of the program ID you registered with the SAP server.
 - d. The **Gateway host** is filled in, by default, with the value from the **Host name** field.
 - e. A default value of **sapgw00** is filled in for **Gateway service**. If you have more than one gateway server in your SAP configuration, change **sapgw00** to the correct value.
 - f. The remaining values in the SAP system connection information section are filled in with the values you entered in the Specify the Discovery Properties window. Change these values, if necessary.

See “Activation specification properties for ALE inbound processing” on page 383 for more information about these properties.

Properties marked with an asterisk (*) are required.

6. In the Event persistence configuration section, select the properties to retain events in memory. Selecting event persistence options ensures a once-only delivery of inbound events. If you do not select the options, performance increases, but there is a risk of losing events in transit if there is an unexpected shut down.
 - a. The **Ensure assured-once event delivery** field is selected by default to get a once-only delivery of the inbound events. This may reduce performance.
 - b. Select **Auto create event table** if you want the adapter to create an event table. This field is selected by default.
 - c. Enter a name in the **Event recovery table name** field
 - d. Enter the JNDI name in the **Event recovery data source (JNDI) name** field
 - e. Enter the user name in the **User name used to connect to event data source** field
 - f. Enter the password in the **Password used to connect to event data source** field
 - g. Enter the database schema name in the **Database schema name** field
7. To set additional properties, click **Advanced**.
 - a. Expand **Processing methods for illegal XML characters**
 - 1) From the **Illegal XML character processing** drop-down list, select one of the options that correspond to your requirement:
 - **Do not validate illegal XML characters**, if you want to work with the default adapter behavior.
 - **Throw exception if contents in the BO contains illegal XML characters**, if you want to proceed after you get an exception message, and log the illegal XML characters in the trace file, at runtime.
 - **Discard all illegal XML characters and related logs**, if you want the adapter to discard the illegal XML characters, and log the characters in the trace file, at runtime.

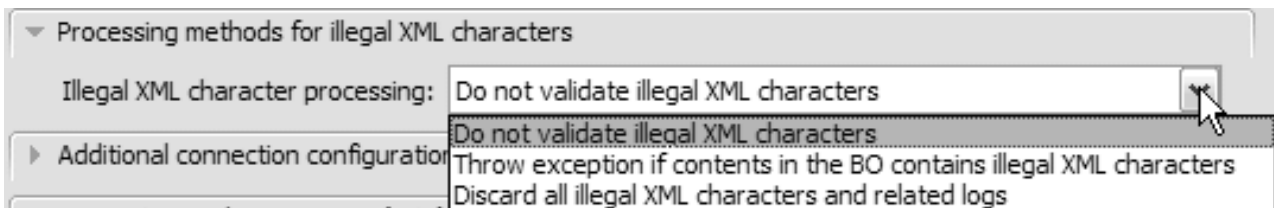


Figure 84. Processing methods for illegal XML characters

- b. Optionally expand **Additional connection configuration** and provide values (or change the default values) for the fields in this section of the window. For example, if your SAP configuration uses load balancing, provide values for the **Message server host** field, **Logon group name** and **SAP system ID**.
- c. You can expand **Event polling configuration** and specify values indicating how events should be polled on the SAP server. For example, you can enter a list of event types in the **Event types to process** field if you want to limit the events that the adapter should process. You can select **Retry EIS connection on startup** if you want the adapter to retry a failed connection when starting. For more information, see “Retry EIS connection on startup” on page 400.

- d. Optionally expand **ALE event status configuration** and select **Ignore IDoc packet errors** if you want to continue to processing an IDoc packet if any errors occur during IDoc processing. If you want to provide update status for ALE processing, select **ALE update status** and fill in the associated fields. Properties marked with an (*) are required. Select **Send ALEAUD per packet** if you want to send one ALEAUD per IDoc packet which will contain confirmations for all IDocs in the packet.
- e. If you are using Secure Network Connection, expand **Secure Network Connection (SNC) Configuration** and select **Enable secure network connection**. Then enter information in the associated fields (name, partner, security level, and library path). Optionally type the name of an X509 certificate.
- f. Optionally set a comma-separated list of Interceptor classes in the **List of Interceptors** field. These Interceptor classes act as User Exits and must implement the `com.ibm.j2ca.extension.monitoring.interceptors.Interceptor` Java interface. Refer to User Exits through Interceptors for more details on Interceptors.
- g. Optionally to set RFC and JCo tracing properties, perform the following steps:
 - 1) Expand **SAP RFC trace configuration** and select **RFC trace on**.
 - 2) Select the **Write JCo traces into adapter logs** checkbox to enable JCo API traces to be generated in the broker trace file. These traces will then be interleaved with the adapter traces. Each JCo API trace logged into the broker trace file will have the string [JCoAPI]. All JCo API traces are logged at the INFO message log level.
 - 3) Select a tracing level from the **RFC trace level** list.
 - 4) Click **Browse** and select a location to which the RFC trace files will be saved.
- h. Optionally expand **Logging and tracing** and specify the following information:
 - If you have multiple instances of the adapter, set **Adapter ID** to a value that is unique for this instance.
 - If you want to mask sensitive information in log and trace files (for example, if you want to avoid making customer information visible in these files), select **Disguise user data as "XXX" in log and trace files**.

See “Activation specification properties for ALE inbound processing” on page 383 for more information about these properties.
8. Click **Next**.
9. Create a module.
 - a. In the Specify the Location Properties window, click **New** in the **Module** field.
 - b. In the Integration Project window, click **Create a module project** or **Create a mediation module project** and click **Next**.
10. In the New Module window, perform the following tasks:
 - a. Type a name for the module.
As you type the name, it is added to the workplace specified in the **Location** field.
This is the default location. If you want to specify a different location, remove the check from **Use default location** and type a new location or click **Browse** and select the location.

- b. Specify whether you want to open the module in the assembly diagram (for module projects) or whether you want to create a mediation flow component (for mediation module projects). By default, these choices are selected.
 - c. Click **Finish**.
11. In the Specify the Location Properties window, perform the following tasks:
 - a. If you want to change the default namespace, clear the **Use default namespace** check box and type a new path in the **Namespace** field.
 - b. 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. This is an optional step.
 - c. Optionally change the name of the interface.
The default name is SAPI inboundInterface. You can change it to a more descriptive title if you prefer.
 - d. If you want to save the business objects so that they can be used by another application, click **Save business objects to a library** and then select a library from the list or click **New** to create a new library.
 - e. Optionally type a description of the module.
12. Click **Finish**.

Results

The new module is added to the Business Integration perspective.

What to do next

Export the module as an EAR file for deployment.

Configuring a module for ALE pass-through IDoc inbound processing

To configure a module to use the adapter for ALE inbound processing, you use the external service wizard IBM Integration Designer in to find an IDoc or set of IDocs, configure the business objects that are generated, and create a deployable module. If you are going to set up an event recovery table to persist inbound events (to ensure once-only delivery of events), you must also set up a data source.

Selecting business objects and services for ALE pass-through IDoc inbound processing

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

About this task

For the ALE pass-through IDoc interface, you can specify IDocs from a system or from a file, but the most likely reason for using the pass-through IDoc interface is to use a generic IDoc.

When you select a generic IDoc, you create one business-object definition that can apply to any IDoc at run time. This selection is helpful if you are processing many IDocs and do not want to create a separate business-object definition for each one.

Note: You see the **Generic IDoc** choice only if you selected **ALE pass-through IDoc** as the interface on the Discovery Connection window.

Note: The adapter now provides an out-of-the-box functionality to parse out the stream data provided in Passthrough IDoc Business Objects.

To use this new feature, you need to configure the DataBinding (after the EMD run) to use the `com.ibm.j2ca.sap.a1e.idoc.datahandler.SAPIDocDataHandlerDataHandler` Java class that is now bundled with the SAP adapter RAR file.

When invoking the Data Handler, you must provide the following details:

1. Name of the specific IDoc For example, SapA1ereq01. The corresponding schema must be available for the Artifact Loader at runtime.
2. Business Object namespace for that specific schema.
3. Data Encoding.

With this setting, your end point will get the specific parsed IDoc without any change to the existing modules.

To learn more about how to work with Java Beans and Data Handlers, refer to the IBM Business Process Manager or WebSphere Enterprise Service Bus documentation.

Procedure

1. In the Find Objects in the Enterprise System window, indicate that you want to select a generic IDoc.
 - a. Expand **ALE**.
 - b. Click **Generic IDoc** .
2. Click the arrow button to add the generic IDoc to the **Selected objects** list.
3. When the Configuration Properties window is displayed, you can set the following configurable options for the selected IDoc:

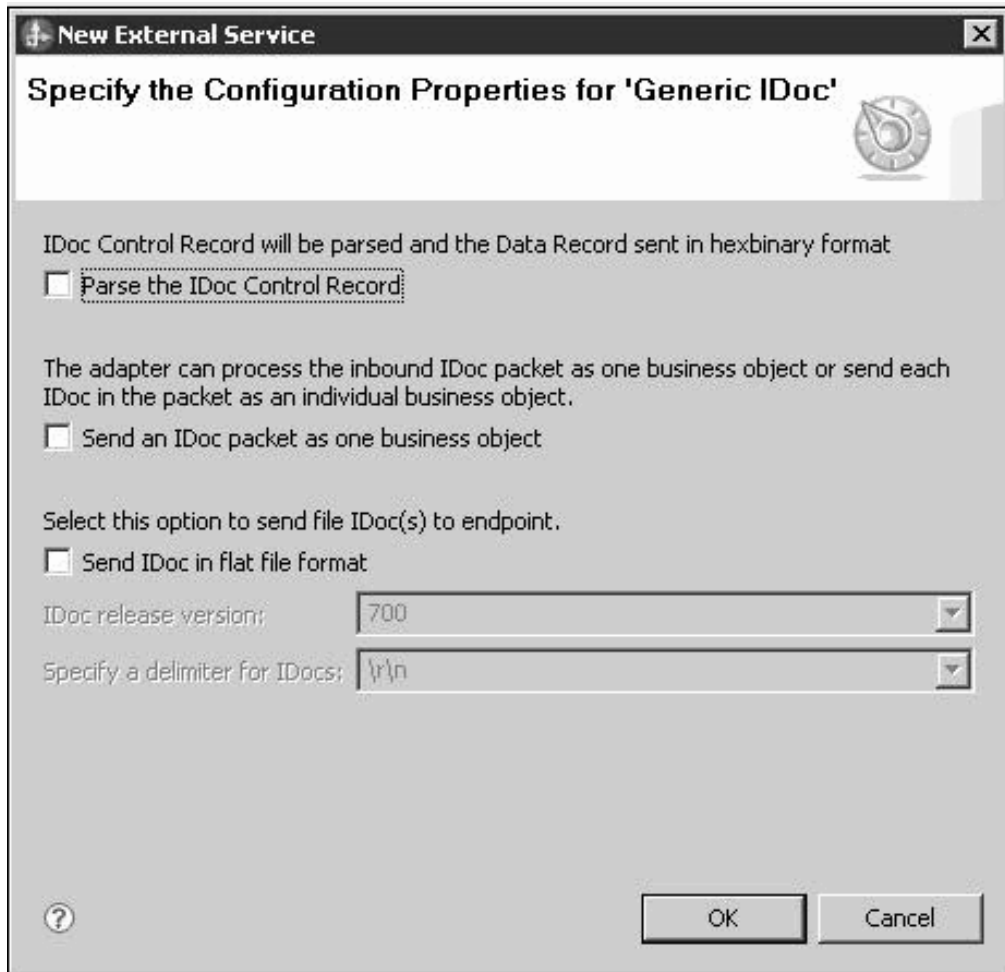


Figure 85. Configuration parameters for a generic IDoc

- a. Enable the checkbox **Parse the IDoc Control Record**, to generate a child business object to hold the parsed Control Record. When enabled, the property to send IDocs in the Flat File format are disabled. This feature enables you to route IDocs based on individual Control Record settings.
- b. Indicate whether you want to have multiple IDocs sent as one packet instead of sending them as individual business objects:
 - If you want to send multiple IDocs as a packet, select the **Send an IDoc packet as one business object** option.
- c. Indicate whether you want to send the IDoc in the Flat File format to the configured end point(s):
 - If you do not want to send the IDoc in the Flat File format, leave the **Send IDoc in flat-file format** option unchecked
 - If you want to send IDoc in the Flat File format, select the **Send IDoc in flat-file format** option:
 - 1) When sending the IDoc in the Flat File format, the Control Record length will remain 524. The Data Record length can vary depending on the segment length which is based on the IDoc version selected. Choose the correct value in the **IDoc release version** field. If the IDoc contains unreleased segments, leave the version field blank.
 - 2) When sending the IDoc in the Flat File format, set a delimiter, which will be added after the Control Record as well as after each Data

Record, in the **Specify a delimiter for IDocs** field. You can choose standard delimiters provided in the drop down or specify a custom one.

4. Click **OK**. Then click **Next**.

Results

You have selected a generic IDoc.

What to do next

Set deployment properties and generate a module.

Setting deployment properties and generating the service

To generate the module, which is the artifact that is deployed on IBM Business Process Manager 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 all the information needed to complete the Specify the Service Generation and Deployment Properties window and that you have performed any prerequisite tasks. For example, a program ID must be registered on the SAP server. If you want to assure once-only delivery of inbound events by persisting events to an event recovery table, make sure a data source has been created to hold the table.

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.

Procedure

1. 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**.
2. Indicate whether you will use an authentication alias or type in a user ID and password to establish a connection to the SAP server:
 - To specify an authentication alias, leave **Specify a Java Authentication and Authorization Services (JAAS) alias security credential** selected. Then, in the **J2C Authentication Data Entry** field, enter the name you specified in the Security section of the administrative console.
If you are not going to use an authentication alias, clear **Specify a Java Authentication and Authorization Services (JAAS) alias security credential**.
 - To use a user ID and password, select **Using security properties from the activation specification**. Type in the user name and password to specify your security credentials.
 - To use other security mechanisms native to the enterprise system or if security is not required, select **Other**

3. Select **With module for use by single application** to embed the adapter files in a module that is deployed to the application server, or select **On server for use by multiple applications** to install the adapter files as a stand-alone adapter.

New External Service

Specify the Service Generation and Deployment Properties

Event recovery table name: cannot be empty.

Service Operations

To modify the names, or add a description to the operations to be generated in the interface file, click Edit Operations. Edit Operations...

Deployment Properties

How do you want to specify the security credentials?

Using an existing JAAAS alias (recommended)
A Java Authentication and Authorization Services (JAAAS) alias is the preferred method.
J2C authentication data entry:

Using security properties from the activation specification
The properties will be stored as plain text; no encryption is used.

User name: *

Password: *

Other
Use if no security is required or will be handled by the EIS system, or the RAR will be deployed on the server, and security will be specified by the properties in the JNDI lookup name.

Deploy connector project:

Specify the settings used to connect to SAP Software at run time:

Connection settings:

Connection Properties

SAP system connection information

Use load balancing
To use load balancing, specify the load balancing properties in the Additional connection configuration panel under the Advanced tab.

Host name: *

RFC program ID: *

Gateway host:

Gateway service:

Client:

Language code: Select...

Code page:

System number:

The user name and password will not be encrypted and will be stored as plain text.

Event persistence configuration

? < Back Next > Finish Cancel

Figure 86. Specify the Service Generation and Deployment Properties window

- **With module for use by single application:** With the adapter files embedded in the module, you can deploy the module to any application server. Use an embedded adapter when you have a single module using the adapter or if multiple modules need to run different versions of the adapter.

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

- **On server for use by multiple applications:** If you do not include the adapter files in a module, you must install them as a stand-alone adapter on each application server where you want to run the module. Use a stand-alone adapter when multiple modules can use the same version of the adapter and you want to administer the adapter in a central location. A stand-alone adapter can also reduce the resources required by running a single adapter instance for multiple modules.
4. If you selected **On server for use by multiple applications** in the previous step, the **Connection properties** list becomes active. Make one of the following selections:
- Select **Specify connection properties** if you want to provide configuration information now. Then continue with step 5.
 - Select **Use predefined connection properties** if you want to use an activation specification configuration that already exists.

If you decide to use predefined connection properties, you must ensure that your resource adapter name matches the name of the installed adapter, because this is how the adapter instance is associated with these properties. If you want to change the resource adapter name in the import or export, use the assembly editor in IBM Integration Designer to change the value in the import or export.

When you select **Use predefined connection properties**, the **JNDI Lookup Name** field is displayed in place of the properties.

- a. Type a value for **JNDI Lookup Name**.
 - b. Click **Next**.
 - c. Go to step 9 on page 213.
5. In the Connection properties section, set or change any connection properties that apply to your configuration.

Notice that some of the values are already filled in. For example, the values that you used in the Specify the Discovery Properties window (such as the **Host name**) are filled in.

- a. Select the **Use load balancing** check box to use load balancing to connect to the SAP system. Load balancing properties which are **Message server host**, **Logon group name** and **SAP system ID** have to be specified in the Additional connection configuration panel under the Advanced tab.
- b. Change the **Host name** field if you are planning to send events from a different SAP server than the one you are using to create the adapter module.
- c. In the **RFC Program ID** field, type the name of the program ID you registered with the SAP server.
- d. The **Gateway host** is filled in, by default, with the value from the **Host name** field.
- e. A default value of **sapgw00** is filled in for **Gateway service**. If you have more than one gateway server in your SAP configuration, change **sapgw00** to the correct value.
- f. The remaining values in the SAP system connection information section are filled in with the values you entered in the Specify the Discovery Properties window. Change these values, if necessary.

See “Activation specification properties for ALE inbound processing” on page 383 for more information about these properties.

Properties marked with an asterisk (*) are required.

6. In the Event persistence configuration section, select the properties to retain events in memory. Selecting event persistence options ensures a once-only delivery of inbound events. If you do not select the options, performance increases, but there is a risk of losing events in transit if there is an unexpected shut down.
 - a. The **Ensure assured-once event delivery** field is selected by default to get a once-only delivery of the inbound events. This may reduce performance.
 - b. Select **Auto create event table** if you want the adapter to create an event table. This field is selected by default.
 - c. Enter a name in the **Event recovery table name** field
 - d. Enter the JNDI name in the **Event recovery data source (JNDI) name** field
 - e. Enter the user name in the **User name used to connect to event data source** field
 - f. Enter the password in the **Password used to connect to event data source** field
 - g. Enter the database schema name in the **Database schema name** field
7. To set additional properties, click **Advanced**.
 - a. Expand **Processing methods for illegal XML characters**
 - 1) From the **Illegal XML character processing** drop-down list, select one of the options that correspond to your requirement:
 - **Do not validate illegal XML characters**, if you want to work with the default adapter behavior.
 - **Throw exception if contents in the BO contains illegal XML characters**, if you want to proceed after you get an exception message, and log the illegal XML characters in the trace file, at runtime.
 - **Discard all illegal XML characters and related logs**, if you want the adapter to discard the illegal XML characters, and log the characters in the trace file, at runtime.

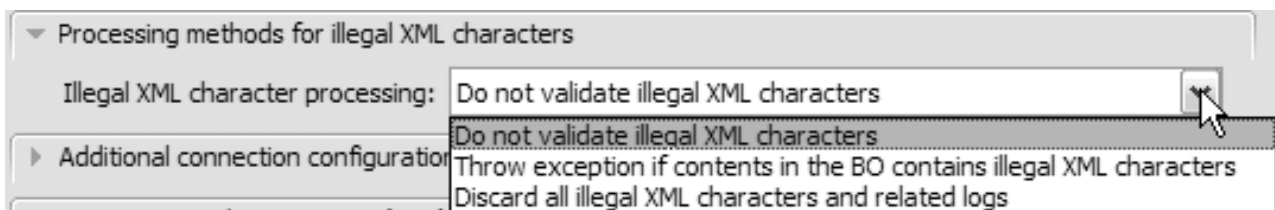


Figure 87. Processing methods for illegal XML characters

- b. Optionally expand **Additional connection configuration** and provide values (or change the default values) for the fields in this section of the window. For example, if your SAP configuration uses load balancing, provide values for the **Message server host** field, **Logon group name** and **SAP system ID**.
- c. You can expand **Event polling configuration** and specify values indicating how events should be polled on the SAP server. For example, you can enter a list of event types in the **Event types to process** field if you want to limit the events that the adapter should process. You can select **Retry**

EIS connection on startup if you want the adapter to retry a failed connection when starting. For more information, see “Retry EIS connection on startup” on page 400.

- d. Optionally expand **ALE event status configuration** and select **Ignore IDoc packet errors** if you want to continue to processing an IDoc packet if any errors occur during IDoc processing. If you want to provide update status for ALE processing, select **ALE update status** and fill in the associated fields. Properties marked with an (*) are required. Select **Send ALEAUD per packet** if you want to send one ALEAUD per IDoc packet which will contain confirmations for all IDocs in the packet.
 - e. If you are using Secure Network Connection, expand **Secure Network Connection (SNC) Configuration** and select **Enable secure network connection**. Then enter information in the associated fields (name, partner, security level, and library path). Optionally type the name of an X509 certificate.
 - f. Optionally set a comma-separated list of Interceptor classes in the **List of Interceptors** field. These Interceptor classes act as User Exits and must implement the `com.ibm.j2ca.extension.monitoring.interceptors.Interceptor` Java interface. Refer to User Exits through Interceptors for more details on Interceptors.
 - g. Optionally to set RFC and JCo tracing properties, perform the following steps:
 - 1) Expand **SAP RFC trace configuration** and select **RFC trace on**.
 - 2) Select the **Write JCo traces into adapter logs** checkbox to enable JCo API traces to be generated in the broker trace file. These traces will then be interleaved with the adapter traces. Each JCo API trace logged into the broker trace file will have the string [JCoAPI]. All JCo API traces are logged at the INFO message log level.
 - 3) Select a tracing level from the **RFC trace level** list.
 - 4) Click **Browse** and select a location to which the RFC trace files will be saved.
 - h. Optionally expand **Logging and tracing** and specify the following information:
 - If you have multiple instances of the adapter, set **Adapter ID** to a value that is unique for this instance.
 - If you want to mask sensitive information in log and trace files (for example, if you want to avoid making customer information visible in these files), select **Disguise user data as "XXX" in log and trace files**.
- See “Activation specification properties for ALE inbound processing” on page 383 for more information about these properties.
8. Click **Next**.
 9. Create a module.
 - a. In the Specify the Location Properties window, click **New** in the **Module** field.
 - b. In the Integration Project window, click **Create a module project** or **Create a mediation module project** and click **Next**.
 10. In the New Module window, perform the following tasks:
 - a. Type a name for the module.

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

- This is the default location. If you want to specify a different location, remove the check from **Use default location** and type a new location or click **Browse** and select the location.
- b. Specify whether you want to open the module in the assembly diagram (for module projects) or whether you want to create a mediation flow component (for mediation module projects). By default, these choices are selected.
 - c. Click **Finish**.
11. In the Specify the Location Properties window, perform the following tasks:
- a. If you want to change the default namespace, clear the **Use default namespace** check box and type a new path in the **Namespace** field.
 - b. 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. This is an optional step.
 - c. Optionally change the name of the interface.
The default name is `SAPInboundInterface`. You can change it to a more descriptive title if you prefer.
 - d. If you want to save the business objects so that they can be used by another application, click **Save business objects to a library** and then select a library from the list or click **New** to create a new library.
 - e. Optionally type a description of the module.
12. Click **Finish**.

Results

The new module is added to the Business Integration perspective.

What to do next

Export the module as an EAR file for deployment.

Configuring a module for Advanced event processing - inbound

To configure a module to use the adapter for Advanced event processing, you use the external service wizard in IBM Integration Designer to find an IDoc or set of IDocs, configure the business objects that are generated, and create a deployable module. To use the Advanced event processing interface, you must first add the adapter-supplied transport files to the SAP server.

Selecting business objects and services for Advanced event (inbound) processing

To specify which function you want to process, you provide information in the external service wizard.

Before you begin

Make sure you have set the connection properties for the external service wizard.

About this task

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

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

Procedure

1. In the Find Objects in the Enterprise System window, indicate which IDoc you want to work with.
 - a. Expand **AEP**.
 - b. Click **Discover IDoc From System** to enable the filter



button.

- c. Click the filter button.

Note: Instead of using the filter button, you can expand **Discover IDoc From System** and select the IDoc from the list. Then skip ahead to step 4.

2. From the Filter Properties window, specify information about the IDoc or IDocs:
 - a. Select **Discover objects by name** or **Discover objects by description** from the **Object attribute to use for discovery** list.
 - b. Type a search string representing the IDoc you want to call.
 - c. Select **Basic IDocs** or **Extension IDocs** from the **IDoc type to use for discovery** field.
 - d. Indicate the number of functions you want returned by changing the value in the **Maximum number of objects to retrieve** field or by accepting the default value.
 - e. Click **OK**.
3. Select the IDoc or IDocs.
 - a. Expand **Discover IDoc From System (filtered)**.
 - b. Click the IDoc you want to use. If you are working with multiple IDocs, click the names of all the IDocs.
4. Click the arrow button to add the IDoc or IDocs to the **Selected objects** list.
5. In the Configuration Parameters window, perform the following tasks to add the IDoc to the list of business objects to be imported.
 - a. Optionally, select the **Use SAP field names to generate attribute names** check box. By default (when the check box is not selected), the field descriptions are used to generate properties.
 - b. In the **IDoc release version** field, specify the SAP release number to identify the IDoc type you want the external service wizard to use for creating business objects.
 - c. Expand the IDoc name and select one or more nodes to be used as the primary key, or leave the default values selected.
 - d. Click **OK**.
6. Click **Next**.

Results

The external service 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

From the Specify Composite Properties window, associate an operation with the IDoc and specify the ABAP function module for the selected operation.

Configuring the selected objects

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

Before you begin

Make sure you have selected and imported the IDoc.

About this task

To configure the business object, use the following procedure.

Procedure

1. In the Configure Composite Properties window, click an IDoc from the **IDoc to configure** list.
If you are configuring only one IDoc, this step is not necessary.
2. Click **Add** in the Service operations for selected IDoc section of the window.
3. Select an operation (for example, **Create**), and click **OK**.
4. In the **ABAP function module name for selected operation** field, type the name of the ABAP function module to associate with this operation.
5. If you are working with multiple IDocs, repeat the previous four steps for each IDoc.
6. In the **Business object 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 external service wizard), change the namespace value.
For example, you could change the namespace to <http://www.ibm.com/xmlns/prod/websphere/j2ca/sap1>.
7. To indicate where the business object information should be stored, type the path to the location in the **Folder** field. This is an optional step.

Note: The above two fields cannot be edited if you are modifying existing artifacts.

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

8. If you want the IDoc or IDocs to be enclosed within a business graph, leave **Generate a business graph for each business object** selected. Otherwise, remove the check.
9. Click **Finish**.

Results

You have associated an operation with each IDoc and associated an ABAP function module with the object. The Service Generation and Deployment Configuration window is displayed.

What to do next

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

Setting deployment properties and generating the service

To generate the module, which is the artifact that is deployed on IBM Business Process Manager 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 Specify the Service Generation and Deployment Properties 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.

Procedure

1. 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**.
2. Indicate whether you will use an authentication alias (instead of typing a user ID and password) to establish a connection to the SAP server:
 - To specify an authentication alias, leave **Specify a Java Authentication and Authorization Services (JAAS) alias security credential** selected. Then, in the **J2C Authentication Data Entry** field, enter the name you specified in the Security section of the administrative console.
If you are not going to use an authentication alias, clear **Specify a Java Authentication and Authorization Services (JAAS) alias security credential**.
 - To use a user ID and password, select **Using security properties from the activation specification**. Type in the user name and password to specify your security credentials.
 - To use other security mechanisms native to the enterprise system or if security is not required, select **Other**
3. Select **With module for use by single application** to embed the adapter files in a module that is deployed to the application server, or select **On server for use by multiple applications** to install the adapter files as a stand-alone adapter.

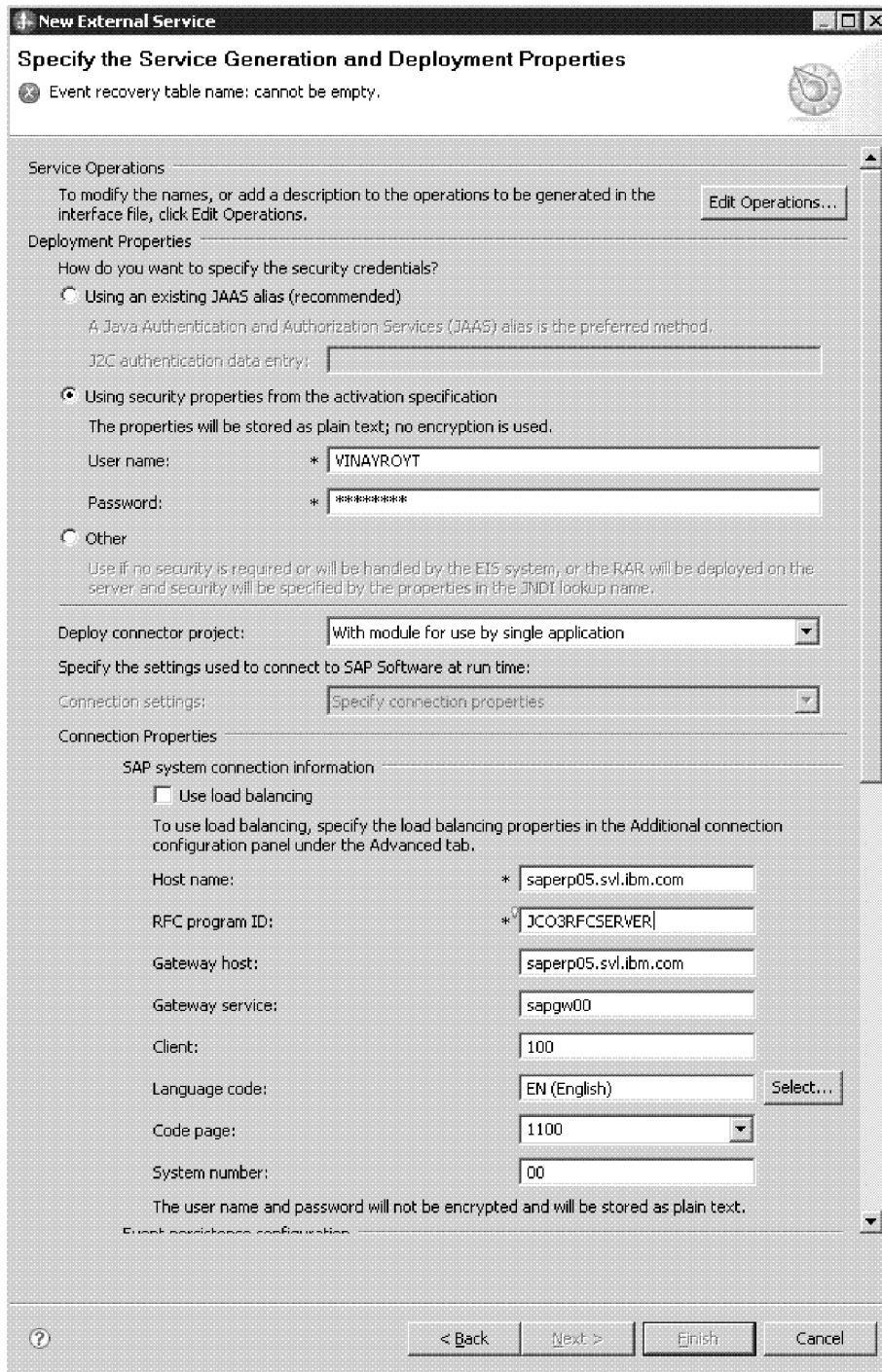


Figure 88. Specify the Service Generation and Deployment Properties window

- With module for use by single application:** With the adapter files embedded in the module, you can deploy the module to any application server. Use an embedded adapter when you have a single module using the adapter or if multiple modules need to run different versions of the adapter. Using an embedded adapter enables you to upgrade the adapter in a single module without the risk of destabilizing other modules by changing their adapter version.

- **On server for use by multiple applications:** If you do not include the adapter files in a module, you must install them as a stand-alone adapter on each application server where you want to run the module. Use a stand-alone adapter when multiple modules can use the same version of the adapter and you want to administer the adapter in a central location. A stand-alone adapter can also reduce the resources required by running a single adapter instance for multiple modules.
4. If you selected **On server for use by multiple applications** in the previous step, the **Connection properties** list becomes active. Make one of the following selections:
 - Select **Specify connection properties** if you want to provide configuration information now. Then continue with step 5.
 - Select **Use predefined connection properties** if you want to use an activation specification configuration that already exists.

If you decide to use predefined connection properties, you must ensure that your resource adapter name matches the name of the installed adapter, because this is how the adapter instance is associated with these properties. If you want to change the resource adapter name in the import or export, use the assembly editor in IBM Integration Designer to change the value in the import or export.

When you select **Use predefined connection properties**, the **JNDI Lookup Name** field is displayed in place of the properties.

 - a. Type a value for **JNDI Lookup Name**.
 - b. Click **Next**.
 - c. Go to step 7 on page 221.
 5. In the Connection properties section, set or change any connection properties that apply to your configuration.

Notice that some of the values are already filled in. For example, the values that you used in the Specify the Discovery Properties window (such as the **Host name**) are filled in.

Select the **Use load balancing** check box to use load balancing to connect to the SAP system. Load balancing properties which are **Message server host**, **Logon group name** and **SAP system ID** have to be specified in the Additional connection configuration panel under the Advanced tab.
 6. To set additional properties, click **Advanced**.
 - a. Expand **Processing methods for illegal XML characters**
 - 1) From the **Illegal XML character processing** drop-down list, select one of the options that correspond to your requirement:
 - **Do not validate illegal XML characters**, if you want to work with the default adapter behavior.
 - **Throw exception if contents in the BO contains illegal XML characters**, if you want to proceed after you get an exception message, and log the illegal XML characters in the trace file, at runtime.
 - **Discard all illegal XML characters and related logs**, if you want the adapter to discard the illegal XML characters, and log the characters in the trace file, at runtime.

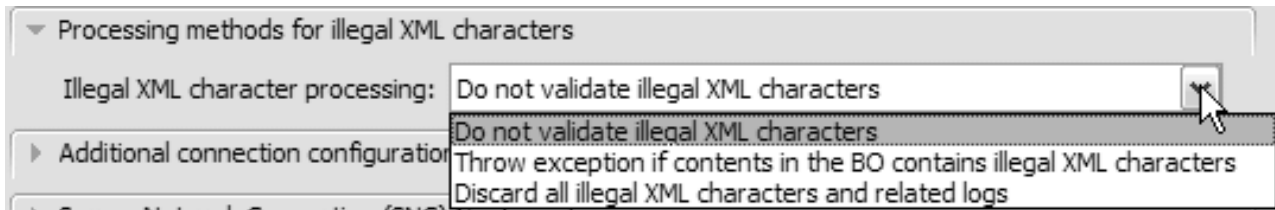


Figure 89. Processing methods for illegal XML characters

- b. Optionally expand **Additional connection configuration** and provide values (or change the default values) for the fields in this section of the window. For example, if your SAP configuration uses load balancing, provide values for the **Message server host** field, **Logon group name** and **SAP system ID**.
- c. You can expand **Event polling configuration** and specify values indicating how events should be polled on the SAP server. For example, you can enter a list of event types in the **Event types to process** field if you want to limit the events that the adapter should process. You can select **Retry EIS connection on startup** if you want the adapter to retry a failed connection when starting. For more information, see “Retry EIS connection on startup” on page 417.
- d. You can expand **Event delivery configuration** if you want to change the default values for the way events are delivered. Then enter a value (or change the default value) in one or more fields. For example, you can change the number of times the SAP server attempts to deliver a failed event.
See “Activation specification properties for Advanced event processing” on page 405 for more information about these properties.
- e. If you are using Secure Network Connection, expand **Secure Network Connection (SNC) Configuration** and select **Enable Secure Network Connection**. Then enter information in the associated fields (name, partner, security level, and library path). Optionally type the name of an X509 certificate.
- f. Optionally set a comma-separated list of Interceptor classes in the **List of Interceptors** field. These Interceptor classes act as User Exits and must implement the `com.ibm.j2ca.extension.monitoring.interceptors.Interceptor` Java interface. Refer to User Exits through Interceptors for more details on Interceptors.
- g. Optionally to set RFC and JCo tracing properties, perform the following steps:
 - 1) Expand **SAP RFC trace configuration** and select **RFC trace on**.
 - 2) Select the **Write JCo traces into adapter logs** checkbox to enable JCo API traces to be generated in the broker trace file. These traces will then be interleaved with the adapter traces. Each JCo API trace logged into the broker trace file will have the string [JCoAPI]. All JCo API traces are logged at the INFO message log level.
 - 3) Select a tracing level from the **RFC trace level** list.
 - 4) Click **Browse** and select a location to which the RFC trace files will be saved.
- h. Optionally expand **Logging and tracing** and specify the following information:

- If you have multiple instances of the adapter, set **Adapter ID** to a value that is unique for this instance.
- If you want to mask sensitive information in log and trace files (for example, if you want to avoid making customer information visible in these files), select **Disguise user data as "XXX" in log and trace files**.

See “Activation specification properties for Advanced event processing” on page 405 for more information about these properties.

7. Create a module.
 - a. In the Specify the Location Properties window, click **New** in the **Module** field.
 - b. In the Integration Project window, click **Create a module project** or **Create a mediation module project** and click **Next**.
8. In the New Module window, perform the following tasks:
 - a. Type a name for the module.
As you type the name, it is added to the workplace specified in the **Location** field.
This is the default location. If you want to specify a different location, remove the check from **Use default location** and type a new location or click **Browse** and select the location.
 - b. Specify whether you want to open the module in the assembly diagram (for module projects) or whether you want to create a mediation flow component (for mediation module projects). By default, these choices are selected.
 - c. Click **Finish**.
9. In the Specify the Location Properties window, perform the following tasks:
 - a. If you want to change the default namespace, clear the **Use default namespace** check box and type a new path in the **Namespace** field.
 - b. 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. This is an optional step.
 - c. Optionally change the name of the interface.
The default name is `SAPInboundInterface`. You can change it to a more descriptive title if you prefer.
 - d. If you want to save the business objects so that they can be used by another application, click **Save business objects to a library** and then select a library from the list or click **New** to create a new library.
 - e. Optionally type a description of the module.
10. Click **Finish**.

Results

The new module is added to the Business Integration perspective.

What to do next

Export the module as an EAR file for deployment.

Chapter 6. Changing interaction specification properties

To change interaction specification properties for your adapter module after generating the service, use the assembly editor in IBM Integration Designer.

Before you begin

You must have used the external service wizard to generate a service for the adapter.

About this task

You might want to change interaction specification properties after you have generated a service for the adapter. Interaction specification properties, which are optional, are set at the method level, for a specific operation on a specific business object. The values you specify appear as defaults in all parent business objects generated by the external service wizard. You can change these properties before you export the EAR file. You cannot change these properties after you deploy the application.

To change the interaction specification properties, use the following procedure:

Procedure

1. From the Business Integration perspective of IBM Integration Designer, expand the module name.
2. Expand **Assembly Diagram** and double-click the interface.
3. Click the interface in the assembly editor. The module properties are displayed.
4. Click the **Properties** tab. You can also right-click the interface in the assembly diagram and click **Show in Properties**.
5. Under **Binding**, click **Method bindings**. The methods for the interface are displayed, one for each combination of business object and operation.
6. Select the method whose interaction specification property you want to change.
7. Change the property in the **Generic** tab. Repeat this step for each method whose interaction specification property you want to change.

Results

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

What to do next

Deploy the module.

Chapter 7. Modifying artifacts

Often business requirements mandate changes to the backend enterprise information system (EIS) data structures. These changes call for regeneration and reconfiguration of artifacts (import and export) that were previously generated using the external service wizard.

A few business scenarios where the output from one enterprise service discovery flow can be reused in successive flows, are outlined as follows:

- When you want to add a new object to the object set.
- When you want to modify configurations on selected objects like changes to operations, operation names, and service level settings like security, transactions, and reliability.
- When you want to remove a discovered object from the object set.
- When you want to rediscover existing objects in the service to synchronize the service if the objects in the back-end system have been updated.

To modify existing artifacts, you can invoke the wizard in one of the following ways. The external service wizard is initialized with previously configured settings.

- In the assembly editor, select the component you want to modify, right-click, and select **Edit Binding**.
- In the Business Integration view, select the component you want to modify, right-click, and select **Edit Binding**.
- Select the component in the assembly editor and select the Properties view. In the Binding tab, click the **Edit** link.

Note: The Edit Binding option is available for artifacts generated using IBM Integration Designer 7.0 only. If you are importing a project interchange from an earlier version of IBM Integration Designer, the Edit Binding option is not available. If you have made any manual changes to the configuration, running the wizard again will overwrite these changes.

Modifying service import for BAPI outbound processing

Modify an import component by rediscovering and reconfiguring the objects using the Edit Binding option in IBM Integration Designer.

About this task

You can invoke the external service wizard to modify the information of a service import interface. The wizard automatically populates the existing information for the selected import interface. You can modify the objects and services, and then regenerate the import component with the modified data.

Procedure

1. Invoke the external service wizard for the selected service interface import component using one of the following methods.
 - In the assembly editor, select the component you want to modify, right-click and select **Edit Binding**.
 - In the Business Integration view, select the component you want to modify, right-click and select **Edit Binding**.

- Select the interface in the assembly editor and select the Properties view. In the Binding tab, click the **Edit** link.

The Find Objects in the Enterprise System window of the external service wizard is displayed. The external service wizard automatically populates the existing configuration details for the selected import interface.

2. In the Find Objects in the Enterprise System window, you can search for the SAP objects, select the objects you want to use in your module, configure each business object, modify an existing business object or remove an existing object. For more information about discovering objects for the BAPI interface, see "Selecting business objects and services for BAPI outbound processing" on page 113.
3. Click **Next**. If you click **Cancel**, the changes you made in the previous step does not take effect.
4. In the Specify Composite Properties window, specify properties that apply to all business objects. For more information, see "Configuring simple BAPI objects" on page 115.
5. Click **Next**.
6. In the Service Generation window, modify the service operations if required.
7. Click **Finish**. The artifacts are updated.

Results

The artifacts are updated.

What to do next

You can test and deploy your module.

Modifying service export for BAPI inbound processing

Modify an export component by rediscovering and reconfiguring the objects using the Edit Binding option in IBM Integration Designer.

About this task

You can invoke the external service wizard to modify the information of a service export interface. The wizard automatically populates the existing information for the selected export interface. You can modify the objects and services, and then regenerate the export component with the modified data.

Procedure

1. Invoke the external service wizard for the selected service interface export component using one of the following methods.
 - In the assembly editor, select the component you want to modify, right-click and select **Edit Binding**.
 - In the Business Integration view, select the component you want to modify, right-click and select **Edit Binding**.
 - Select the interface in the assembly editor and select the Properties view. In the Binding tab, click the **Edit** link.

The Find Objects in the Enterprise System window of the external service wizard is displayed. The external service wizard automatically populates the existing configuration details for the selected export interface.

2. In the Find Objects in the Enterprise System window, you can search for the SAP objects, select the objects you want to use in your module, configure each business object, modify an existing business object or remove an existing object. For more information about discovering objects for the BAPI interface, see “Selecting business objects and services for BAPI inbound processing” on page 179.
3. Click **Next**. If you click **Cancel**, the changes you made in the previous step does not take effect.
4. In the Specify Composite Properties window, specify properties that apply to all business objects. For more information, see “Configuring the selected objects” on page 181.
5. Click **Next**.
6. In the Service Generation window, modify the service operations if required.
7. Click **Finish**. The artifacts are updated.

Results

The artifacts are updated.

What to do next

You can test and deploy your module.

Modifying service import for ALE outbound processing

Modify an import component by rediscovering and reconfiguring the objects using the Edit Binding option in IBM Integration Designer.

About this task

You can invoke the external service wizard to modify the information of a service import interface. The wizard automatically populates the existing information for the selected import interface. You can modify the objects and services, and then regenerate the import component with the modified data.

Procedure

1. Invoke the external service wizard for the selected service interface import component using one of the following methods.
 - In the assembly editor, select the component you want to modify, right-click and select **Edit Binding**.
 - In the Business Integration view, select the component you want to modify, right-click and select **Edit Binding**.
 - Select the interface in the assembly editor and select the Properties view. In the Binding tab, click the **Edit** link.

The Find Objects in the Enterprise System window of the external service wizard is displayed. The external service wizard automatically populates the existing configuration details for the selected import interface.

2. In the Find Objects in the Enterprise System window, you can search for the SAP objects, select the objects you want to use in your module, configure each business object, modify an existing business object or remove an existing object. For more information about discovering objects for the ALE interface, see “Selecting business objects and services for ALE outbound processing” on page 140.

3. Click **Next**. If you click **Cancel**, the changes you made in the previous step does not take effect.
4. In the Specify Composite Properties window, specify properties that apply to all business objects. For more information, see "Configuring the selected objects" on page 148.
5. Click **Next**.
6. In the Service Generation window, modify the service operations if required.
7. Click **Finish**. The artifacts are updated.

Results

The artifacts are updated.

What to do next

You can test and deploy your module.

Modifying service export for ALE inbound processing

Modify an export component by rediscovering and reconfiguring the objects using the Edit Binding option in IBM Integration Designer.

About this task

You can invoke the external service wizard to modify the information of a service export interface. The wizard automatically populates the existing information for the selected export interface. You can modify the objects and services, and then regenerate the export component with the modified data.

Procedure

1. Invoke the external service wizard for the selected service interface export component using one of the following methods.
 - In the assembly editor, select the component you want to modify, right-click and select **Edit Binding**.
 - In the Business Integration view, select the component you want to modify, right-click and select **Edit Binding**.
 - Select the interface in the assembly editor and select the Properties view. In the Binding tab, click the **Edit** link.

The Find Objects in the Enterprise System window of the external service wizard is displayed. The external service wizard automatically populates the existing configuration details for the selected export interface.

2. In the Find Objects in the Enterprise System window, you can search for the SAP objects, select the objects you want to use in your module, configure each business object, modify an existing business object or remove an existing object. For more information about discovering objects for the BAPI interface, see "Selecting business objects and services for ALE inbound processing" on page 189.
3. Click **Next**. If you click **Cancel**, the changes you made in the previous step does not take effect.
4. In the Specify Composite Properties window, specify properties that apply to all business objects. For more information, see "Configuring the selected objects" on page 198.
5. Click **Next**.

6. In the Service Generation window, modify the service operations if required.
7. Click **Finish**. The artifacts are updated.

Results

The artifacts are updated.

What to do next

You can test and deploy your module.

Modifying service import for Query interface for SAP software outbound processing

Modify an import component by rediscovering and reconfiguring the objects using the Edit Binding option in IBM Integration Designer.

About this task

You can invoke the external service wizard to modify the information of a service import interface. The wizard automatically populates the existing information for the selected import interface. You can modify the objects and services, and then regenerate the import component with the modified data.

Procedure

1. Invoke the external service wizard for the selected service interface import component using one of the following methods.
 - In the assembly editor, select the component you want to modify, right-click and select **Edit Binding**.
 - In the Business Integration view, select the component you want to modify, right-click and select **Edit Binding**.
 - Select the interface in the assembly editor and select the Properties view. In the Binding tab, click the **Edit** link.

The Find Objects in the Enterprise System window of the external service wizard is displayed. The external service wizard automatically populates the existing configuration details for the selected import interface.

2. In the Find Objects in the Enterprise System window, you can search for the SAP objects, select the objects you want to use in your module, configure each business object, modify an existing business object or remove an existing object. For more information about discovering objects for the Query interface, see "Selecting business objects and services" on page 159.
3. Click **Next**. If you click **Cancel**, the changes you made in the previous step does not take effect.
4. In the Specify Composite Properties window, specify properties that apply to all business objects. For more information, see "Configuring the selected objects" on page 164.
5. Click **Next**.
6. In the Service Generation window, modify the service operations if required.
7. Click **Finish**. The artifacts are updated.

Results

The artifacts are updated.

What to do next

You can test and deploy your module.

Modifying service import for Advanced event processing outbound

Modify an import component by rediscovering and reconfiguring the objects using the Edit Binding option in IBM Integration Designer.

About this task

You can invoke the external service wizard to modify the information of a service import interface. The wizard automatically populates the existing information for the selected import interface. You can modify the objects and services, and then regenerate the import component with the modified data.

Procedure

1. Invoke the external service wizard for the selected service interface import component using one of the following methods.
 - In the assembly editor, select the component you want to modify, right-click and select **Edit Binding**.
 - In the Business Integration view, select the component you want to modify, right-click and select **Edit Binding**.
 - Select the interface in the assembly editor and select the Properties view. In the Binding tab, click the **Edit** link.

The Find Objects in the Enterprise System window of the external service wizard is displayed. The external service wizard automatically populates the existing configuration details for the selected import interface.

2. In the Find Objects in the Enterprise System window, you can search for the SAP objects, select the objects you want to use in your module, configure each business object, modify an existing business object or remove an existing object. For more information about discovering objects for the BAPI interface, see "Selecting business objects and services for Advanced event (outbound) processing" on page 170.
3. Click **Next**. If you click **Cancel**, the changes you made in the previous step does not take effect.
4. In the Specify Composite Properties window, specify properties that apply to all business objects. For more information, see "Configuring the selected objects" on page 172.
5. Click **Next**.
6. In the Service Generation window, modify the service operations if required.
7. Click **Finish**. The artifacts are updated.

Results

The artifacts are updated.

What to do next

You can test and deploy your module.

Modifying service export for Advanced event processing inbound

Modify an export component by rediscovering and reconfiguring the objects using the Edit Binding option in IBM Integration Designer.

About this task

You can invoke the external service wizard to modify the information of a service export interface. The wizard automatically populates the existing information for the selected export interface. You can modify the objects and services, and then regenerate the export component with the modified data.

Procedure

1. Invoke the external service wizard for the selected service interface export component using one of the following methods.
 - In the assembly editor, select the component you want to modify, right-click and select **Edit Binding**.
 - In the Business Integration view, select the component you want to modify, right-click and select **Edit Binding**.
 - Select the interface in the assembly editor and select the Properties view. In the Binding tab, click the **Edit** link.

The Find Objects in the Enterprise System window of the external service wizard is displayed. The external service wizard automatically populates the existing configuration details for the selected export interface.

2. In the Find Objects in the Enterprise System window, you can search for the SAP objects, select the objects you want to use in your module, configure each business object, modify an existing business object or remove an existing object. For more information about discovering objects for the BAPI interface, see "Selecting business objects and services for Advanced event (inbound) processing" on page 214.
3. Click **Next**. If you click **Cancel**, the changes you made in the previous step does not take effect.
4. In the Specify Composite Properties window, specify properties that apply to all business objects. For more information, see "Configuring the selected objects" on page 216.
5. Click **Next**. If you click **Cancel**, the changes you made in the previous step does not take effect.
6. In the Service Generation window, modify the service operations if required.
7. Click **Finish**. The artifacts are updated.

Results

The artifacts are updated.

What to do next

You can test and deploy your module.

Chapter 8. Deploying the module

Deploy a module to place the files that make up your module and adapter into an operational environment for production or testing. In IBM Integration Designer, the integrated test environment features runtime support for IBM Business Process Manager or WebSphere Enterprise Service Bus, or both, depending on the test environment profiles that you selected during installation.

Deployment environments

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

In IBM Integration Designer, you can deploy your modules to one or more servers in the test environment. This is typically the most common practice for running and testing business integration modules. However, you can also export modules for server deployment on IBM Business Process Manager or WebSphere Enterprise Service Bus as EAR files using the Process Administrative Console or command-line tools.

Deploying the module for testing

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

Adding external software dependencies

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

About this task

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

Adding external software dependencies on the server

You must copy the required `sapjco3.jar` jar file and related files to your runtime environment before you can run your adapter applications.

About this task

To obtain the required files and copy them to IBM Business Process Manager or WebSphere Enterprise Service Bus, use the following procedure.

1. Obtain the `sapjco3.jar` file and the associated files for your operating system from your SAP administrator or from the SAP Web site. Also obtain the

CWYAP_SAPAdapterExt.jar in the <IID_INSTALL_ROOT>>/ResourceAdapters/SAP_7.0.0.0/ext folder from the adapter. The files are listed in the following table.

Note: The software dependencies differ, depending on the version of SAP Software Tools you use.

Table 22. External software dependency files required by SAP Software

Operating system	Files to be copied
Windows	sapjco3.jar, sapidoc3.jar, and any *.dll files that come with the SAP JCo download from the SAP Web site. CWYAP_SAPAdapterExt.jar from the SAP adapter.
UNIX (including UNIX System Services on z/OS)	sapjco3.jar, sapidoc3.jar and any .so and .o files that come with the SAP JCo download from the SAP Web site. CWYAP_SAPAdapterExt.jar from the SAP adapter.

2. For SAP Releases 6.40 and 7.0, unzip the R3DLLINST.ZIP archive (C runtime 7.1) from the SAP note 684106 and run the "R3DLLINS.EXE" file in the subdirectory NTPATCH. For SAP releases 4.6D EX2, Web AS 6.40 EX2, SAP NetWeaver 7.01 and 7.10 and higher, download the installation program **vcredist_<platform>.exe**. Then process the program. The vcredist_<platform > installation packages are delivered with the installation master DVDs of SAP releases 7.01 and 7.10 and are located in NPATCH directory.
3. SAP JCo requires **dbghelp.dll** on Windows environment. This dll is found in the system32 directory on most Windows systems. Copy this dll onto your Windows environment if you do not have it.
4. Copy the files to the server.
 - In a testing environment in IBM Integration Designer, copy the files to the appropriate directory such as \${IID_INSTALL_ROOT}/runtimes/bi_v7/lib/ext directory or \${WPS_INSTALL_ROOT}/lib/ext directory.
 - In a production environment, copy the files to the \${WPS_INSTALL_ROOT}/lib/ext directory of IBM Business Process Manager or WebSphere Enterprise Service Bus.
 - For z/OS, add the specified files to the following locations:
 - a. Add the sapjco3.jar file, sapidoc3.jar file, and the CWYAP_SAPAdapterExt.jar file to the \${}/classes directory.
 - b. Add the .so files to the \${WPS_INSTALL_ROOT}/lib directory.
 - For IBM i system, add the specified files and variables to the /SAPJCO directory:
 - a. Add sapjco3.jar, sapidoc3.jar file, and any *.dll files that come with the SAP JCo download from the SAP Web site.
 - b. Add the CWYAP_SAPAdapterExt.jar from the SAP adapter.
 - c. Set the following at the *SYS level on the IBMi server:
 - Add a **LIBPATH** variable for the SAPJCO folder
 - Add a **CLASSPATH** variable that points to the /SAPJCO/sapjco3.jar directory
 - Add a system-wide variable **QIBM_JAVA_PASE_STARTUP** that points to the /usr/lib/start64 directory
 - d. Restart the server and deploy the application again
 - For all other operating systems, add the specified files to the following locations:

- a. Add the SAP Java Connector interface (sapjco3.jar, sapidoc3.jar, and CWYAP_SAPAdapterExt.jar) to the lib subdirectory of the IBM Business Process Manager or WebSphere Enterprise Service Bus.
- b. Add the other SAP JCO files to the bin subdirectory of the IBM Business Process Manager or WebSphere Enterprise Service Bus installation directory.

The installation directory is typically in the `runtimes\bi_v6` directory of the IBM Integration Designer installation directory.

Results

The `sapjco3.jar` and associated files are now part of your runtime environment.

Adding external software dependencies when the adapter is bundled

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

About this task

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

Procedure

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

Results

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

Generating and wiring a target component for testing inbound processing

Before deploying to the test environment a module that includes an adapter for inbound processing, you must first generate and wire a target component. This target component serves as the *destination* to which the adapter sends events.

Before you begin

You must have generated an export module, using the external service wizard.

About this task

Generating and wiring a target component for inbound processing is required in a testing environment only. It is not necessary when deploying the adapter in a production environment.

The target component receives events. You *wire* the export to the target component (connecting the two components) using the assembly editor in IBM Integration Designer. The adapter uses the wire to pass event data (from the export to the target component).

Procedure

1. Create the target component.
 - a. From the Business Integration perspective of IBM Integration Designer, expand **Assembly Diagram** and double-click the export component. If you did not change the default value, the name of the export component is the name of your adapter + **InboundInterface**.

An interface specifies the operations that can be called and the data that is passed, such as input arguments, returned values, and exceptions. The **InboundInterface** contains the operations required by the adapter to support inbound processing and is created when you run the external service wizard.
 - b. Create a new component by expanding **Components**, selecting **Untyped Component**, and dragging the component to the Assembly Diagram. The cursor changes to the placement icon.
 - c. Click the component to have it displayed in the Assembly Diagram.
2. Wire the components.
 - a. Click and drag the export component to the new component.
 - b. Save the assembly diagram. Click **File > Save**.
3. Generate an implementation for the new component.
 - a. Right-click on the new component and select **Generate Implementation > Java**.
 - b. Select **(default package)** and click **OK**. This creates an endpoint for the inbound module.

The Java implementation is displayed in a separate tab.
 - c. **Optional:** Add print statements to print the data object received at the endpoint for each of the endpoint methods.
 - d. Click **File > Save** to save the changes.

What to do next

Continue deploying the module for testing.

Adding the module to the server

In IBM Integration Designer, you can add modules to one or more servers in the test environment.

Before you begin

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

About this task

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

Procedure

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

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

What to do next

Test the functionality of your module and the adapter.

Testing the module for outbound processing using the test client

Test the assembled module and adapter for outbound processing using the IBM Integration Designer integration test client.

Before you begin

You need to add the module to the server first.

About this task

Testing a module is performed on the interface operations of your components, which enables you to determine whether the components are correctly implemented and the references are correctly wired.

Procedure

1. Select the module you want to test, right-click on it, and select **Test > Test Module**.
2. For information about testing a module using the test client, see the *Testing modules and components* topic in the IBM Integration Designer information center.

What to do next

If you are satisfied with the results of testing your module and adapter, you can deploy the module and adapter to the production environment.

Deploying the module for production

Deploying a module created with the external service wizard to IBM Business Process Manager or WebSphere Enterprise Service Bus in a production environment is a two-step process. First, you export the module in IBM Integration Designer as an enterprise archive (EAR) file. Second, you deploy the EAR file using the IBM Business Process Manager or WebSphere Enterprise Service Bus Process Administrative Console.

Adding external software dependencies on the server

You must copy the required `sapjco3.jar` jar file and related files to your runtime environment before you can run your adapter applications.

About this task

To obtain the required files and copy them to IBM Business Process Manager or WebSphere Enterprise Service Bus, use the following procedure.

1. Obtain the `sapjco3.jar` file and the associated files for your operating system from your SAP administrator or from the SAP Web site. Also obtain the `CWYAP_SAPAdapterExt.jar` in the `<IID_INSTALL_ROOT>/ResourceAdapters/SAP_7.0.0.0/ext` folder from the adapter. The files are listed in the following table.

Note: The software dependencies differ, depending on the version of SAP Software Tools you use.

Table 23. External software dependency files required by SAP Software

Operating system	Files to be copied
Windows	<code>sapjco3.jar</code> , <code>sapidoc3.jar</code> , and any *.dll files that come with the SAP JCo download from the SAP Web site. <code>CWYAP_SAPAdapterExt.jar</code> from the SAP adapter.
UNIX (including UNIX System Services on z/OS)	<code>sapjco3.jar</code> , <code>sapidoc3.jar</code> and any .so and .o files that come with the SAP JCo download from the SAP Web site. <code>CWYAP_SAPAdapterExt.jar</code> from the SAP adapter.

2. For SAP Releases 6.40 and 7.0, unzip the `R3DLLINST.ZIP` archive (C runtime 7.1) from the SAP note 684106 and run the "`R3DLLINS.EXE`" file in the subdirectory `NTPATCH`. For SAP releases 4.6D EX2, Web AS 6.40 EX2, SAP NetWeaver 7.01 and 7.10 and higher, download the installation program `vcredist_<platform>.exe`. Then process the program. The `vcredist_<platform >` installation packages are delivered with the installation master DVDs of SAP releases 7.01 and 7.10 and are located in `NPATCH` directory.
3. SAP JCo requires `dbghelp.dll` on Windows environment. This dll is found in the `system32` directory on most Windows systems. Copy this dll onto your Windows environment if you do not have it.
4. Copy the files to the server.
 - In a testing environment in IBM Integration Designer, copy the files to the appropriate directory such as `${IID_INSTALL_ROOT}/runtimes/bi_v7/lib/ext` directory or `${WPS_INSTALL_ROOT}/lib/ext` directory.
 - In a production environment, copy the files to the `${WPS_INSTALL_ROOT}/lib/ext` directory of IBM Business Process Manager or WebSphere Enterprise Service Bus.
 - For z/OS, add the specified files to the following locations:

- a. Add the sapjco3.jar file, sapidoc3.jar file, and the CWYAP_SAPAdapterExt.jar file to the \${}/classes directory.
- b. Add the .so files to the \${WPS_INSTALL_ROOT}/lib directory.
- For IBM i system, add the specified files and variables to the /SAPJCO directory:
 - a. Add sapjco3.jar, sapidoc3.jar file, and any *.dll files that come with the SAP JCo download from the SAP Web site.
 - b. Add the CWYAP_SAPAdapterExt.jar from the SAP adapter.
 - c. Set the following at the *SYS level on the IBMi server:
 - Add a **LIBPATH** variable for the SAPJCO folder
 - Add a **CLASSPATH** variable that points to the /SAPJCO/sapjco3.jar directory
 - Add a system-wide variable **QIBM_JAVA_PASE_STARTUP** that points to the /usr/lib/start64 directory
 - d. Restart the server and deploy the application again
- For all other operating systems, add the specified files to the following locations:
 - a. Add the SAP Java Connector interface (sapjco3.jar, sapidoc3.jar, and CWYAP_SAPAdapterExt.jar) to the lib subdirectory of the IBM Business Process Manager or WebSphere Enterprise Service Bus.
 - b. Add the other SAP JCO files to the bin subdirectory of the IBM Business Process Manager or WebSphere Enterprise Service Bus installation directory.

The installation directory is typically in the runtimes\bi_v6 directory of the IBM Integration Designer installation directory.

Results

The sapjco3.jar and associated files are now part of your runtime environment.

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

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

Before you begin

You must set **Deploy connector project** to **On server for use by multiple adapters** in the Specify the Service Generation and Deployment Properties window of the external service wizard.

About this task

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

Procedure

1. If the server is not running, right-click your server in the **Servers** view and select **Start**.

2. When the server status changes to **Started**, right-click the server and select **Administration > Run administrative console**.
3. Log on to the administrative console.
4. Click **Resources > Resource Adapters > Resource adapters**.
5. In the Resource adapters page, click **Install RAR**.

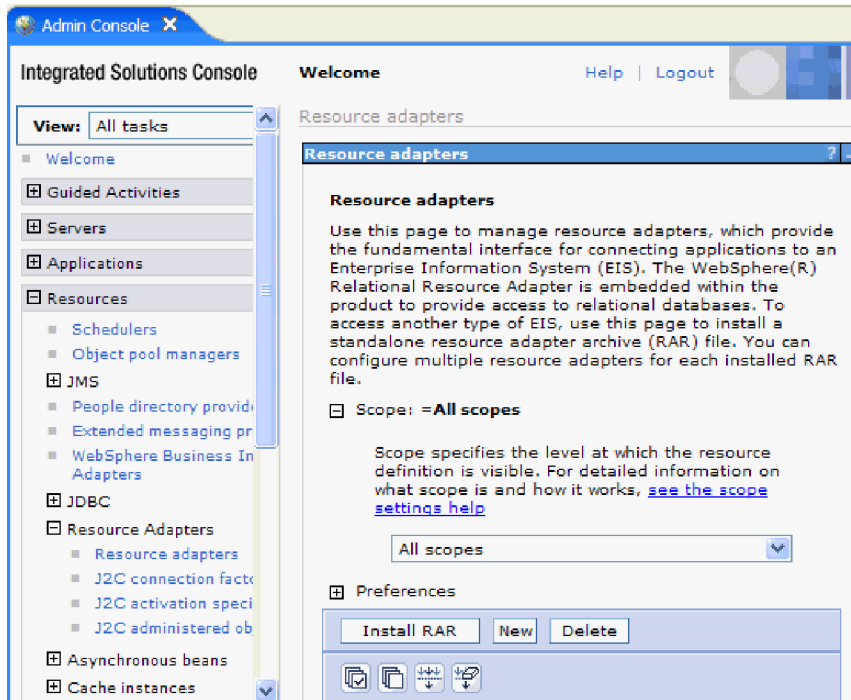


Figure 90. The Install RAR button on the Resource adapters page

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

What to do next

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

Exporting the module as an EAR file

Using IBM Integration Designer, export your module as an EAR file. By creating an EAR file, you capture all of the contents of your module in a format that can be easily deployed to IBM Business Process Manager or WebSphere Enterprise Service Bus.

Before you begin

Before you can export a module as an EAR file, you must have created a module to communicate with your service. The module should be displayed in the IBM Integration Designer Business Integration perspective.

About this task

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

Procedure

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

Results

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

What to do next

Install the module in the Process Administrative Console. This deploys the module to IBM Business Process Manager or WebSphere Enterprise Service Bus.

Installing the EAR file

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

Before you begin

You must have exported your module as an EAR file before you can install it on IBM Business Process Manager or WebSphere Enterprise Service Bus.

About this task

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

Procedure

1. If the server is not running, right-click your server in the **Servers** view and select **Start**.

2. When the server status changes to **Started**, right-click the server and select **Administration > Run administrative console**.
3. Log on to the administrative console.
4. Click **Applications > New Application > New Enterprise Application**.



Figure 91. Preparing for the application installation window

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

Results

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

What to do next

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

Deploying the module in a clustered environment

In IBM Integration Designer, you can deploy the IBM WebSphere Adapter for SAP Software in a clustered environment.

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

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

Deploying module embedded in the application

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

Before you begin

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

- IBM Integration Designer version 7.5.0.0 or above.
- A clustered topology deployment environment on the IBM Business Process Manager or WebSphere Enterprise Service Bus available from IBM Integration Designer.
- Create a clustered topology deployment environment, as shown in the following **Gold Topology** configuration figure.

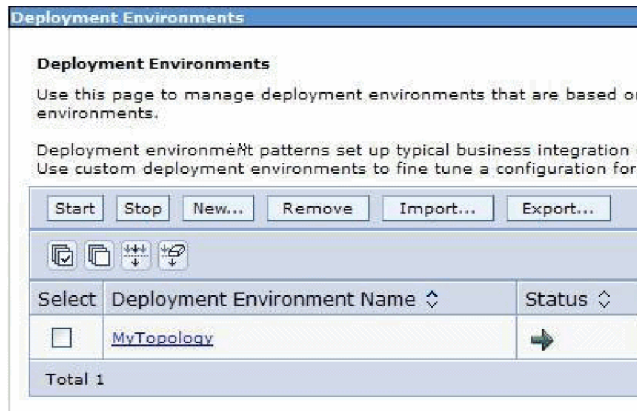


Figure 92. Deployment environment

- Deploy the adapter and the adapter applications (EAR files) in the AppTarget (the target that hosts the SCA container).

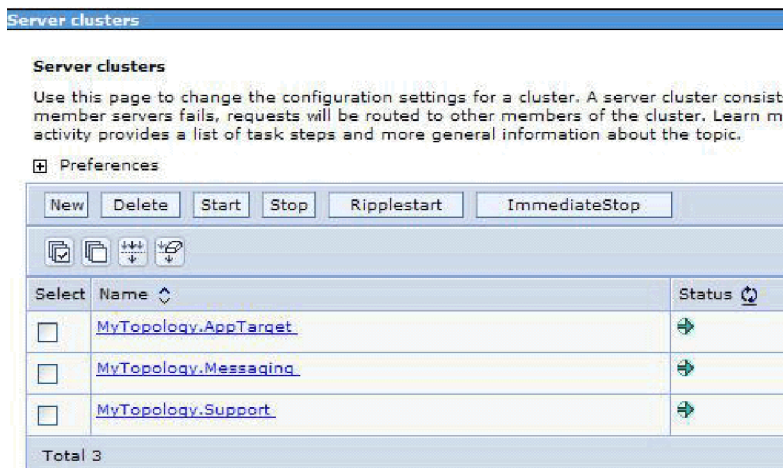


Figure 93. Deployment environment showing server clusters

About this task

To create an application with the embedded adapter, use the external service wizard.

Procedure

1. In the Service Configuration Properties window, from the **Deploy connector project** property list, select **With module for use by single application**.
2. Create the module as described in the Business process management samples for WebSphere Adapters.
3. In the **Dependencies** option for the module, after the module is created, ensure that the **Deploy with module** option is selected for the adapter.
4. If the server is not running, right-click your server in the **Servers** view and select **Start**.
5. When the server status changes to **Started**, right-click the server and select **Administration > Run administrative console**.

6. From the **Deployment Manager Admin Console**, click **Install applications** to deploy the application.
7. On the **Step 2: Map modules to servers** window, select the module and click **Next**. For the embedded adapter option, the adapter is deployed as part of the application.
8. In the Enterprise Applications view, select the new application `<adapter_name>EmbeddedModuleApp`. The new application is displayed after the application is deployed at the deployment manager level.
9. Select the node and click **Installed applications** to view the deployed application on each individual node.

Results

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

Deploying module at node level with embedded activation specification

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

Before you begin

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

- IBM Integration Designer version 7.5.0.0 or above.
- A clustered topology deployment environment on the IBM Business Process Manager or WebSphere Enterprise Service Bus available from IBM Integration Designer.
- Create a clustered topology deployment environment, as shown in the following **Gold Topology** configuration figure.

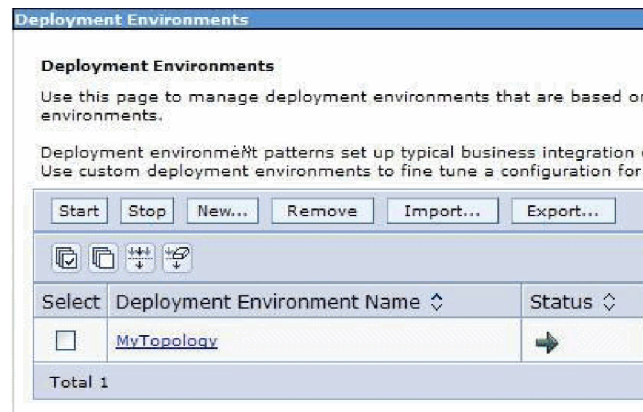


Figure 94. Deployment environment

- Deploy the adapter and the adapter applications (EAR files) in the AppTarget (the target that hosts the SCA container).

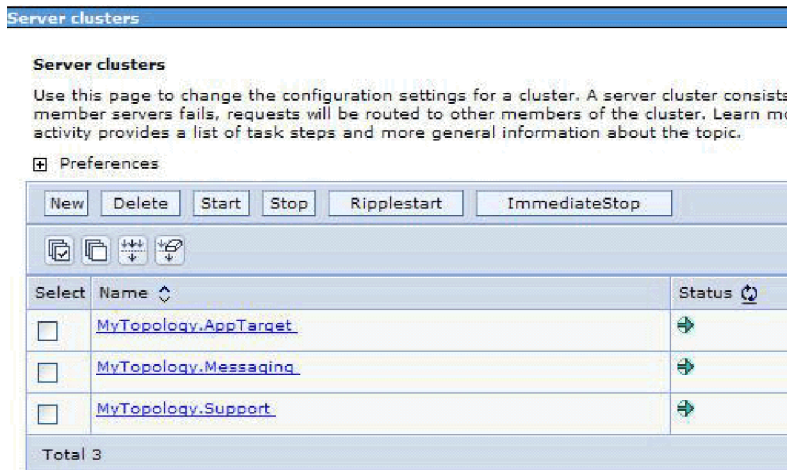


Figure 95. Deployment environment showing server clusters

About this task

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

Procedure

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

13. For node level deployment, check if the adapter RAR is deployed at the node level.
14. To deploy the adapter at the cluster level, click **Resources > Resource Adapters > Resource adapters**.
15. In the Resource adapters window, set the **Scope** to **Cluster**, and then click **New**.
16. Select the RAR deployed at the node level.
17. Check if the adapter RAR is now deployed at the cluster level. Deploy the application after the adapter is deployed at the node level on the individual nodes, and then at the cluster level.
18. From the **Deployment Manager Admin Console**, click **Install applications** to deploy the application.
19. On the **Step 2: Map modules to servers** window, select the module and click **Next**. The adapter is not part of the deployed application.
20. In the **Admin Console**, click **Resources > Resource Adapters > IBM WebSphere Adapter for SAP Software > J2C activation specifications** to view the activation specification from the adapter deployed at the cluster level.

Results

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

Deploying module at node level with JNDI activation specification

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

Before you begin

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

- IBM Integration Designer version 7.5.0.0 or above.
- A clustered topology deployment environment on the IBM Business Process Manager or WebSphere Enterprise Service Bus available from IBM Integration Designer.
- Create a clustered topology deployment environment, as shown in the following **Gold Topology** configuration figure.

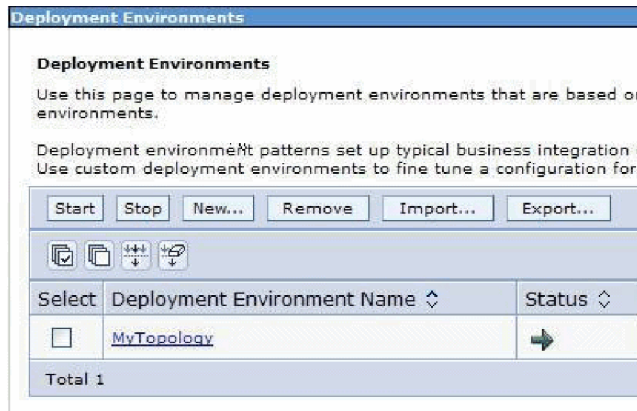


Figure 96. Deployment environment

- Deploy the adapter and the adapter applications (EAR files) in the AppTarget (the target that hosts the SCA container).

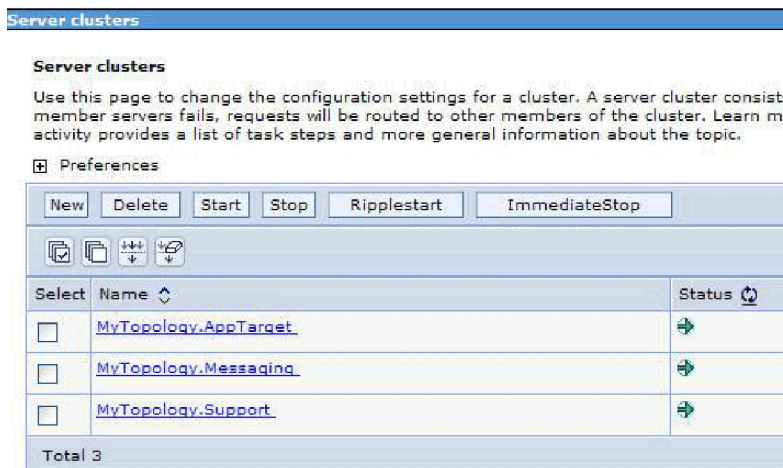


Figure 97. Deployment environment showing server clusters

About this task

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

Procedure

1. In the Service Configuration Properties window, from the **Deploy connector project** property list, select **On server for use by multiple applications**.
2. From the **Connection properties** list, select **Use JNDI lookup name configured on server**.
3. In the **JNDI lookup name** property field, specify the JNDI name. Use this same JNDI name when you create the activation specification from the Admin Console.
4. Create the module as described in the Business process management samples for WebSphere Adapters.
5. In the **Dependencies** option for the module, ensure that the **Deploy with module** option is not selected for the adapter.

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

Results

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

Chapter 9. Administering the adapter module

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

Configuring logging and tracing

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

Configuring logging properties

Use the Process 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 IBM 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.

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

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

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

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

Procedure

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

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

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

Results

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

Changing the log and trace file names

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

Before you begin

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

About this task

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

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

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

Procedure

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

Changing configuration properties for embedded adapters

To change the configuration properties after you deploy the adapter as part of a module, you use the administrative console of the runtime environment. You can update resource adapter properties (used for general adapter operation), managed connection factory properties (used for outbound processing), and activation specification properties (used for inbound processing). For information about configuring logging properties and changing the log and trace file names, see “Configuring logging and tracing” on page 251.

Setting resource adapter properties for embedded adapters

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

Before you begin

Your adapter module must be deployed on IBM Business Process Manager or WebSphere Enterprise Service Bus.

About this task

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

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

Procedure

1. If the server is not running, right-click your server in the **Servers** view and select **Start**.
2. When the server status changes to **Started**, right-click the server and select **Administration > Run administrative console**.
3. Log on to the administrative console.
4. Select **Applications > Application Types > WebSphere enterprise application**.
5. From the Enterprise Applications list, click the name of the adapter module whose properties you want to change. The **Configuration** page is displayed.

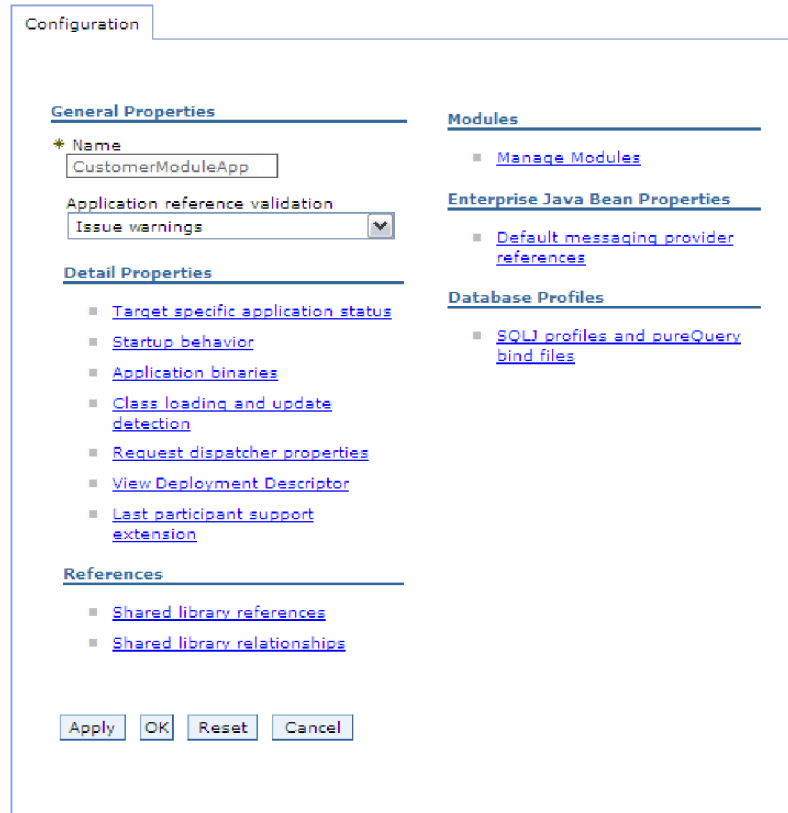


Figure 98. The Manage Modules selection in the Configuration tab

6. Under **Modules**, click **Manage Modules**.
7. Click **IBM WebSphere Adapter for SAP Software**.
8. From the **Additional Properties** list, click **Resource Adapter**.
9. On the next page, from the **Additional Properties** list, click **Custom properties**.
10. For each property you want to change, perform the following steps.

Note: See “Resource adapter properties” on page 332 for more information about these properties.

- a. Click the name of the property. The **Configuration** page for the selected property is displayed.
 - b. Change the contents of the **Value** field or type a value, if the field is empty.
 - c. Click **OK**.
11. In the Messages area, click **Save**.

Results

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

Setting managed (J2C) connection factory properties for embedded adapters

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

Before you begin

Your adapter module must be deployed on IBM Business Process Manager or WebSphere Enterprise Service Bus.

About this task

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

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

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

Procedure

1. If the server is not running, right-click your server in the **Servers** view and select **Start**.
2. When the server status changes to **Started**, right-click the server and select **Administration > Run administrative console**.
3. Log on to the administrative console.
4. Select **Applications > Application Types > WebSphere enterprise application**.
5. In the Enterprise Applications list, click the name of the adapter module whose properties you want to change.

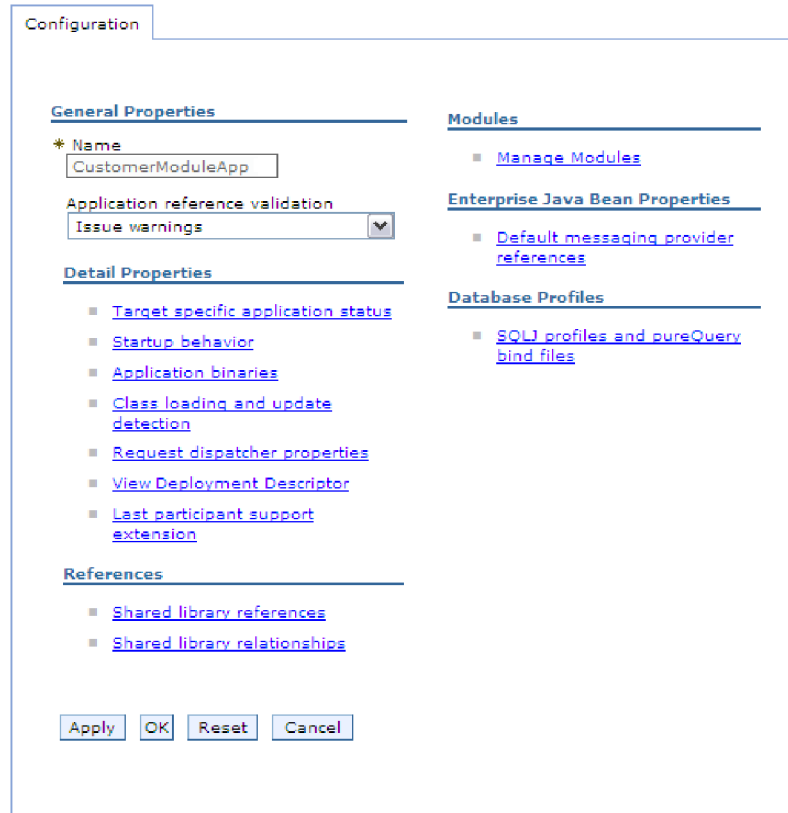


Figure 99. The Manage Modules selection in the Configuration tab

6. Under **Modules**, click **Manage Modules**.
7. Click **IBM WebSphere Adapter for SAP Software**.
8. In the **Additional Properties** list, click **Resource Adapter**.
9. On the next page, from the **Additional Properties** list, click **J2C connection factories**.
10. Click the name of the connection factory associated with your adapter module.
11. In the **Additional Properties** list, click **Custom properties**.
 Custom properties are those J2C connection factory properties that are unique to IBM WebSphere Adapter for SAP Software. Connection pool and advanced connection factory properties are properties you configure if you are developing your own adapter.
12. For each property you want to change, perform the following steps.

Note: See “Managed connection factory properties” on page 334 for more information about these properties.

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

Results

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

Setting activation specification properties for embedded adapters

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

Before you begin

Your adapter module must be deployed on IBM Business Process Manager 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.

Procedure

1. If the server is not running, right-click your server in the **Servers** view and select **Start**.
2. When the server status changes to **Started**, right-click the server and select **Administration > Run administrative console**.
3. Log on to the administrative console.
4. Select **Applications > Application Types > WebSphere enterprise application**.
5. From the Enterprise Applications list, click the name of the adapter module whose properties you want to change.

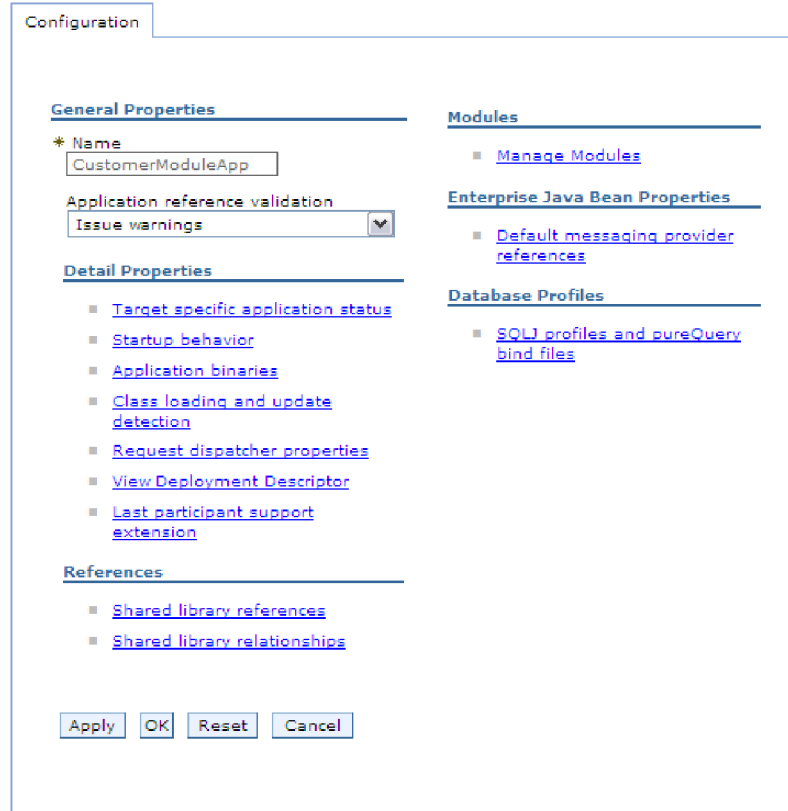


Figure 100. The Manage Modules selection in the Configuration tab

6. Under **Modules**, click **Manage Modules**.
7. Click **IBM WebSphere Adapter for SAP Software**.
8. From the **Additional Properties** list, click **Resource Adapter**.
9. On the next page, from the **Additional Properties** list, click **J2C activation specifications**.
10. Click the name of the activation specification associated with the adapter module.
11. From the **Additional Properties** list, click **J2C activation specification custom properties**.
12. For each property you want to change, perform the following steps.

Note: See “Activation specification properties for ALE inbound processing” on page 383, “Activation specification properties for BAPI inbound processing” on page 366, or “Activation specification properties for Advanced event processing” on page 405 for information about these properties.

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

Results

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

Changing configuration properties for stand-alone adapters

To set configuration properties after you install a stand-alone adapter, use the administrative console of the runtime environment. Provide the general information about the adapter and then set the resource adapter properties (which are used for general adapter operation). If the adapter is used for outbound operations, create a connection factory and then set the properties for it. If the adapter is used for inbound operations, create an activation specification and then set the properties for it. For information about configuring logging properties and changing the log and trace file names, see “Configuring logging and tracing” on page 251.

Setting resource adapter properties for stand-alone adapters

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

Before you begin

Your adapter must be installed on IBM Business Process Manager or WebSphere Enterprise Service Bus.

About this task

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

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

Procedure

1. If the server is not running, right-click your server in the **Servers** view and select **Start**.
2. When the server status changes to **Started**, right-click the server and select **Administration > Run administrative console**.
3. Log on to the administrative console.
4. Click **Resources > Resource Adapters > Resource adapters**.
5. In the Resource adapters page, click **IBM WebSphere Adapter for SAP Software**.
6. In the **Additional Properties** list, click **Custom properties**.
7. For each property you want to change, perform the following steps.

Note: See “Resource adapter properties” on page 332 for more information about these properties.

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

Results

The resource adapter properties associated with your adapter are changed.

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

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

Before you begin

Your adapter must be installed on IBM Business Process Manager or WebSphere Enterprise Service Bus.

About this task

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

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

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

Procedure

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

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

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

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

The newly created connection factory is displayed.

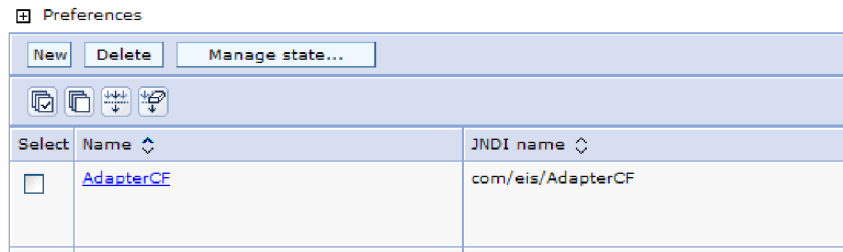


Figure 101. User-defined connection factories for use with the resource adapter

8. In the list of connection factories, click the one you want to use.
9. In the **Additional Properties** list, click **Custom properties**.
Custom properties are those J2C connection factory properties that are unique to WebSphere Adapter for SAP Software. Connection pool and advanced connection factory properties are properties you configure if you are developing your own adapter.
10. For each property you want to change, perform the following steps.
 - Note:** See “Managed connection factory properties” on page 334 for more information about these properties.
 - a. Click the name of the property.
 - b. Change the contents of the **Value** field or type a value, if the field is empty.
 - c. Click **OK**.
11. After you have finished setting properties, click **Apply**.
12. In the Messages area, click **Save**.

Results

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

Setting activation specification properties for stand-alone adapters

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

Before you begin

Your adapter must be installed on IBM Business Process Manager 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.

Procedure

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

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

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

- a. Click **New**.
 - b. In the **General Properties** section of the **Configuration** tab, type a name for the activation specification. For example, you can type AdapterAS.
 - c. Type a value for **JNDI name**. For example, you can type com/eis/AdapterAS.
 - d. Select the authentication alias from the **Authentication alias** list if you have configured the adapter interface for an authentication alias. If the authentication alias is not present in the list, create one. Refer to the related links section for more information about creating an authentication alias.
 - e. Select a message listener type. The available listener types correspond to:
 - The ALE inbound processing interface
 - The ALE inbound processing interface with local transaction support
 - The BAPI inbound processing interface
 - The Advanced event processing inbound interface
 - f. Click **OK**.
 - g. Click **Save** in the **Messages** box at the top of the page.
The newly created activation specification is displayed.
8. In the list of activation specifications, click the one you want to use.
 9. In the Additional Properties list, click **J2C activation specification custom properties**.
 10. For each property you want to set, perform the following steps.

Note: See “Activation specification properties for ALE inbound processing” on page 383, “Activation specification properties for BAPI inbound processing” on page 366, or “Activation specification properties for Advanced event processing” on page 405 for information about these properties.

- a. Click the name of the property.
 - b. Change the contents of the **Value** field or type a value, if the field is empty.
 - c. Click **OK**.
11. After you have finished setting properties, click **Apply**.

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

Results

The activation specification properties associated with your adapter are set.

Related tasks:

“Creating an authentication alias” on page 104

An authentication alias is a feature that encrypts the password used by the adapter to access the SAP server. The adapter can use it to connect to the SAP server instead of using a user ID and password stored in an adapter property.

Starting the application that uses the adapter

Use the administrative console of the server to start an application that uses the adapter. By default, the application starts automatically when the server starts.

About this task

Use this procedure to start the application, whether it is using an embedded or a stand-alone adapter. For an application that uses an embedded adapter, the adapter starts when the application starts. For an application that uses a stand-alone adapter, the adapter starts when the application server starts.

Procedure

1. If the server is not running, right-click your server in the **Servers** view and select **Start**.
2. When the server status changes to **Started**, right-click the server and select **Administration > Run administrative console**.
3. Log on to the administrative console.
4. Click **Applications > Application Types > WebSphere enterprise applications**.

Note: The administrative console is labeled “Integrated Solutions Console”.

5. Select the application that you want to start. The application name is the name of the EAR file you installed, without the .EAR file extension.
6. Click **Start**.

Results

The status of the application changes to **Started**, and a message stating that the application has started displays at the top of the administrative console.

Stopping the application that uses the adapter

Use the administrative console of the server to stop an application that uses the adapter. By default, the application stops automatically when the server stops.

About this task

Use this procedure to stop the application, whether it is using an embedded or a stand-alone adapter. For an application with an embedded adapter, the adapter stops when the application stops. For an application that uses a stand-alone adapter, the adapter stops when the application server stops.

Procedure

1. If the server is not running, right-click your server in the **Servers** view and select **Start**.
2. When the server status changes to **Started**, right-click the server and select **Administration > Run administrative console**.
3. Log on to the administrative console.
4. Click **Applications > Application Types > WebSphere enterprise applications**.

Note: The administrative console is labeled “Integrated Solutions Console”.

5. Select the application that you want to stop. The application name is the name of the EAR file you installed, without the .EAR file extension.
6. Click **Stop**.

Results

The status of the application changes to **Stopped**, and a message stating that the application has stopped is displayed at the top of the administrative console.

Managing Advanced event processing

To manage the Advanced event processing interface, use the IBM WebSphere BI Station tool. You can view and maintain events in the current events queue, future events queue, and archive events queue, and you can view and maintain adapter log files. In addition, you can maintain the SAP gateway service connections.

Displaying the current events queue

You can display the outgoing current events queue to check for events that have not yet been retrieved by WebSphere Adapter for SAP Software.

Before you begin

Make sure you have successfully installed the IBM WebSphere BI Station tool on the SAP server.

About this task

The events in the current events queue are awaiting retrieval by the adapter. You can display the queue to check the status of the events.

To display the contents of the current events queue, use the following procedure.

Procedure

1. If IBM WebSphere BI Station is not currently displayed, enter transaction `/n/CWLD/HOME_AEP`.
2. To display the Management page, click **Management**.
3. Under **Event Queues**, click **Current Events**.
4. Display the current events queue by performing one of the following steps from the Current Event Selection page:
 - To display all events in the current events queue, click **Execute**.
 - To limit the number of events that are displayed, enter values in one or more fields, or use the arrow keys to select values for the fields, and click **Execute**.

For example, to display only those entries associated with a particular business object, enter the name of the business object in the **Object Name** field, or click the **Object Name** field and select a value from the list.

Results

A list of events is displayed.

WebSphere BI: Current Events

Event ID	Stat	Object Name	Verb
A total of 29 events were selected from the event table.			
<input type="checkbox"/>	2 R	SAP4_CustomerPartner	Create
<input type="checkbox"/>	3 R	SAP4_CustomerMaster	Create
<input type="checkbox"/>	4 R	SAP4_CustomerMaster	Create
<input type="checkbox"/>	5 R	SAP4_CustomerMaster	Create

Figure 102. The Current Events window

Displaying the future events queue

You can display the future events queue to check for events that have not yet been transferred to the current events queue.

Before you begin

Make sure you have successfully installed the IBM WebSphere BI Station tool on the SAP server.

About this task

The events in the future events queue are awaiting transfer to the current events queue. You can display the queue to check the status of the events.

To display the contents of the future events queue, use the following procedure.

Procedure

1. If IBM WebSphere BI Station is not currently displayed, enter transaction `/n/CWLD/HOME_AEP`.
2. To display the Management page, click **Management**.
3. Under **Event Queues**, click **Future Events**.
4. Display the future events queue by performing one of the following steps from the Future Event Selection page:
 - To display all events in the future events queue, click **Execute**.
 - To limit the number of events that are displayed, enter values in one or more fields, or use the arrow keys to select values for the fields, and click **Execute**.

For example, to display only those entries associated with a particular business object, enter the name of the business object in the **Object Name** field, or click the **Object Name** field and select a value from the list.

Results

A list of events is displayed.

Maintaining the archive table

Using the IBM WebSphere BI Station tool, you can display the archive table and determine the status of archived events. From the table, you can identify events that need to be resubmitted for polling when a runtime environment subscribes to them.

Before you begin

Make sure you have successfully installed the IBM WebSphere BI Station tool on the SAP server.

About this task

When you display events in the archive table, you can resubmit the events for processing, or you can delete the events from the table.

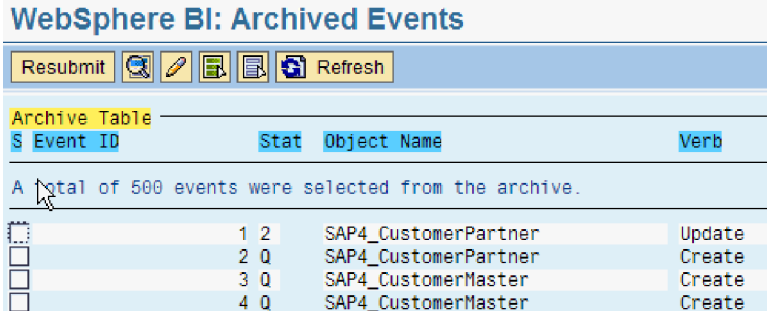
To maintain the archive table, perform one or more of the following steps.

Procedure

1. If IBM WebSphere BI Station is not currently displayed, enter transaction /n/CWLD/HOME_AEP.
2. To display the Management page, click **Management**.
3. Under **Event Queues**, click **Archived Events**.
4. Display the event queue by performing one of the following steps from the Archived Event Selection page:
 - a. To display all events, click the Execute button (F8).
 - b. To limit the number of events that are displayed, enter values in one or more fields, or use the arrow keys to select values for the fields.
For example, to display only those entries associated with a particular business object, enter the name of the business object in the **Object Name** field, or click **Object Name**, click the arrow button (F4), and then select the name from the list.

Results

A list of events is displayed.



WebSphere BI: Archived Events			
Archive Table			
S Event ID	Stat	Object Name	Verb
A total of 500 events were selected from the archive.			
<input type="checkbox"/>	1 2	SAP4_CustomerPartner	Update
<input type="checkbox"/>	2 0	SAP4_CustomerPartner	Create
<input type="checkbox"/>	3 0	SAP4_CustomerMaster	Create
<input type="checkbox"/>	4 0	SAP4_CustomerMaster	Create

Figure 103. The archived events table

What to do next

Resubmit one or more events for processing, or delete one or more events.

Resubmitting archived events

You can resubmit one or more events from the archive table to the event queue for reprocessing.

Before you begin

The Archived Events page should be displayed.

About this task

Resubmitting events moves the events from the archive table to the event table; however, the events do not pass through event distribution, event restriction, or event priority.

To resubmit one or more events, perform the following procedure.

Procedure

1. To select the event to be resubmitted, select the check box next to the name of the event. You can select multiple events.
2. Click **Resubmit**.

Results

The status of the operation is displayed.

Deleting events from the archive table

You can delete one or more events from the archive table. You can delete the files from the Management page, or you can schedule their deletion.

Before you begin

The Management page of IBM WebSphere BI Station should be displayed.

About this task

To delete events from the archive table, perform the following steps:

Procedure

1. If IBM WebSphere BI Station is not currently displayed, enter transaction `/n/CWLD/HOME_AEP`.
2. To display the Management page, click **Management**.
3. Under **Maintenance**, click **Delete Event Archive**.
4. From the WebSphere BI Delete Entries from Event Archive Table page, enter values for one or more fields to limit the events that are deleted.

For example, to delete only those entries associated with a particular business object, enter the name of the business object in the **Object Name** field, or click **Object Name**, click the arrow button (F4), and then select the name from the list.

5. Click the Execute button (F8).

Note: To schedule automatic deletion of archive events, contact your Basis administrator and schedule report `/CWLD/TRUN_EVENT_ARCHIVE_TAB`.

Results

The event or events are deleted.

Managing the adapter log file

The adapter log in the SAP application displays in reverse chronological order all events and errors that relate to the SAP server, such as Create or Update operations, or events that arrive in the event queue. The log file lists the date, time, and event for each log entry. The log file is a good source to start troubleshooting problems.

Setting logging options

You can specify the level of detail you want logged in the adapter log file, as well as the number of entries and type of data you want displayed.

Before you begin

Make sure you have successfully installed the IBM WebSphere BI Station tool on the SAP server.

About this task

To set the logging options, use the following procedure.

Procedure

1. If IBM WebSphere BI Station is not currently displayed, enter transaction `/n/CWLD/HOME_AEP`.
2. Click **Configuration**.
3. To set the logging level, select one of the values under **Logging Level**. The four levels of logging are shown in the following table:

Table 24. Logging levels

Level	Description	Recommended use
0	Off	Not recommended
1	Log only warnings and errors	Production system
2	Log every event with minimal information	
3	Log each event in detail, including every attribute of every business object	Development or debugging system

4. To change how many events are displayed, type a value in the **Number of entries to display in log** field.
5. To display only errors in the log, select **Display errors only**.
6. To display only entries for the user listed next to **User Name**, select **Display entries for this user**.
7. To specify how much (or how little) detail to display in the log, select one of the values under **Default Level of Detail to Display**.

Results

You have set the configuration settings that will be used when the log is displayed.

Displaying the adapter log

To view recently processed objects and the details associated with them, display the adapter log.

Before you begin

Make sure you have successfully installed the IBM WebSphere BI Station tool on the SAP server.

About this task

You can specify how much detail you want displayed, and you can filter the data so that only certain types of information are displayed.

To display the adapter log, use the following procedure.

Procedure

1. If IBM WebSphere BI Station is not currently displayed, enter transaction /n/CWLD/HOME_AEP.
2. To display the Management page, click **Management**.
3. Under **Activity**, click **Log**.
4. To change the amount of information that is displayed, click either **Fewer Details** or **More Details**.
5. To display only specific information, click **Filter Data**, enter values in the fields, and click **Filter**.

You can choose to display log entries associated with a specific user or with selected objects. You can display entries for a range of dates or a range of numbers. You can indicate how many entries should be displayed and whether to show errors and warnings only.

Results

The log is displayed.

Limiting the size of the adapter log

The adapter log can, over time, take up a significant amount of disk space. To save disk space, you can set this log to automatically truncate. When you set automatic truncation, by default SAP prints the truncated entries to the default printer of the user who set up the job. Therefore, you might also want to control the print options.

Before you begin

Make sure you have successfully installed the IBM WebSphere BI Station tool on the SAP server.

About this task

To limit the size of the adapter log, use the following procedure.

Procedure

1. If IBM WebSphere BI Station is not currently displayed, enter transaction /n/CWLD/HOME_AEP.
2. To display the Management page, click **Management**.

3. Under **Maintenance**, click **Delete Log**.
4. In the WebSphere BI Delete Log Entries page, enter values to indicate which log entries you want to delete.

You can delete a range of entries or the entries associated with a specific object. You can delete entries associated with a specific user or entries that were logged within a range of dates. You can also indicate that only entries older than a certain number of days should be deleted, and you can specify that a certain number of the most recent entries not be deleted.

The entries being deleted from the log are saved to the file specified in the **Output truncated data to** field.

5. Click the Execute button.

Note: To schedule the automatic truncation of the event log, set up the truncation options and contact your Basis administrator to schedule report /CWLD/DELETE_LOG.

Results

The specified log entries are deleted.

Monitoring SAP gateway connections

You can monitor the SAP gateway service connections between the adapter and the SAP application. Each entry displays information such as adapter host name, user name, and connection status.

Before you begin

Make sure you have successfully installed the IBM WebSphere BI Station tool on the SAP server.

About this task

To monitor the gateway connections, use the following procedure.

Procedure

1. If IBM WebSphere BI Station is not currently displayed, enter transaction /n/CWLD/HOME_AEP.
2. To display the Management page, click **Management**.
3. Under **Activity**, click **Gateway**.
4. Click a server name to see more details.

Results

A list of active connections is displayed.

Monitoring performance using Performance Monitoring Infrastructure

Performance Monitoring Infrastructure (PMI) is a feature of the administrative console that allows you to dynamically monitor the performance of components in the production environment, including IBM WebSphere Adapter for SAP Software. PMI collects adapter performance data, such as average response time and total number of requests, from various components in the server and organizes the data into a tree structure. You can view the data through the Tivoli® Performance

Viewer, a graphical monitoring tool that is integrated with the Process Administrative Console in IBM Business Process Manager or WebSphere Enterprise Service Bus.

About this task

You can monitor the performance of your adapter by having PMI collect data at the following points:

- At outbound processing to monitor outbound requests.
- At inbound event retrieval to monitor the retrieval of an event from the event table.
- At inbound event delivery to monitor the delivery of an event to the endpoint or endpoints.

Before you enable and configure PMI for your adapter, you must first set the level of tracing detail and run some events from which to gather performance data.

To learn more about how PMI can help you monitor and improve the overall performance of your adapter environment, search for PMI on the IBM Business Process Manager or WebSphere Enterprise Service Bus website:
<http://www.ibm.com/software/webservers/appserv/was/library/>.

Configuring Performance Monitoring Infrastructure

You can configure Performance Monitoring Infrastructure (PMI) to gather adapter performance data, such as average response time and total number of requests. After you configure PMI for your adapter, you can monitor the adapter performance using Tivoli Performance viewer.

Before you begin

Before you can configure PMI for your adapter, you must first set the level of tracing detail and run some events to gather the performance data.

1. To enable tracing and to receive event data, the trace level must be set to either fine, finer, finest, or all. After *=info, add a colon and a string, for example:
*=info: WBILocationMonitor.CEI.ResourceAdapter.
=finest: WBILocationMonitor.LOG.ResourceAdapter.=finest:

For instructions on setting the trace level, see “Enabling tracing with the Common Event Infrastructure” on page 275.

2. Generate at least one outbound request or inbound event to produce performance data that you can configure.

Procedure

1. Enable PMI for your adapter.
 - a. In the administrative console, expand **Monitoring and Tuning**, and then select **Performance Monitoring Infrastructure (PMI)**.
 - b. From the list of servers, click the name of your server.
 - c. Select the Configuration tab, and then select the **Enable Performance Monitoring (PMI)** check box.
 - d. Select **Custom** to selectively enable or disable statistics.

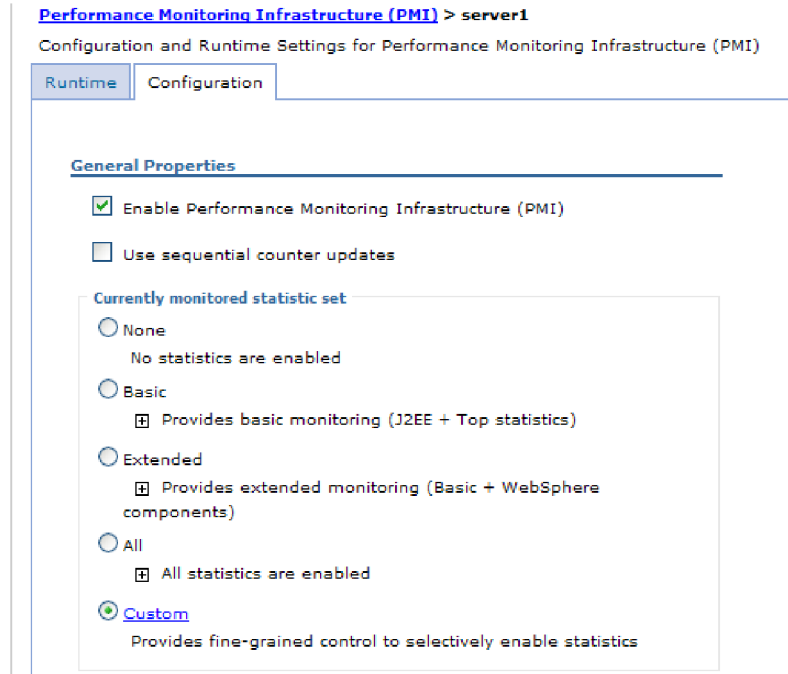


Figure 104. Enabling Performance Monitoring Infrastructure

- e. Click **Apply** or **OK**.
 - f. Click **Save**. PMI is now enabled.
2. Configure PMI for your adapter.
 - a. In the administrative console, expand **Monitoring and Tuning**, and then select **Performance Monitoring Infrastructure (PMI)**.
 - b. From the list of servers, click the name of your server.
 - c. Select **Custom**.
 - d. Select the **Runtime** tab. The following figure shows the Runtime tab.

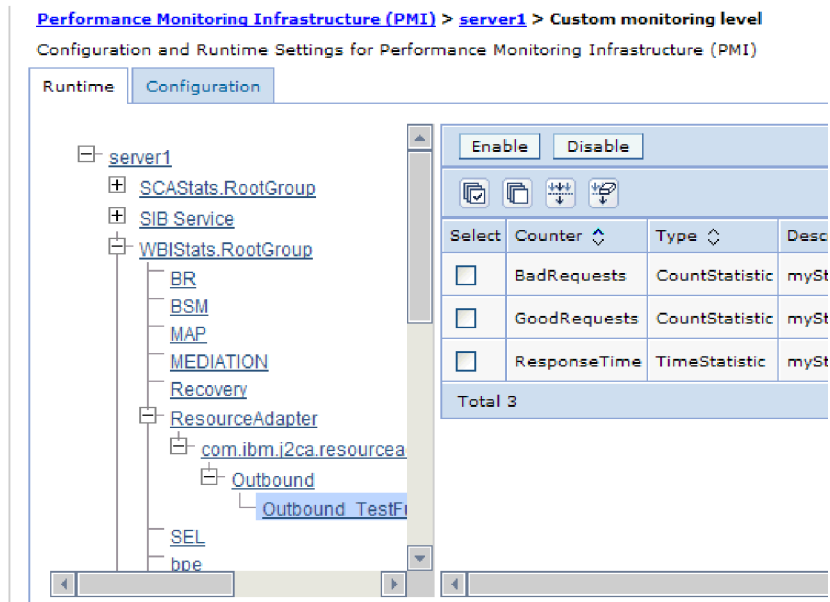


Figure 105. Runtime tab used for configuring PMI

- e. Click **WBISStats.RootGroup**. This is a PMI sub module for data collected in the root group. This example uses the name WBISStats for the root group.
- f. Click **ResourceAdapter**. This is a sub module for the data collected for the JCA adapters.
- g. Click the name of your adapter, and select the processes you want to monitor.
- h. In the right pane, select the check boxes for the statistics you want to gather, and then click **Enable**.

Results

PMI is configured for your adapter.

What to do next

Now you can view the performance statistics for your adapter.

Viewing performance statistics

You can view adapter performance data through the graphical monitoring tool, Tivoli Performance Viewer. Tivoli Performance Viewer is integrated with the Process Administrative Console in IBM Business Process Manager or WebSphere Enterprise Service Bus.

Before you begin

Configure Performance Monitoring Infrastructure for your adapter.

Procedure

1. In the administrative console, expand **Monitoring and Tuning**, expand **Performance Viewer**, then select **Current Activity**.
2. In the list of servers, click the name of your server.
3. Under your server name, expand **Performance Modules**.

4. Click **WBISStatsRootGroup**.
5. Click **ResourceAdapter** and the name of your adapter module.
6. If there is more than one process, select the check boxes for the processes whose statistics you want to view.

Results

The statistics are displayed in the right panel. You can click **View Graph** to view a graph of the data, or **View Table** to see the statistics in a table format.

The following figure shows adapter performance statistics.

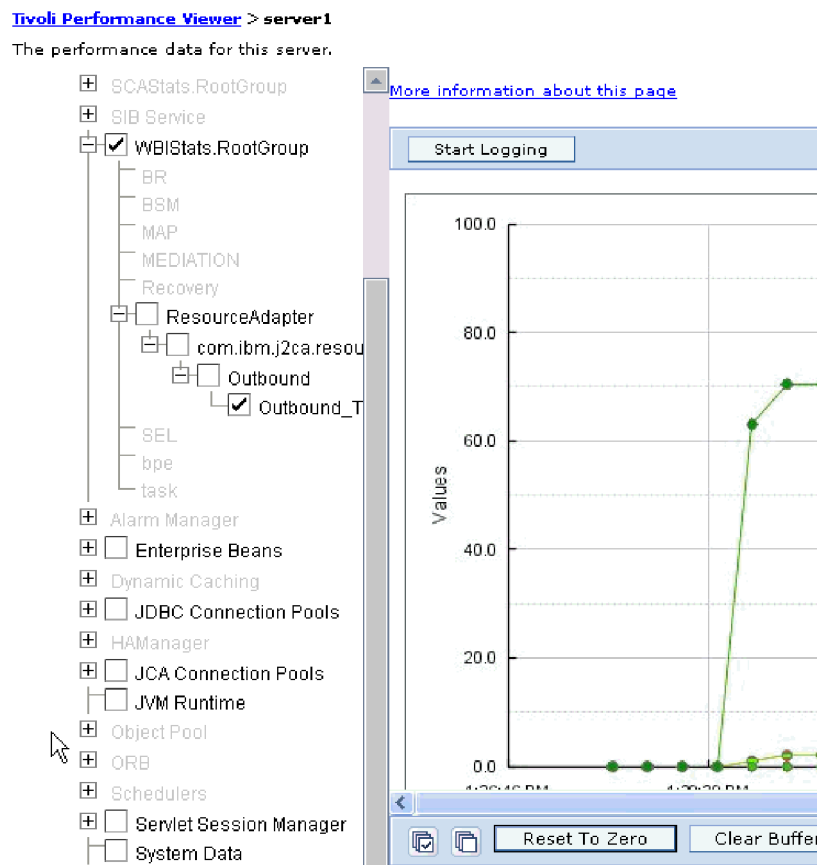


Figure 106. Adapter performance statistics, using graph view

Enabling tracing with the Common Event Infrastructure

The adapter can use the Common Event Infrastructure (CEI), a component embedded in the server, to report data about critical business events such as the starting or stopping of a poll cycle. Event data can be written to a database or a trace log file depending on configuration settings.

About this task

Use this procedure to report CEI entries in the trace log file by using the Common Base Event Browser within the administrative console.

Procedure

1. In the administrative console, click **Troubleshooting**.
2. Click **Logs and Trace**.
3. From the list of servers, click the name of your server.
4. In the **Change Log Detail Levels** box, click the name of the CEI database (for example, `WBIEventMonitor.CEI.ResourceAdapter.*`) or the trace log file (for example, `WBIEventMonitor.LOG.ResourceAdapter.*`) to which you want the adapter to write event data.
5. Select the level of detail about business events that you want the adapter to write to the database or trace log file, and (optionally) adjust the granularity of detail associated with messages and traces.
 - **No Logging**. Turns off event logging.
 - **Messages Only**. The adapter reports an event.
 - **All Messages and Traces**. The adapter reports details about an event.
 - **Message and Trace Levels**. Settings for controlling the degree of detail the adapter reports about the business object payload associated with an event. If you want to adjust the detail level, select one of the following options:
 - Fine**. The adapter reports the event but none of the business object payload.
 - Finer**. The adapter reports the event and the business object payload description.
 - Finest**. The adapter reports the event and the entire business object payload.
6. Click **OK**.

Results

Event logging is enabled. You can view CEI entries in the trace log file or by using the Common Base Event Browser within the administrative console.

Adding dependency libraries to the deployed resource adapter

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

Stand-alone deployment

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

EAR deployment

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

About this task

There are two methods to do this task:

- Using enhanced EAR editor in IBM Integration Designer
- Using administrative console of the IBM Business Process Manager or WebSphere Enterprise Service Bus

Using enhanced EAR editor

You can use the EAR editor in IBM Integration Designer to add the dependency libraries.

About this task

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

Procedure

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

Results

The dependent libraries are added as shared libraries.

Using administrative console of the WebSphere Application Server

You can use the administrative console of the IBM Business Process Manager or WebSphere Enterprise Service Bus to add the dependency libraries.

Before you begin

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

Procedure

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

Results

The dependent libraries are added as shared libraries.

Chapter 10. Troubleshooting and support

Common troubleshooting techniques and self-help information help you identify and solve problems quickly. For information about configuring logging properties and changing the log and trace file names, see “Configuring logging and tracing” on page 251.

Techniques for troubleshooting problems

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

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

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

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

What are the symptoms of the problem?

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

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

Where does the problem occur?

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

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

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

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

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

When does the problem occur?

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

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

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

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

Under which conditions does the problem occur?

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

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

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

Can the problem be reproduced?

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

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

The following questions can help you with reproducing the problem:

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

Log and Trace Analyzer

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

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

The adapter's component ID for filtering and highlighting is a string composed of the characters SAPRA plus the value of the adapter ID property. For example, if the adapter ID property is set to 001, the component ID is SAPRA001.

If you run multiple instances of the same adapter, ensure that the first eight characters of the adapter ID property are unique for each instance so that you can correlate the log and trace information to a particular adapter instance. By making the first seven characters of an adapter ID property unique, the component ID for multiple instances of that adapter is also unique, allowing you to correlate the log and trace information to a particular instance of an adapter. For example, when you set the adapter ID property of two instances of WebSphere Adapter for SAP Software to 001 and 002. The component IDs for those instances, SAPRA001 and SAPRA002, are short enough to remain unique, enabling you to distinguish them as separate adapter instances. However, instances with longer adapter ID properties cannot be distinguished from each other. If you set the adapter ID properties of two instances to Instance01 and Instance02, you will not be able to examine the log and trace information for each adapter instance because the component ID for both instances is truncated to SAPRAInstance.

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

For more information, see the “Adapter ID (AdapterID)” on page 333 property.

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-object level metadata to detect errors, use the following procedure.

Procedure

1. Identify the parameters that define RFC error codes and their possible values.
2. Display the business object in the XML Schema Editor.
3. From the Properties tab, in the Extensions section, select **sapBAPIBusinessObjectTypeMetadata**.
4. Click **Add**, and select **sapasi:ErrorConfiguration**, as shown in the following figure.

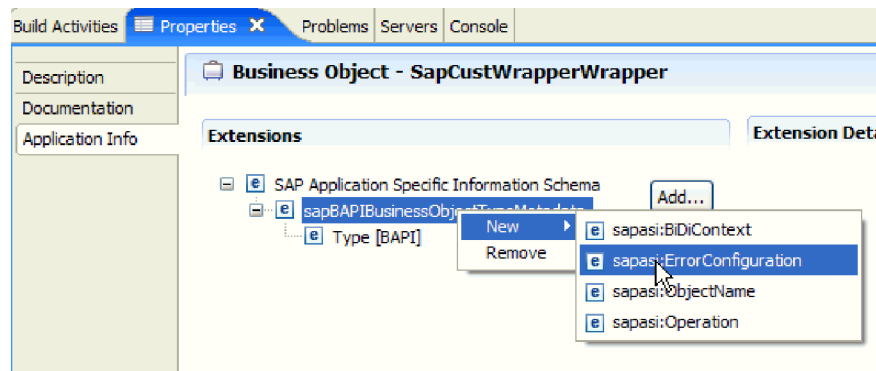


Figure 107. Selecting ErrorConfiguration

5. Add the application-specific information for ErrorParameter, ErrorCode, and ErrorDetail to the business object by right-clicking **sapasi:ErrorConfiguration**, clicking **New**, and selecting **sapasi:ErrorParameter**, **sapasi:ErrorCode**, and **sapasi:ErrorDetail**.

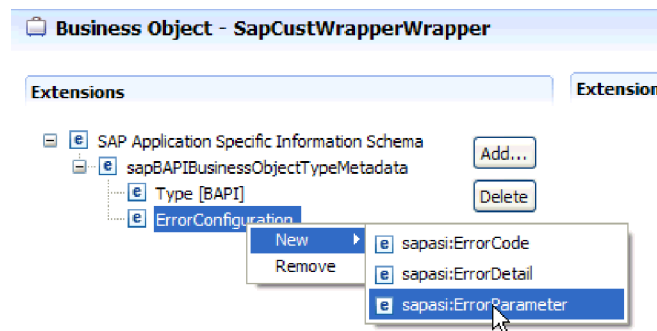


Figure 108. Selecting ErrorCode, ErrorDetail, and ErrorParameter

- 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.

Results

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

Resolving errors during Query interface for SAP Software processing

To avoid data being wrongly truncated with a delimiter, you must change the default function module used by the adapter to retrieve data from SAP tables in the query interface for SAP Software, which processes non-unicode systems for other than English languages.

Before you begin

You must have developer access to SAP system and you should have basic knowledge of advanced business application programming (ABAP) to create a custom module.

About this task

On non-unicode systems, the default function used to retrieve data from SAP tables (RFC_READ_TABLE) might truncate the data wrongly. To avoid the data being wrongly truncated, create a custom function on the SAP server and use this newly created function to retrieve data.

Use the following procedure to create the custom retrieve function and to specify it during configuration:

Procedure

1. Enter an appropriate delimiter in the XSD schema file. In the case of KNA1, the schema generated is SapKna1.xsd.
2. Open this file in a text editor. Ensure that the entered delimiter is unique. By default, the value is set to the pipe symbol (|).
3. In the SAP EIS, Go to SE37 tcode, click **Goto > Function Groups > Create Group**. The Create Function Group window is displayed.
4. In the **Function Group** field, type ZRFC_READ_TABLE as the function group name.
5. In the **Short text** field, type a short description for the function group. For example, type function Group for ZRFC_READ_TABLE as short text for the function group.
6. In the **Person Responsible** field, type the name of the person responsible for creating the function group. Click **Save**.
7. Go to SE80 tcode to activate the function group.
8. Select the package name where you have saved the ZRFC_READ_TABLE.
9. Click the display button and expand **Function Groups**.
10. Select ZRFC_READ_TABLE , select **Activate** using right-click option.

11. Go to SE37 tcode to copy the function group.
12. In the **Function Module** field, enter ZRFC_READ_TABLE.
13. Click **Function Module > Other Functions > Copy**. The Copy Function Module window is displayed.
14. In the **fr. Function module** field, type RFC_READ_TABLE.
15. In the **To Function module** field, type ZRFC_READ_TABLE.
16. In the **Function group** field, type ZRFC_READ_TABLE. Click **Copy**. Click **Change** in the SE37 tcode window.
17. Select **Source code** tab and copy the following code:

```

FUNCTION ZRFC_READ_TABLE.
*-----
*"*Local interface:
*  IMPORTING
*    VALUE(QUERY_TABLE) LIKE DD02L-TABNAME
*    VALUE(DELIMITER) LIKE SONV-FLAG DEFAULT SPACE
*    VALUE(NO_DATA) LIKE SONV-FLAG DEFAULT SPACE
*    VALUE(ROWSKIPS) LIKE SOID-ACCNT DEFAULT 0
*    VALUE(ROWCOUNT) LIKE SOID-ACCNT DEFAULT 0
*  TABLES
*    OPTIONS STRUCTURE RFC_DB_OPT
*    FIELDS STRUCTURE RFC_DB_FLD
*    DATA STRUCTURE TAB512
*    WA1 STRUCTURE TAB512
*  EXCEPTIONS
*    TABLE_NOT_AVAILABLE
*    TABLE_WITHOUT_DATA
*    OPTION_NOT_VALID
*    FIELD_NOT_VALID
*    NOT_AUTHORIZED
*    DATA_BUFFER_EXCEEDED
*-----
"
CALL FUNCTION 'VIEW_AUTHORITY_CHECK'
  EXPORTING
    VIEW_ACTION           = 'S'
    VIEW_NAME             = QUERY_TABLE
  EXCEPTIONS
    NO_AUTHORITY          = 2
    NO_CLIENTINDEPENDENT_AUTHORITY = 2
    NO_LINEDEPENDENT_AUTHORITY = 2
    OTHERS                 = 1.

IF SY-SUBRC = 2.
  RAISE NOT_AUTHORIZED.
ELSEIF SY-SUBRC = 1.
  RAISE TABLE_NOT_AVAILABLE.
ENDIF.

* -----
* find out about the structure of QUERY_TABLE
* -----
DATA BEGIN OF TABLE_STRUCTURE OCCURS 10.
  INCLUDE STRUCTURE DFIES.
DATA END OF TABLE_STRUCTURE.
"DATA TABLE_HEADER LIKE X030L.
DATA TABLE_TYPE TYPE DD02V-TABCLASS.

CALL FUNCTION 'DDIF_FIELDINFO_GET'
  EXPORTING
    TABNAME           = QUERY_TABLE
*   FIELDNAME        = ' '
*   LANGU             = SY-LANGU
*   LFIELDNAME       = ' '

```

```

* ALL_TYPES           = ' '
* GROUP_NAMES        = ' '
IMPORTING
* X030L_WA           =
  DDOBJTYPE          = TABLE_TYPE
* DFIES_WA           =
* LINES_DESCR        =
TABLES
  DFIES_TAB          = TABLE_STRUCTURE
* FIXED_VALUES       =
EXCEPTIONS
  NOT_FOUND          = 1
  INTERNAL_ERROR     = 2
  OTHERS             = 3

IF SY-SUBRC <> 0.
  RAISE TABLE_NOT_AVAILABLE.
ENDIF.
IF TABLE_TYPE = 'INTTAB'.
  RAISE TABLE_WITHOUT_DATA.
ENDIF.

* -----
* isolate first field of DATA as output field
* (i.e. allow for changes to structure DATA!)
* -----
DATA LINE_LENGTH TYPE I.
FIELD-SYMBOLS <D>.
ASSIGN COMPONENT 0 OF STRUCTURE DATA TO <D>.
DESCRIBE FIELD <D> LENGTH LINE_LENGTH in character mode.

* -----
* if FIELDS are not specified, read all available fields
* -----
DATA NUMBER_OF_FIELDS TYPE I.
DESCRIBE TABLE FIELDS LINES NUMBER_OF_FIELDS.
IF NUMBER_OF_FIELDS = 0.
  LOOP AT TABLE_STRUCTURE.
    MOVE TABLE_STRUCTURE-FIELDNAME TO FIELDS-FIELDNAME.
    APPEND FIELDS.
  ENLOOP.
ENDIF.

* -----
* for each field which has to be read, copy structure information
* into tables FIELDS_INT (internal use) and FIELDS (output)
* -----
DATA: BEGIN OF FIELDS_INT OCCURS 10,
      FIELDNAME LIKE TABLE_STRUCTURE-FIELDNAME,
      TYPE      LIKE TABLE_STRUCTURE-INTTYPE,
      DECIMALS  LIKE TABLE_STRUCTURE-DECIMALS,
      LENGTH_SRC LIKE TABLE_STRUCTURE-INTLEN,
      LENGTH_DST LIKE TABLE_STRUCTURE-LENG,
      OFFSET_SRC LIKE TABLE_STRUCTURE-OFFSET,
      OFFSET_DST LIKE TABLE_STRUCTURE-OFFSET,
    END OF FIELDS_INT,
    LINE_CURSOR TYPE I.

LINE_CURSOR = 0.
* for each field which has to be read ...
LOOP AT FIELDS.

  READ TABLE TABLE_STRUCTURE WITH KEY FIELDNAME = FIELDS-FIELDNAME.
  IF SY-SUBRC NE 0.
    RAISE FIELD_NOT_VALID.
  ENDIF.

* compute the place for field contents in DATA rows:

```

```

* if not first field in row, allow space for delimiter
IF LINE_CURSOR <> 0.
  IF NO_DATA EQ SPACE AND DELIMITER NE SPACE.
    LINE_CURSOR = LINE_CURSOR + 1. "SARMA
    MOVE DELIMITER TO DATA+LINE_CURSOR .
  ENDIF.
  LINE_CURSOR = LINE_CURSOR + STRLEN( DELIMITER ).
ENDIF.

* ... copy structure information into tables FIELDS_INT
* (which is used internally during SELECT) ...
FIELDS_INT-FIELDNAME = TABLE_STRUCTURE-FIELDNAME.
FIELDS_INT-LENGTH_SRC = TABLE_STRUCTURE-INTLEN .
FIELDS_INT-LENGTH_DST = TABLE_STRUCTURE-LENG .
FIELDS_INT-OFFSET_SRC = TABLE_STRUCTURE-OFFSET .
FIELDS_INT-OFFSET_DST = LINE_CURSOR .
FIELDS_INT-TYPE = TABLE_STRUCTURE-INTTYPE.
FIELDS_INT-DECIMALS = TABLE_STRUCTURE-DECIMALS.
* compute the place for contents of next field in DATA rows
LINE_CURSOR = LINE_CURSOR + TABLE_STRUCTURE-LENG.
IF LINE_CURSOR > LINE_LENGTH AND NO_DATA EQ SPACE.
  RAISE DATA_BUFFER_EXCEEDED.
ENDIF.
APPEND FIELDS_INT.

* ... and into table FIELDS (which is output to the caller)
FIELDS-FIELDTEXT = TABLE_STRUCTURE-FIELDTEXT.
FIELDS-TYPE = TABLE_STRUCTURE-INTTYPE.
FIELDS-LENGTH = FIELDS_INT-LENGTH_DST + 2 .
FIELDS-OFFSET = FIELDS_INT-OFFSET_DST + 2.
MODIFY FIELDS.

ENDLOOP.
* end of loop at FIELDS

* -----
* read data from the database and copy relevant portions into DATA
* -----
* output data only if NO_DATA equals space (otherwise the structure
* information in FIELDS is the only result of the module)
IF NO_DATA EQ SPACE.

DATA: BEGIN OF WORK, BUFFER(30000), END OF WORK.
FIELD-SYMBOLS: <WA> TYPE ANY, <COMP> TYPE ANY.
ASSIGN WORK TO <WA>> CASTING TYPE (QUERY_TABLE).
IF ROWCOUNT > 0.
  ROWCOUNT = ROWCOUNT + ROWSKIPS.
ENDIF.
SELECT * FROM (QUERY_TABLE) INTO <WA>WHERE (OPTIONS).

  IF SY-DBCNT GT ROWSKIPS.

* copy all relevant fields into DATA (output) table
  LOOP AT FIELDS_INT.
    IF FIELDS_INT-TYPE = 'P'.
      ASSIGN COMPONENT FIELDS_INT-FIELDNAME
      OF STRUCTURE <WA> TO <COMP>
      TYPE FIELDS_INT-TYPE
      DECIMALS FIELDS_INT-DECIMALS.
    ELSE.
      ASSIGN COMPONENT FIELDS_INT-FIELDNAME
      OF STRUCTURE <WA> TO <COMP>
      TYPE FIELDS_INT-TYPE.
    ENDIF.
    MOVE <COMP> TO
    <D>+FIELDS_INT-OFFSET_DST(FIELDS_INT-LENGTH_DST).
  ENDLOOP.

```



```

*   end of loop at FIELDS_INT
  APPEND DATA.
  IF ROWCOUNT > 0 AND SY-DBCNT GE ROWCOUNT. EXIT. ENDIF.

  ENDIF.

ENDSELECT.

ENDIF.

ENDFUNCTION.

```

18. Go to SE37 tcode and select ZRFC_READ_TABLE. Click **Change**.
19. Click **Attributes** tab, select **Remote-Enabled Module** in the Processing Type pane.
20. Click **Save**.
21. Click **Activate**.
22. To configure the function you just created for Query interface for SAP Software in the external service wizard, specify the name of the custom function you created in step 1 on page 283, in the **Custom retrieve function name** field in the Configure Composite Properties window.

Results

The adapter retrieves an error-free data from the SAP tables during query interface.

SAP dependencies when using the WebSphere Adapter for SAP Software with the Advanced Event Processing (AEP) interface

The AEP transports require certain standard SAP function modules to import and work correctly. These standard function modules are required by the WebSphere BI Station Tool (SAPGUI transaction /CWLD/HOME_AEP), which is used to monitor and modify events for processing by the AEP interface

About this task

When using an SAP instance other than SAP ERP 4.7, ECC 5.0/ECC 6.0 (like SRM which differ significantly from ECC/ERP systems in terms of available standard SAP Function Modules), please ensure that you have the below listed SAP Standard Function Modules in your SAP instance(s) to run the adapter using the AEP (Advanced Event Processing) interface.

Procedure

1. AUTHORITY_CHECK_DATASET
2. BDC_CLOSE_GROUP
3. BDC_INSERT
4. BDC_OPEN_GROUP
5. BDC_RECORD_TRANSACTION
6. BP_FIND_JOBS_WITH_PROGRAM
7. BP_JOB_CREATE
8. BP_JOBLIST_PROCESSOR
9. DDIF_DTEL_GET
10. DDIF_FIELDINFO_GET
11. DDIF_NAMETAB_GET DYNP_VALUES_READ

12. DYNP_VALUES_UPDATE
13. EDI_DOCUMENT_STATUS_DISPLAY
14. ENQUEUE_READ
15. F4_USER
16. FUNCTION_DELETE
17. FUNCTION_EXISTS
18. FUNCTION_IMPORT_DOKU
19. HELP_VALUES_GET_WITH_TABLE
20. IDOCTYPE_READ
21. IMPORT_DYNPRO
22. NAME_OF_CURRENT_TRACE_FILE
23. NAMETAB_GET
24. NUMBER_GET_NEXT
25. POPUP_TO_CONFIRM
26. POPUP_TO_CONFIRM_LOSS_OF_DATA
27. POPUP_TO_CONFIRM_STEP
28. POPUP_TO_CONFIRM_WITH_MESSAGE
29. POPUP_TO_DECIDE
30. POPUP_TO_DECIDE_WITH_MESSAGE
31. POPUP_TO_DISPLAY_TEXT
32. POPUP_TO_GET_VALUE
33. POPUP_TO_INFORM
34. POPUP_WITH_3_BUTTONS_TO_CHOOSE
35. READ_TEXT
36. RS_CREATE_VARIANT
37. RS_FUNCTIONMODULE_INSERT
38. RS_TOOL_ACCESS
39. RS_TREE_CONSTRUCT
40. RS_TREE_LIST_DISPLAY
41. RS_TREE_SET_NODE
42. RS_VARIANT_ADD
43. RS_VARIANT_CATALOG
44. RS_VARIANT_CHECK_TSTC
45. RS_VARIANT_DELETE
46. SAPGUI_PROGRESS_INDICATOR
47. SEGMENTDEFINITION_READ
48. SQLT_GEN_TRACE_RECORDS (or) SQLT_GEN_TRACE_RECORDS_NEW
49. SQLT_GET_TRACE_RECORDS
50. SWO_OBJTYPE_EXIST
51. SWO_QUERY_API_METHODS
52. SWO_QUERY_BASEDATA
53. SWO_QUERY_KEYFIELDS
54. SWO_TYPE_INFO_GET
55. TABLE_CUSTOMIZING_MAINTENANCE
56. WS_FILENAME_GET

57. WS_QUERY

Results

Importing the transports on SAP systems will not fail if all of the above listed standard SAP Function Modules are present.

Resolving memory-related issues

You can increase the IBM Business Process Manager 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 large IDoc is sent from the SAP server to IBM Business Process Manager 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.

Supported codepages for WebSphere Adapter for SAP Software

Since the adapter is based on Java, it automatically converts language characters to Unicode when sending data to the broker. It also automatically converts Unicode characters to the code page of the SAP system (if set to non-Unicode) and the SAP JCo support when sending data to the SAP system (if set to non-Unicode).

The adapter works with the SAP JCo API, which solely supports 8000 Shift-JIS. Refer to the following SAP notes for a list of supported code pages for data conversion.

- SAP note 794411
- SAP note 73606

First-failure data capture (FFDC) support

The adapter supports first-failure data capture (FFDC), which provides persistent records of failures and significant software incidents that occur during run time in IBM Business Process Manager or WebSphere Enterprise Service Bus.

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

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

For more information about first-failure data capture (FFDC), see the IBM Business Process Manager or WebSphere Enterprise Service Bus documentation.

Avoiding stale connection problems in the SAP adapter

Connection related problems can be resolved in WebSphere Adapter for SAP Software using two new properties, **connectionRetryLimit** and **connectionRetryInterval** defined in the Managed connection factory of the SAP adapter.

Before you begin

The two properties are used to provide two features during outbound communication of the adapter and are optional.

About this task

In the service generation and deployment configuration window explained in the “Setting deployment properties and generating the service” on page 117 topic for the appropriate interface,

Procedure

1. Click on the **Advanced -> Additional connection configuration**
2. Set the **Maximum number of retries in case of system connection failure** to the appropriate positive integer
 - if the property value is set to 0, then adapter does not perform any EIS connection validation and executes the outbound operation. If the EIS connection is invalid, the outbound operation fails. Though the subsequent requests get executed successfully provided the SAP system is functional, the current request fails.
 - if the property value is set to greater than 0, then during each request the adapter validates that the EIS connection is active/alive .
 - If connection is valid then operation is completed. if connection is invalid, the adapter invalidates the current managed connection so that a new managed connection is created (new physical connection).
 - If the connection is created successfully, the outbound operation is completed otherwise a `ResourceException` error is thrown.
3. Set the **Time interval between retries if connection fails (milliseconds)** field with the appropriate integer to indicate time in milliseconds in between retries. This property is enabled only when the **connectionRetryLimit** property has a value greater than 0.

Results

The two new features take care of connections which are timed out or become stale after the EIS restarts. Even though this option solves most of connection related problems, the adapter is not guaranteed to work 100% error free for connection problems.

Related reference:

“Managed connection factory properties” on page 334

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

Resolving selector exception error

Selector exception errors can be resolved in WebSphere Adapter for SAP Software by identifying the IDoc which has not been configured. Using the external service wizard, configure the IDoc to resolve the error.

About this task

For Ale Inbound and BAPI inbound

Procedure

1. If you configure the adapter to work for a specified IDoc/ BAPI (eg., ALEREQ01.BAPI BAPI_CUSTOMER_GETLIST, etc.)
2. A different IDoc is sent from the SAP system (eg., ORDERS05), a function selector exception is logged which is displayed as follows:
 - For ALE inbound:
Business object definition for 'SapOrders05' in namespace 'http://www.ibm.com/xmlns/prod/websphere/j2ca/sap/saporders05' not found. The adapter has not been configured for the IDoc Type ORDERS05
 - For BAPI inbound:
Business object definition for 'SapBapiCustomerGetlistWrapper' in namespace 'http://www.ibm.com/xmlns/prod/websphere/j2ca/sap/sapbapicustomergetlistwrapper' not found. The adapter has not been configured for the BAPI BAPI_CUSTOMER_GETLIST

Results

To resolve the function selector exception, run the EMD again and select the particular IDoc. Refer to “Configuring the module for inbound processing” on page 178 to configure the adapter for the selected IDoc.

Related tasks:

“Configuring a module for ALE inbound processing” on page 189

To configure a module to use the adapter for ALE inbound processing, you use the external service wizard in IBM Integration Designer to find an IDoc or set of IDocs, configure the business objects that are generated, and create a deployable module. If you are going to set up an event recovery table to persist inbound events (to ensure once-only delivery of events), you must also set up a data source.

“Configuring a module for BAPI inbound processing” on page 179

To configure a module to use the adapter for BAPI inbound processing, you use the external service wizard in IBM Integration Designer to find RFC-enabled functions. You then configure the business objects that are generated and create a deployable module.

Resolving a service 'sapxxnn' unknown error

While deploying an inbound module to test environment, if you get an error message stating that the "service 'sapxxnn' unknown" (where xx is two letters, and nn is two numbers), you can resolve this by adding a missing entry into the services file.

This error is caused when the SAP gateway protocol 'sapxxnn' is not existing in the services file. In UNIX, it is located in **etc/services** and in Windows, it is located in **\WINDOWS\system32\drivers\etc\services**. To verify and resolve this error:

- Open the services file in a text editor (notepad or wordpad) and search for a valid 'sapxxnn' entry.
- If it is not in the services file, add the line 'sapxxnn' (minus the quotes) at the bottom of the file. There must be a carriage return at the end of this line if it is the last line in the file.
- You also need to add **port/protocol** in addition to the 'sapxxnn' entry. For example, **sapgw00 3600/tcp**

Resolving SAP JCo environment setup errors

When deploying a module (Inbound or Outbound) using any interface with the JCo3 jar, the module deployment may fail with the `NoClassDefFoundError` or a `ClassNotFoundException` in the run time environment. The error can be resolved by making sure the `CWYAP_SAPAdapterExt.jar`, which is a mandatory JAR, is available in the IBM Business Process Manager or WebSphere Enterprise Service Bus run time class path.

When using JCo, the `CWYAP_SAPAdapterExt.jar` is a mandatory JAR and must be available in the IBM Business Process Manager or WebSphere Enterprise Service Bus run time classpath. If the IBM Business Process Manager or WebSphere Enterprise Service Bus run time cannot find the JAR, you will see a stacktrace in the trace/log as shown below:

```
Caused by: java.lang.ClassNotFoundException: com.ibm.j2ca.sap.ext.JCo3DestinationDataProvider
    at java.net.URLClassLoader.findClass(URLClassLoader.java:496)
    at com.ibm.ws.bootstrap.ExtClassLoader.findClass(ExtClassLoader.java:132)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:631)
    at com.ibm.ws.bootstrap.ExtClassLoader.loadClass(ExtClassLoader.java:87)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:597)
```

To resolve this error, perform the following steps:

1. Stop the IBM Business Process Manager or WebSphere Enterprise Service Bus instance

2. Copy the CWYAP_SAPAdapterExt.jar JAR into the \$WPS_Root\lib\ext folder
3. Restart IBM Business Process Manager or WebSphere Enterprise Service Bus and deploy the module again.

Adapter returns version conflict exception message

Adapter returns version conflict exception message

Problem

When you install multiple adapters with different versions of CWYBS_AdapterFoundation.jar, and if a lower version of the CWYBS_AdapterFoundation.jar is loaded during run time, the adapter returns the ResourceAdapterInternalException error message, due to a version conflict. For example, when you install Oracle E-Business Suite adapter version 7.0.0.3 and WebSphere Adapter for SAP Software version 7.5.0.3, the following error message is displayed "The version of CWYBS_AdapterFoundation.jar is not compatible with IBM WebSphere Adapter for SAP Software" as IBM WebSphere Adapter for SAP Software loads file:/C:/IBM/WebSphere/ProcServer7/profiles/ProcSrv01/installedConnectors/CWYOE_OracleEBS.rar/CWYBS_AdapterFoundation.jar with version 7.0.0.3. However, the base level of this jar required is version 7.5.0.3.

Solution

Migrate all adapters to the same version level.

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

Unreleased IDoc segments cause exceptions

When running the external service wizard or at runtime, unreleased segments in IDocs cause the following exception error messages:

- During object discovery using the external service wizard, you might get the exception "The segment could not be read for the release <release number>".
- At run time you might encounter an error message: "Caused by: commonj.connector.runtime.DataBindingException: Failed in population return object" error. This occurs even if all the business object definitions were generated successfully for the IDoc when you ran the external service wizard.

For any IDoc segment there may be multiple segment definitions at different release levels. Some of these segments may not have been released. If an IDoc contains segments that are unreleased for the selected IDOC release version, then the external service wizard may fail. This might cause a problem for the adapter at run time as the adapter will not be able to find the business object definitions for the unreleased children segments. For example, if the IDoc MATMAS05 has an unreleased segment E2MARAM005, the external wizard will create an E2MRAM004 (released segment) child business object definition for MATMAS05 business object instead of E2MARAM005.

To resolve this error, you can specify the release version for the IDoc when using the external service wizard to discover IDocs. The default value is the version of the SAP system you are running the wizard against. If the IDoc you selected has unreleased segments, you should set the field "IDoc release version" on "Configure Properties for <IDoc name> page of the wizard to empty (zero length string).

Invalid meta data exception

Using Adapter for SAP Software, you can either configure an outbound or an inbound module for the adapter connections.

If you try to configure the inbound and immediately configure an outbound for the same module, the outbound artifacts overwrite the inbound artifacts, resulting in both the artifacts not working. This is a tooling limitation. The following error message is displayed:

InvalidMetadataException occurs in SAP BAPI Outbound Invocation

Disabling end point applications of the passive adapter

Problem

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

Cause

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

Solution

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

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

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

Result

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

Frequently Asked Questions

This section provides you with answers to common questions about WebSphere Adapter for SAP Software.

- “How to use the Reset JCO Client?”
- “What are the mandatory fields while deploying an adapter at the node level?”

How to use the Reset JCO Client?

The reset property (Reset the JCO client after closing the connection handle) is used to obtain the latest changes of the ABAP definition (BAPI) dynamically. The changes can be any logical change in the ABAP code or a parameter change in the function module. This property is applicable only for BAPI Outbound modules.

Following are the advantages of using the Reset JCO client feature:

- Obtain the latest changes to the ABAP definition on the SAP system by discarding the cached contents in JCO repository.
- Get the changes of the BAPI definition dynamically, without restarting the adapter.

What are the mandatory fields while deploying an adapter at the node level?

When you deploy the adapter at the node level and manually create the activation specification and the managed connection factory properties ensure that you specify the mandatory fields as indicated below.

For ALE inbound, the following example indicates the mandatory properties that you need to specify in the activation specification properties. You can locate the following properties in the **.export** file.

```
<properties>
<BONamespace>http://www.ibm.com/xmlns/prod/websphere/j2ca/sap</BONamespace>
<applicationServerHost>9.184.167.115applicationServerHost>9.184.167.115>
<assuredOnceDelivery>>falseassuredOnceDelivery>>false>
<client>100client>100>
<gatewayHost>9.184.167.115gatewayHost>9.184.167.115>
<gatewayService>sapgw10gatewayService>sapgw10>
<language>ENlanguage>EN>
<password>saperp03password>saperp03>
<rfcProgramID>TESTRFCSEVERrfcProgramID>TESTRFCSEVER>
<systemNumber>10systemNumber>10>
<userName>TESTuserName>TEST>
</properties>
```

For BAPI inbound, the following example indicates the mandatory properties that you need to specify in the activation specification properties. You can locate the following properties in the **.export** file.

```
<properties>
<BONamespace>http://www.ibm.com/xmlns/prod/websphere/j2ca/sap</BONamespace>
<applicationServerHost>9.184.167.115applicationServerHost>9.184.167.115>
<assuredOnceDelivery>>falseassuredOnceDelivery>>false>
<client>100client>100>
<gatewayHost>9.184.167.115gatewayHost>9.184.167.115>
<gatewayService>sapgw10gatewayService>sapgw10>
<language>EN (English)language>EN (English)>
<password>saperp02password>saperp02>
```

```
<rfcProgramID>TESTRFCSEVERrfcProgramID>TESTRFCSEVER<
<systemNumber>10systemNumber>10>
<userName>TESTuserName>TEST>
</properties>
```

For BAPI outbound, the following example indicates the mandatory properties that you need to specify in the activation specification properties. You can locate the following properties in the **.import** file.

```
<resourceAdapter>
<connection type="com.ibm.j2ca.sap.SAPManagedConnectionFactory" interactionType="com.ibm.j2ca.sap.SAPManagedConnectionFactory" >
  <properties>
    <adapterID>001<,adapterID>
    <applicationServerHost>9.184.167.115applicationServerHost>9.184.167.115>
    <language>ENlanguage>EN>
    <password>saperp02password>saperp02>
    <systemNumber>10M/systemNumber>
    <userName>TESTuserName>TEST>
  </properties>
</connection>
```

Support

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

Searching knowledge bases (Web search)

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

About this task

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

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

- Search for content by using the IBM® Support Assistant (ISA).
ISA is a no-charge software serviceability workbench that helps you answer questions and resolve problems with IBM software products. You can find instructions for downloading and installing ISA on the ISA website.
- Find the content that you need by using the IBM Support Portal.
The IBM Support Portal is a unified, centralized view of all technical support tools and information for all IBM systems, software, and services. The IBM Support Portal lets you access the IBM electronic support portfolio from one place. You can tailor the pages to focus on the information and resources that you need for problem prevention and faster problem resolution. Familiarize yourself with the IBM Support Portal by viewing the demo videos (https://www.ibm.com/blogs/SPNA/entry/the_ibm_support_portal_videos) about this tool. These videos introduce you to the IBM Support Portal, explore troubleshooting and other resources, and demonstrate how you can tailor the page by moving, adding, and deleting portlets.

- Search for content by using the IBM masthead search. You can use the IBM masthead search by typing your search string into the Search field at the top of any ibm.com® page.
- Search for content by using any external search engine, such as Google, Yahoo, or Bing. If you use an external search engine, your results are more likely to include information that is outside the ibm.com domain. However, sometimes you can find useful problem-solving information about IBM products in newsgroups, forums, and blogs that are not on ibm.com.

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

Getting Fixes

A product fix might be available to resolve your problem.

About this task

To get product fixes, perform the following steps.

Procedure

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

Individual fixes are published as often as necessary to resolve defects in IBM Business Process Manager or WebSphere Enterprise Service Bus IBM WebSphere Adapter for SAP Software. In addition, two kinds of cumulative collections of fixes, called fix packs and refresh packs, are published periodically for IBM WebSphere Adapter for SAP Software, in order to bring users up to the latest maintenance level. You should install these update packages as early as possible in order to prevent problems.

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

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

Self-help resources

Use the resources of IBM software support to get the most current support information, obtain technical documentation, download support tools and fixes,

and avoid problems with WebSphere Adapters. The self-help resources also help you diagnose problems with the adapter and provide information about how to contact IBM software support.

Support website

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

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

Recommended fixes

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

Technotes

Technotes provide the most current documentation about WebSphere Adapter for SAP Software, including the following topics:

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

For a list of technotes for WebSphere Adapter for SAP Software, see <http://www-01.ibm.com/support/docview.wss?uid=swg27024045>.

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

Chapter 11. Reference information

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

Business object information

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

Application-specific information

Application-specific information (ASI) is metadata that specifies adapter-dependent information about how to process business objects for the adapter for SAP Software. When the external service wizard generates a business object, it automatically generates a business object definition, which is saved as an XSD (XML Schema Definition) file. The business object definition contains the application-specific information for that business object. If you want to change the generated ASI, you can modify the metadata values either from the Properties tab in the Business Integration perspective of IBM Integration Designer or by using the business object editor.

BAPI business object application-specific information

BAPI application-specific information (ASI) is metadata that specifies adapter-dependent information about how to process BAPI business objects for the WebSphere Adapter for SAP Software.

Business object-level metadata for BAPI

WebSphere Adapter for SAP Software uses application-specific information (ASI) to create queries for Create, Retrieve, Update, and Delete operations. ASI for BAPI is generated by the external service wizard at the following levels: the business-object level, the operation-level and the property-level.

The sections that follow describe the metadata elements for each level.

Business object-level metadata defines the top-level wrapper of the business object.

The following table lists and describes the business object-level metadata elements for a BAPI business object.

Table 25. Metadata elements: Wrapper of a BAPI business object

Metadata element	Description
Type	The business object type. For a simple BAPI, the value is BAPI. For a BAPI work unit business object, this value is BAPITXN. For a BAPI result set, this value is BAPIRS.

Table 25. Metadata elements: Wrapper of a BAPI business object (continued)

Metadata element	Description
Operation	<p>The valid operations include Create, Update, Delete, and Retrieve. The specified operation metadata is defined in the sapBAPIOperationTypeMetadata tag and contains the following:</p> <ul style="list-style-type: none"> • MethodName: Name of the BAPI associated with the operation. • Name: Name of the operation. <p>Note: This is applicable if the Generate BAPIs within Wrappers check box is selected.</p> <p>If you do not select the Generate BAPIs within Wrappers check box, top-level business objects are automatically generated for each BAPI selected. The adapter internally assigns the Execute operation for each top-level business object generated.</p>

The following illustration is an example of BAPI business object metadata:

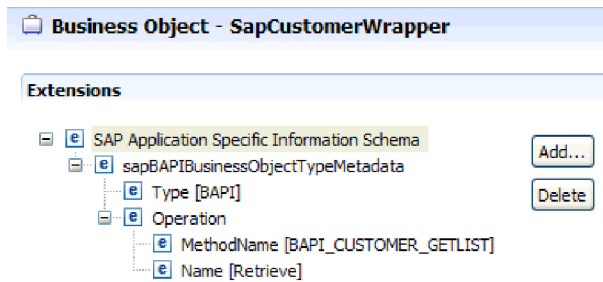


Figure 109. Business-object metadata for SapCustomerWrapper

The following illustration is an example of BAPI work unit business object metadata:

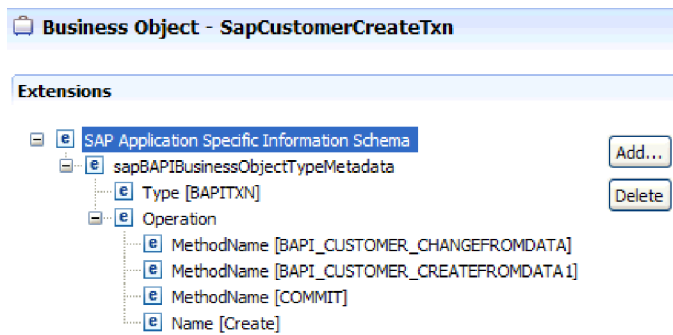


Figure 110. Business-object metadata for SapCustomerCreateTxn

The following illustration is an example of BAPI result set business object metadata:

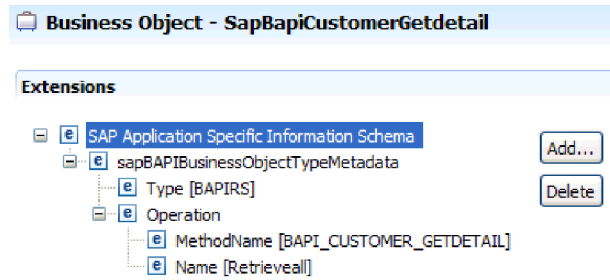


Figure 111. Business-object metadata for SapBapiCustomerGetdetail

Property-level metadata for BAPI business objects

Property-level metadata represents child objects or an array of child objects.

The following table describes the metadata elements of a complex property (child) or structure or table property (an array of child objects).

Table 26. Property-level metadata elements: BAPI business object

Metadata element	Description
FieldName	The BAPI field name as represented in SAP.
FieldType	The type of the property as it exists in SAP.
PrimaryKey	An indication about whether this property is a primary key.
ParameterType	The direction of the mapping. <ul style="list-style-type: none"> • 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).
MaxLength	The length of the field.
ForeignKey	The foreign-key relationship. This element applies only to BAPI result sets.
DecimalPlaces	For fields with a FieldType of Decimal, the value of the precision level. This value is extracted from metadata on the SAP server.
Description	The description of the field. This value is extracted from metadata on the SAP server.

The following illustration is an example of property-level metadata for a BAPI business object:

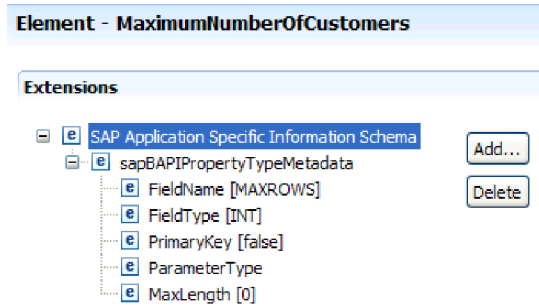


Figure 112. Property-level metadata for MaximumNumberOfCustomers

The following illustration is an example of property-level metadata for a BAPI result set business object:

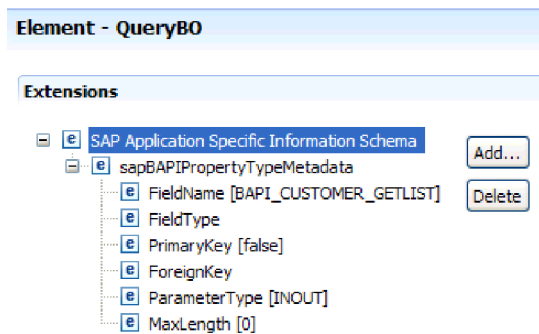


Figure 113. Property-level metadata for QueryBO

Operation-level metadata for BAPI business objects

Operation-level metadata specifies the method name of the BAPI in the SAP system. This name is used by the adapter to determine the action to take on the BAPI.

The following table describes the operation-level metadata elements of a BAPI business object.

Table 27. Operation-level metadata elements: BAPI business object

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.

Operation-level metadata for a BAPI, a BAPI work unit, and a BAPI result set are shown in the figures in the “Business object-level metadata for BAPI” on page 299. Notice that the BAPI work unit has three MethodName values listed—two for the BAPIs in the transaction and one for the COMMIT. The operations are listed in the sequence in which they are called.

ALE business object application-specific information

ALE application-specific information (ASI) is metadata that specifies adapter-dependent information about how to process ALE business objects for the adapter for SAP Software.

The type of metadata that is generated depends on whether you are using the ALE interface or the ALE pass-through IDoc interface:

- **ALE interface**

The WebSphere Adapter for SAP Software uses application-specific information (ASI) to create queries for Create, Retrieve, Update, and Delete operations.

ASI for objects generated with the ALE interface is available at the following levels:

- The IDoc business-object level (for individual IDocs)
- The IDoc wrapper business-object level (for IDoc packets)
- The operation level for individual IDoc business objects
- The property level

For ALE inbound processing, the adapter for SAP Software uses ASI to determine which of the supported operations (Create, Retrieve, Update, or Delete) to run on the endpoint.

Note: There is no metadata at the IDoc Data Record or IDoc Control Record child business object-level.

- **ALE pass-through IDoc interface**

ASI for objects generated with the ALE pass-through IDoc interface is available at the following levels:

- The IDoc business-object level
- The property level

The sections that follow describe the metadata elements for each level.

Business-object-level metadata for ALE

- **ALE interface**

Business object-level metadata for ALE interface 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 28. Business object-level metadata elements: ALE business object

Metadata element	Description
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. If you select the corresponding property (check box) in the external service wizard, make sure you set this property to true.
Type	The business object type. Possible values are IDOC or UNPARSEDIDOC.

Table 28. Business object-level metadata elements: ALE business object (continued)

Metadata element	Description
Operation	<p>Each <i>outbound</i> operation contains the following parameters:</p> <p>Name Name of the operation: For outbound processing, it is always Execute.</p> <p>Each <i>inbound</i> operation contains the following parameters:</p> <p>Name Name of the operation: Create, Update, or Delete.</p> <p>MsgType The message type configured for the IDoc.</p> <p>MsgCode The message code configured for the IDoc.</p> <p>MsgFunction The message function configured for the IDoc.</p>

The following illustration is an example of ALE business object metadata for an outbound operation:

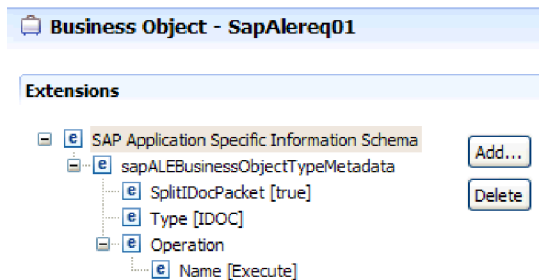


Figure 114. Business-object metadata for SapAlereq01

- **ALE pass-through IDoc interface**

Business object-level metadata for ALE pass-through IDoc interface business objects defines the top-level wrapper of an IDoc.

The following tables describe the business-object metadata elements of an ALE pass-through IDoc interface business object.

Table 29. Business object-level metadata elements: Generic IDoc business object

Metadata element	Description
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. If you select the corresponding property (check box) in the external service wizard, make sure you set this property to true.
Type	The business object type. For a generic IDoc, this value is PASSTHROUGHIDOC.
Delimiter	Use a delimiter to split an IDoc's control record (fixed length) or IDoc segments (if less than specified length). The possible values are any strings without escape characters, \n or \\r\n. Input a delimiter in the wizard at the following location "Selecting business objects and services for ALE pass-through IDoc outbound processing" on page 153

Property-level metadata for ALE business objects

Property-level metadata either represents child objects or an array of child objects.

The following table describes the property-level metadata elements of an ALE business object or an ALE pass-through IDoc interface business object.

Table 30. Property-level metadata elements: ALE business object

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	The xpath to the primary key on the control or data record business object property, which you set using the external service wizard.
MaxLength	The length of the field.

The following illustration is an example of ALE property-level metadata for the qRFCQueueName property:

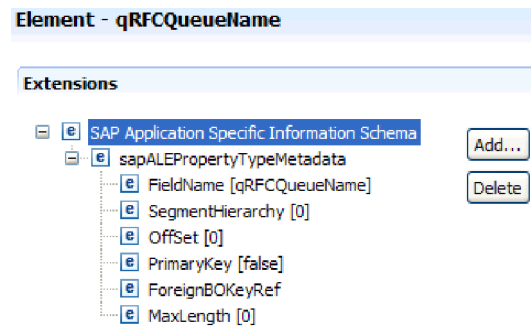


Figure 115. Property-level metadata for qRFCQueueName

Operation-level metadata for ALE business objects

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 operation-level metadata elements of an ALE business object.

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 elements: ALE business object

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).

Table 31. Operation-level metadata elements: ALE business object (continued)

Metadata element	Description
MsgFunction	The message function configured for the IDoc (for inbound objects only).

Query interface for SAP Software business objects application-specific information

Query interface for SAP Software application-specific information (ASI) is metadata that specifies adapter-dependent information about how to process Query interface for SAP Software business objects for WebSphere Adapter for SAP Software.

Business object-level metadata for Query interface for SAP Software

The adapter for SAP Software uses application-specific information (ASI) to create queries for Create, Retrieve, Update, and Delete operations. ASI for Query interface for SAP Software is generated by the external service wizard at the following levels: the table and query business-object level and at the property-level.

The sections that follow describe the metadata elements for each level.

The following table describes the business object-level metadata elements of a Query interface for SAP Software table business object.

Table 32. Business object-level metadata elements: Query interface for SAP Software table business object

Metadata element	Description
TableName	The name of the table that this business object represents.
Type	The interface type the business object is supporting, which for the Query interface for SAP Software is QISS.

The following illustration is an example of Query interface for SAP Software business object-level metadata:

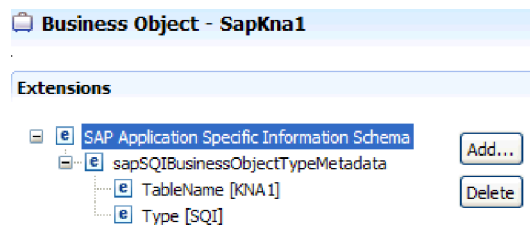


Figure 116. Business-object metadata for SapKna1

Property-level metadata for Query interface for SAP Software business objects

Property-level metadata represents child objects or an array of child objects.

The following table describes the property-level metadata elements of a Query interface business object.

Table 33. Property-level metadata elements: Query interface for SAP Software business object

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	The foreign key relationship (if this property is a key), which is the reference to the parent table key parameter. For an example of how the foreign key relationship is established using the external service wizard, see the illustration of the External Service wizard following this table.
MaxLength	The length of the field.

The following screen capture illustrates where the foreign key relation is formed using the external service wizard:

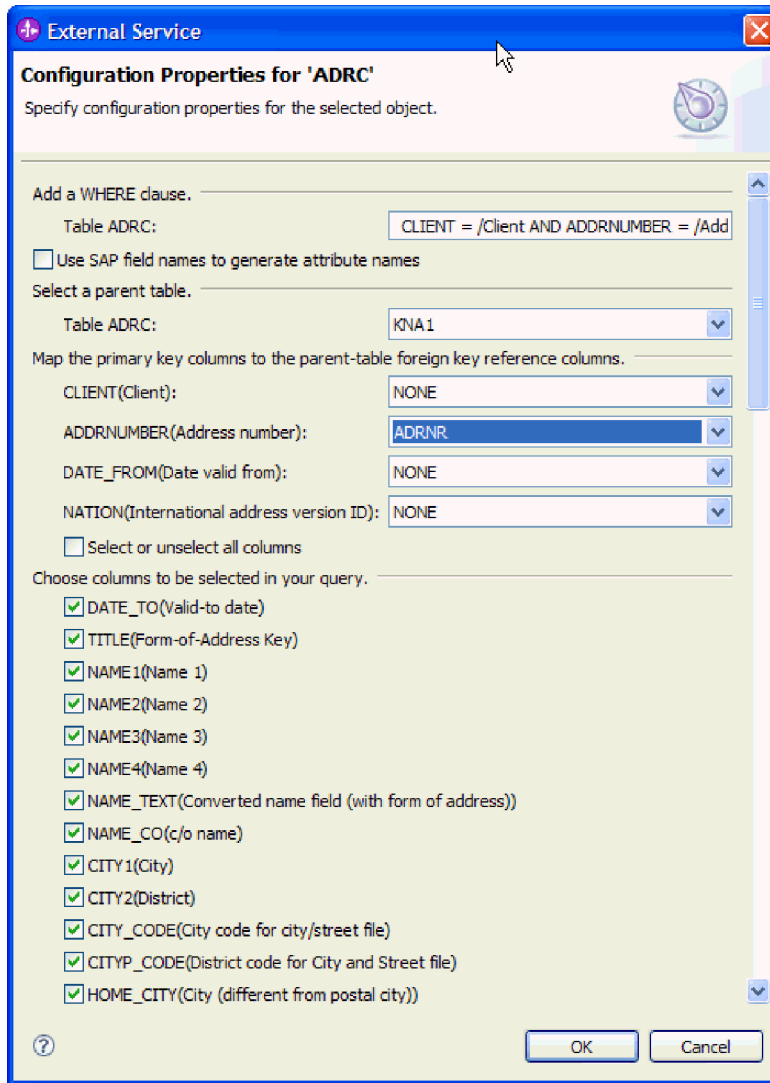


Figure 117. Mapping the primary key columns to the parent-table foreign key reference columns

The following illustration is an example of Query interface for SAP Software property-level metadata:

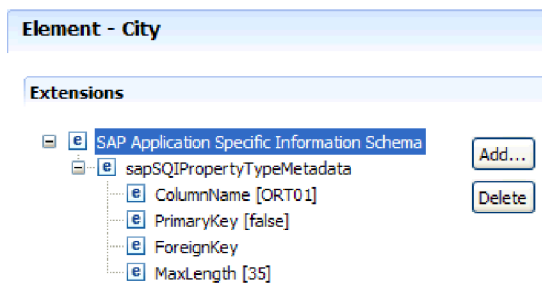


Figure 118. Property-level metadata for City

Advanced event processing business object application-specific information

Advanced event processing application-specific information (ASI) is metadata that specifies adapter-dependent information about how to process business objects for the adapter for SAP Software.

The adapter for SAP Software uses application-specific information (ASI) to create queries for Create, Retrieve, Update, and Delete operations. ASI for Advanced event processing business objects is generated by the external service wizard at the following levels: the IDoc business-object level (for individual IDocs), the Operation-level for individual IDoc business objects, and the property level.

Note: There is no metadata at the IDoc Data Record or IDoc Control Record child business object-level.

The sections that follow describe the metadata elements for each level.

Business-object-level metadata for Advanced event processing

Business object-level metadata for Advanced event processing business objects defines the top-level wrapper of an IDoc.

The following table describes the business object-level metadata elements of an Advanced event processing business object.

Table 34. Business object-level metadata elements: Advanced event processing

Metadata element	Description
Type	The business object type. The business object type will always be AEP.
Operation	<p>Each <i>outbound</i> operation contains the following parameters:</p> <p>Name Name of the operation (Create, Update, Delete or Retrieve)</p> <p>MethodName The name of the Advanced event processing handler for the operation.</p> <p>RouterName The name of the router.</p> <p>Each <i>inbound</i> operation contains the following parameters:</p> <p>Name Name of the operation (Create, Update, or Delete).</p> <p>MethodName The name of the Advanced event processing handler for the operation.</p> <p>RouterName The name of the router.</p>

For AEP inbound processing, **MethodName** should represent a method that retrieves data from the SAP system. The data retrieved might correspond to a Create, Update or Delete operation. For example, when you *create* a customer in the SAP system, this operation generates an event in the AEP event table (with CustomerID as key). The AEP inbound processing retrieves the data for the customer that was created and sends it to endpoint. A similar processing sequence would occur for customer update or customer delete operations in the SAP system.

The following illustration is an example of Advanced event processing business object metadata for an outbound operation:

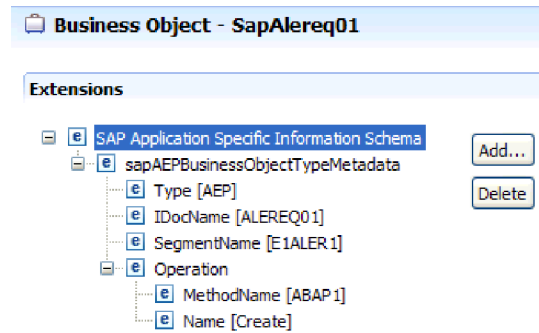


Figure 119. Business-object metadata for SapAlereq01

Property-level metadata for Advanced event processing business objects

Property-level metadata can represent either child objects or an array of child objects.

The following table describes the property-level metadata elements of an Advanced event processing business object.

Table 35. Property-level metadata elements: Advanced event processing business object

Metadata element	Description
IDOCName	Name of the IDOC
FieldName	Actual BAPI Field name as represented in SAP
PrimaryKey	An indication of whether this property is a primary key.
ForeignKey	Foreign key relationship
MaxLength	The length of the field.

The following illustration is an example of Advanced event processing property-level metadata for the Messagetype property:

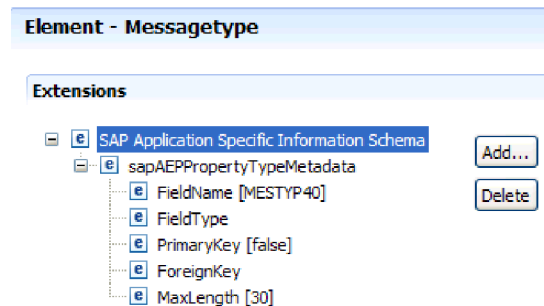


Figure 120. Property-level metadata for Messagetype

Operation-level metadata for Advanced event processing business objects

Operation-level metadata for an Advanced event processing 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 Advanced event processing business object operation.

Note: Outbound objects use only the Name metadata element.

Table 36. Operation-level metadata elements: Advanced event processing business object

Metadata element	Description
Name	The name of the operation.
MethodName	The name of the ABAP handler for this operation.
RouterName	The name of the router.

Supported data operations

For outbound processing, an operation is the name of the action *implemented by the adapter* so that the client application component can perform the operation on the SAP server. The adapter uses the application-specific information (ASI) inside the business object definition to implement the operation. The name of the operation typically indicates the type of action to be implemented, such as *create* or *update*. For inbound processing, adapters implement an operation by delivering events to their endpoints. For inbound processing, the action associated with the event varies depending on the interface (ALE or Advanced event processing). When ALE is the interface, the action is pushed to the adapter and the adapter delivers the event to an endpoint. When Advanced event processing is the interface, the event status is polled by the adapter and processed accordingly.

Supported data operations on BAPI business objects

The operation of a BAPI business object is the name of the BAPI call that an adapter issues on the SAP server during outbound processing. The BAPI method determines the operation associated with it. The adapter uses the application-specific information (ASI) inside the business object definition to implement the operation.

BAPIs and BAPI work unit

Operations of a business object are invoked by the 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 for BAPIs and BAPI work unit.

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 37. Supported operations: BAPI business objects

Operation	Definition
Create	The top-level business object and all contained children are created.

Table 37. Supported operations: BAPI business objects (continued)

Operation	Definition
Update	The top-level business object is modified. This operation can include adding and deleting child objects.
Delete	The top-level business object and any contained children are deleted.
Retrieve	The top-level business object and any contained children are retrieved.
Execute	The top-level business object and any contained children is executed. Note: This operation is available only if the Generate BAPIs within Wrappers check box is not selected. If the Configure Wrapper Business Object for Selected BAPI check box is selected, then other operations, such as Create, Update, Delete, and Retrieve are available.

For an operation that is not supported, the adapter logs the appropriate error and produces a ResourceException.

Result sets

The following table defines the operation that the adapter supports for BAPI result sets.

Table 38. Supported operation: BAPI result sets

Operation	Definition
RetrieveAll	All the matching records for the BAPI result set are retrieved.

The adapter uses the metadata information from the wrapper business object to find the operation associated with the received RFC-enabled function name. The adapter uses the application-specific information (ASI) inside the business object definition to implement the operation. After the adapter determines the operation, it sets it on the business object before sending it to the endpoint.

Supported data operations on ALE business objects

The operations that are supported by ALE business objects vary, depending on whether the business object is an outbound or inbound object. The adapter uses the application-specific information (ASI) inside the business object definition to implement the operation.

Note: Business objects generated with the ALE pass-through IDoc interface are not associated with an operation.

Outbound business objects

The operation of an ALE outbound business object is invoked by the application component that makes calls to SAP through the adapter. The adapter supports the following outbound operation.

Table 39. Supported operation: ALE outbound business objects

Operation	Definition
Execute	<p>Posts the IDoc business object to the SAP application. This is a one-way, asynchronous operation.</p> <ul style="list-style-type: none"> • 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.

Inbound business objects

For ALE inbound business objects, the application-specific information of an operation contains the message type, message code, and message function for an IDoc type. The adapter supports the following inbound operations.

Table 40. Supported operations: ALE inbound business objects

Operation	Definition
Create	The top-level business object and all contained children are created.
Update	The top-level business object is modified. This operation can include adding and deleting child objects.
Delete	The top-level business object and any contained children are deleted.

The adapter uses the IDoc control record field data to determine the operation that is set on the business object before sending it to the endpoint. The following fields in the control record are used to determine the operation:

- Logical_message_type (MESTYP)
- Logical_message_code (MESCOD)
- Logical_message_function(MESFCT)

Supported data operations of Query interface for SAP Software 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 adapter uses the application-specific information (ASI) inside the business object definition to implement the operation.

The supported operations for the Query interface for SAP Software are listed in the following table.

Table 41. Supported operations: Query interface for SAP Software business objects

Operation	Description
RetrieveAll	Returns a result set in the form of a container of SAP query business objects, which represent the data for each row retrieved from the table. If a table business object is sent to the SAP server (instead of a container business object), the rows are returned one at a time.

Table 41. Supported operations: Query interface for SAP Software business objects (continued)

Operation	Description
Exists	Provides a means to check for the existence of any records in SAP for a defined search criteria. The Exists operation does not return any data; it indicates whether the data exists in SAP. If no data is found, the adapter generates an exception.

Supported data operations on Advanced event processing business objects

The operations that are supported by Advanced event processing business objects vary, depending on whether the business object is an outbound or inbound object. The adapter uses the application-specific information (ASI) inside the business object definition to implement the operation.

Outbound business objects

The operation of an Advanced event processing outbound business object is invoked by the client application that makes calls to SAP through the adapter. The adapter supports the following outbound operation.

Table 42. Supported operation: Advanced event processing outbound business objects

Operation	Definition
Create	The top-level business object and all contained children are created.
Update	The top-level business object is modified. This operation can include adding and deleting child objects.
Delete	The top-level business object and any contained children are deleted.
Retrieve	The top-level business object and any contained children are retrieved.

Inbound business objects

For Advanced event processing inbound business objects, the application-specific information of an operation contains the message type, message code, and message function for an IDoc type. The adapter supports the following inbound operations.

Table 43. Supported operations: Advanced event processing inbound business objects

Operation	Definition
Create	The top-level business object and all contained children are created.
Update	The top-level business object is modified. This operation can include adding and deleting child objects.
Delete	The top-level business object and any contained children are deleted.

For IBM Business Process Manager or WebSphere Enterprise Service Bus, the verb value in event table determines the operation name for AEP inbound processing.

For WebSphere Application Server, after the message is received by the endpoint, the adapter the verb value in the event table to determine the operation that is set in OutputRecord().

Naming conventions

When the external service wizard generates a business object, it provides a name for the business object that is based on the name of the corresponding business function in the SAP server. The convention applied by the SAP server when naming a business object will vary slightly depending on whether the name is for a BAPI business object, an ALE business object, an Advanced event processing business object, or a Query interface for SAP Software business object.

Naming conventions for BAPI business objects

The external service wizard provides names for the business objects for BAPIs, BAPI work unit, and BAPI result sets. At its core, the business object name reflects the structure of the business function on the SAP server.

BAPIs

When naming business objects for BAPIs, the external service wizard adds a prefix of Sap then converts the name of the business function to mixed case, removing any separators such as spaces or underscores, capitalizes the first letter of each word and may add an element-specific suffix (for example, BG for business graph or Wrapper for top-level business object).

The following table describes the convention applied by the external service wizard when naming BAPI business objects.

Table 44. Naming conventions for BAPI business objects

Element	Naming convention
Name of the business graph	Sap + <i>Name of the wrapper object you specify in the external service wizard</i> + BG For example: SapSalesOrderBG
Name of the top-level business object	Sap + <i>Name of the wrapper object you specify in the external service wizard</i> + Wrapper For example: SapSalesOrderWrapper
Name of the BAPI business object	Sap + <i>Name of the BAPI interface</i> For example: SapBapiSalesOrderCreateFromDat1 Note: The top-level object can contain more than one BAPI object.
Name of the child object	Sap + <i>Name of the Structure/Table</i> For example: SapReturn

Note that business graph generation is optional and is supported for IBM Business Process Manager or WebSphere Enterprise Service Bus only.

When you select the **Generate business objects according to SAP naming conventions** option in the Specify the Discovery Properties window, the adapter generates all the Business Object names same as that of the SAP with original

casing. This enables the elements (at child and grand-child levels) which are complex type to appear in the SAP original casing.

If the module contains structures that have the same name, for example, the RETURN structure, the adapter handles Business Object name duplication depending on the value specified for the **Enforce the same naming convention for business objects** EMD property. The behaviour differs based on the selection in the following manner:

1. If the **Enforce the same naming convention for business objects** checkbox is checked, the adapter generates all the Business Object name without appending any hashcode. The hashcode is appended to the namespace of each Business Object instead of the name.

Example: If the repeating structure is RETURN, the adapter generates the corresponding Business Object name SapReturn. If there are three occurrences of the RETURN structure in the same module, then all three Business Objects will have the name SapReturn.

2. If the **Enforce the same naming convention for business objects** checkbox is unchecked, the adapter appends the hashcode to the subsequent Business Object that has the same name. This is done to avoid duplication.

Example: If the repeating structure is RETURN, the adapter generates the corresponding Business Object name SapReturn. If there are three occurrences of the RETURN structure in the same module,

- The first occurrence of the SapReturn Business Object has the name SapReturn.
- The second occurrence of the SapReturn Business Object has the name SapReturn203510914 where 203510914 is the hashcode of the String SapReturn.
- The third occurrence of the SapReturn Business Object has name SapReturn619647890 where 619647890 is the hashcode of the String SapReturn203510914.

BAPI work unit

The following table describes the convention applied by the external service wizard when naming a BAPI work unit business object.

Table 45. Naming conventions for BAPI work unit business objects

Element	Naming convention
Name of the business graph	Sap + <i>Name of the wrapper object you specify in the external service wizard</i> + Txn + BG For example: SapCustomerTxnBG
Name of the top-level business object	Sap + <i>Name of the wrapper object you specify in the external service wizard</i> + Txn For example: SapCustomerTxn
Name of the BAPI business object	Sap + <i>Name of the BAPI interface</i> For example: SapCustomer
Name of the child object	Sap + <i>Name of the Structure/Table</i> For example: SapReturn

Note that business graph generation is optional and is supported for IBM Business Process Manager or WebSphere Enterprise Service Bus only.

If the module contains structures that have the same name, for example, the RETURN structure, the adapter handles Business Object name duplication depending on the value specified for the **Enforce the same naming convention for business objects** EMD property. The behaviour differs based on the selection in the following manner:

1. If the **Enforce the same naming convention for business objects** checkbox is checked, the adapter generates all the Business Object name without appending any hashcode. The hashcode is appended to the namespace of each Business Object instead of the name.

Example: If the repeating structure is RETURN, the adapter generates the corresponding Business Object name SapReturn. If there are three occurrences of the RETURN structure in the same module, then all three Business Objects will have the name SapReturn.

2. If the **Enforce the same naming convention for business objects** checkbox is unchecked, the adapter appends the hashcode to the subsequent Business Object that has the same name. This is done to avoid duplication.

Example: If the repeating structure is RETURN, the adapter generates the corresponding Business Object name SapReturn. If there are three occurrences of the RETURN structure in the same module,

- The first occurrence of the SapReturn Business Object has the name SapReturn.
- The second occurrence of the SapReturn Business Object has the name SapReturn203510914 where 203510914 is the hashcode of the String SapReturn.
- The third occurrence of the SapReturn Business Object has name SapReturn619647890 where 619647890 is the hashcode of the String SapReturn203510914.

BAPI result sets

The following table describes the convention applied by the external service wizard when naming a BAPI result sets business object.

Table 46. Naming conventions for BAPI result sets

Element	Naming convention
Name of the top-level business object	Sap + <i>Name of the object you specify in the external service wizard</i> + Resultset For example: SapCustomerGetDetailResultset
Name of the result set BAPI business object	Sap + <i>Name of the BAPI interface</i> For example: SapBapiCustomerGetDetail
Name of the child object	Sap + <i>Name of the Structure/Table</i> For example: SapReturn
Name of the query business object	Sap + <i>Formatted name of the query BAPI interface</i> For example: SapBapiCustomerGetList

If the module contains structures that have the same name, for example, the RETURN structure, the adapter handles Business Object name duplication depending on the value specified for the **Enforce the same naming convention for business objects** EMD property. The behaviour differs based on the selection in the following manner:

1. If the **Enforce the same naming convention for business objects** checkbox is checked, the adapter generates all the Business Object name without appending any hashcode. The hashcode is appended to the namespace of each Business Object instead of the name.

Example: If the repeating structure is RETURN, the adapter generates the corresponding Business Object name SapReturn. If there are three occurrences of the RETURN structure in the same module, then all three Business Objects will have the name SapReturn.

2. If the **Enforce the same naming convention for business objects** checkbox is unchecked, the adapter appends the hashcode to the subsequent Business Object that has the same name. This is done to avoid duplication.

Example: If the repeating structure is RETURN, the adapter generates the corresponding Business Object name SapReturn. If there are three occurrences of the RETURN structure in the same module,

- The first occurrence of the SapReturn Business Object has the name SapReturn.
- The second occurrence of the SapReturn Business Object has the name SapReturn203510914 where 203510914 is the hashcode of the String SapReturn.
- The third occurrence of the SapReturn Business Object has name SapReturn619647890 where 619647890 is the hashcode of the String SapReturn203510914.

Naming conventions for ALE business objects

The external service wizard provides names for the ALE business graph, top-level business object, and the business object itself. At its core, the business object name reflects the structure of the business function on the SAP server.

Note: If you are using the ALE pass-through IDoc interface, the following naming conventions apply:

- When you select **Generic IDoc** from the Find Objects in the Enterprise System window, the external service wizard creates a business object named SapGenericIDocObject. The naming convention described in the following sections does not apply to generic IDocs.
- When you discover an IDoc from the system or from a file, the object is named according to the naming convention for top-level wrapper objects, as described in Table 47 on page 319. No other objects are generated.

When naming business objects for ALE, the external service wizard adds a prefix of Sap then converts the name of the IDoc and extension to mixed case, removing any separators such as spaces or underscores, capitalizes the first letter of each word and may add an element-specific suffix (for example, BG for business graph).

The following table describes the convention applied by the external service wizard when naming ALE business objects.

Note: The *[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 47. Naming conventions for ALE business objects

Element	Naming convention
Name of the business graph	Sap + <i>Name of IDoc</i> + [<i>Name of Extension type IDoc</i>] + BG For example: SapAlereq01BG
Name of the top-level wrapper object	Sap + <i>Name of IDoc</i> + [<i>Name of Extension type IDoc</i>] For example: SapAlereq01
Name of the IDoc business object for basic IDocs	Sap + <i>Name of IDoc</i> + B0 For example, the business object for the IDoc MATMAS03 is: SapMatmas03B0
Name of the IDoc business object for extension type IDocs	Sap + <i>Name of IDoc</i> + <i>Name of Extension type IDoc</i> For example, the business object for the IDoc DELVRY03 and extension SD_DESADV_PDC is: SapDelvry03SdDesadvPdc

Note that business graph generation is optional and is supported for IBM Business Process Manager or WebSphere Enterprise Service Bus only.

In the case of an IDoc duplicate name, the external service wizard adds a unique suffix to differentiate 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 appended to the name by the external service wizard.

Naming conventions for Query interface for SAP Software business objects

The external service wizard provides names for the Query interface for SAP Software container, business graph, top-level business object, table object, and query object. At its core, the business object name reflects the structure of the business function on the SAP server

When naming business objects for the Query interface for SAP Software, the external service wizard adds a prefix of Sap then converts the name of the business function or SAP table to mixed case, removing any separators such as spaces or underscores, capitalizes the first letter of each word and may add an element-specific suffix (for example, BG for business graph or Container for a container).

The following table describes the convention applied by the external service wizard when naming a Query interface for SAP Software business object.

Table 48. Naming convention for a Query interface for SAP Software business object

Element	Naming convention
Name of the container	Sap + <i>Name of the object you specify in the external service wizard</i> + Container For example: SapCustomerContainer
Name of the business graph	Sap + <i>Name of the object you specify in the external service wizard</i> + BG For example: SapCustomerBG

Table 48. Naming convention for a Query interface for SAP Software business object (continued)

Element	Naming convention
Name of the table object	Sap + <i>Name of the SAP table</i> For example: SapKna1
Name of the query object	Sap + <i>Name of the SAP table</i> + Querybo For example: SapKna1Querybo

Note that business graph generation is optional and is supported for IBM Business Process Manager or WebSphere Enterprise Service Bus only.

Naming conventions for Advanced event processing business objects

The external service wizard provides names for Advanced event processing business graph, top-level business object, and the business object itself. At its core, the business object name reflects the structure of the business function on the SAP server.

When naming business objects for the Advanced event processing interface, the external service wizard adds a prefix of Sap then converts the name of the IDoc and extension to mixed case, removing any separators such as spaces or underscores, capitalizes the first letter of each word and may add an element-specific suffix (for example, BG for business graph).

The following table describes the convention applied by the external service wizard when naming advanced event processing business objects.

Note: The [*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 49. Naming convention for Advanced event processing business objects

Element	Naming convention
Name of the top-level wrapper object	Sap + <i>Name of IDoc</i> + [<i>Name of Extension type IDoc</i>] For example: SapAepreq01
Name of the IDoc business object for basic IDocs	Sap + <i>Name of IDoc</i> For example, the business object for the IDoc MATMAS03 is: SapMatmas03
Name of the IDoc business object for extension type IDocs	Sap + <i>Name of IDoc</i> + <i>Name of Extension type IDoc</i> For example, the business object for the IDoc DELVRY03 and extension SD_DESADV_PDC is: SapDelvry03SdDesadvPdc

Note that business graph generation is optional and is supported for IBM Business Process Manager or WebSphere Enterprise Service Bus only.

In the case of an IDoc duplicate name, the external service wizard adds a unique suffix to differentiate 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 appended to the name by the external service wizard.

Outbound configuration properties

IBM WebSphere Adapter for SAP Software has several categories of outbound connection configuration properties, which you set with the external service wizard while generating or creating objects and services. You can change the resource adapter and managed connection factory properties after you deploy the module to IBM Business Process Manager or WebSphere Enterprise Service Bus using IBM Integration Designer or the Process Administrative Console, but connection properties for the external service wizard cannot be changed after deployment.

Guide to information about properties

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

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

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

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

Connection properties for the wizard

External service connection properties establish a connection between the external service wizard of IBM Integration Designer, a tool that is used to create business objects, and the SAP server. The properties you configure in the external service wizard specify such things as connection configuration, bidi properties and tracing and logging options.

Once a connection between the external service wizard and the SAP server is established, the external service wizard is able to access the metadata it needs from the SAP server to create business objects.

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

The external service connection properties and their purpose are described in the following table. A complete description of each property is provided in the sections that follow the table. For information on how to read the property detail tables in the sections that follow, see “Guide to information about properties” on page 321.

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

Table 50. External service connection properties Adapter for SAP Software

Property name	Description
"Bidi direction " on page 324	The orientation component of the bidi format specification.
"Bidi ordering schema" on page 324	The ordering scheme of the bidi format specification.
"Bidi numeric shaping" on page 324	The numeric shaping component of the bid format specification.
"Bidi shaping" on page 325	The shaping component of the bidi format specification.
"Bidi symmetric swapping" on page 325	The symmetric swapping component of the bid format specification.
"Client" on page 325	The client number of the SAP system to which the adapter connects.
"Codepage number" on page 326	Indicates the numeric identifier of the code page.
"Folder for RFC trace files" on page 326	Sets the fully qualified local path to the folder into which the RFC trace files are written.
"Host name" on page 327	Specifies the IP address or the name of the application server host that the adapter logs on to.
"Language code" on page 327	Specifies the language in which the adapter logs on.
"Log file output location property" on page 328	Specifies the location of the log file for external service.
"Logging level property" on page 328	Specifies the type error for which logging will occur during external service.
"Password" on page 329	The password of the user account of the adapter on the SAP application server.
"RFC trace level" on page 329	Specifies the global trace level.
"RFC trace on" on page 330	Specifies whether to generate a text file detailing the RFC activity for each event listener.
"SAP interface name" on page 330	Indicates the SAP interface to be used.
"System number" on page 331	The system number of the SAP application server.
"User name" on page 331	The user account for the adapter on the SAP server.

The external service wizard uses the bidirectional connection properties to apply the proper bidirectional transformation on the data passed to the SAP server.

For more details on how to set the character code set on IBM Business Process Manager or WebSphere Enterprise Service Bus for processing multilingual data (including bidirectional data), see the technical article titled "Overview of Bidirectional script support in WebSphere Process Server".

The bidi properties specify the bidirectional format for data coming from an external application into the adapter in the form of any business object supported by this adapter.

You should accept the default values for the bidirectional formatting properties on the external service wizard providing SAP server bidirectional format specification. When combined, these bidirectional properties define one single bidirectional format.

The default values for bidirectional formatting properties listed below are based on Windows bidirectional formatting. If the enterprise information system supports a bidirectional format other than the Windows standard bidirectional format, you must make appropriate changes to the bidi properties listed below.

Bidi direction

This property specifies the orientation component of the bidi format specification.

Table 51. Bidi direction details

Required	No
Possible values	Possible values include: <ul style="list-style-type: none">• LTR The orientation is left-to-right• RTL The orientation is right-to-left• contextualLTR The orientation is left-to-right because of the context. A character that is not categorized as LTR that is located between two strong characters with a different writing direction, will inherit the main context's writing direction (in an LTR document the character will become LTR).• contextualRTL The orientation is right-to-left because of the context. A character that is not categorized as RTL that is located between two strong characters with a different writing direction, will inherit the main context's writing direction (in an RTL document the character will become RTL).
Default	LTR
Property type	String
Usage	Specifies the orientation component of the bidi format specification.
Globalized	Yes
Bidi supported	No

Bidi ordering schema

This property specifies the ordering scheme of the bidi format specification.

Table 52. Bidi ordering schema details

Required	No
Possible values	Implicit Visual
Default	Implicit
Property type	String
Usage	Specifies the ordering scheme of the bidi format specification.
Globalized	Yes
Bidi supported	No

Bidi numeric shaping

This property specifies the numeric shaping component of the bidi format specification.

Table 53. Bidi numeric details

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

Table 53. Bidi numeric details (continued)

Possible values	Nominal National Contextual
Default	Nominal
Property type	String
Usage	Specifies the numeric shaping component of the bidi format specification.
Globalized	Yes
Bidi supported	No

Bidi shaping

This property specifies the shaping component of the bidi format specification.

Table 54. Bidi shaping details

Required	No
Possible values	Nominal Shaped Initial Middle Final Isolated
Default	Nominal
Property type	String
Usage	Specifies the shaping component of the bidi format specification.
Globalized	Yes
Bidi supported	No

Bidi symmetric swapping

This property specifies the symmetric swapping component of the bidi format specification.

Table 55. Bidi symmetric swapping details

Required	No
Possible values	True False
Default	True
Property type	Boolean
Usage	This property specifies the symmetric swapping component of the bidi format specification.
Globalized	Yes
Bidi supported	No

Client

This property is the client number of the SAP system to which the adapter connects.

Table 56. Client details

Required	Yes
Possible values	You can enter a range of values from 000 to 999.
Default	100
Property type	Integer
Usage	When an application attempts to log on to the SAP server, the SAP server requires that the application have a Client number associated with it. The Client property value identifies the client (the adapter) that is attempting to log onto the SAP server.
Globalized	No
Bidi supported	No

Codepage number

The numeric identifier of the code page.

Table 57. Codepage number details

Required	No
Possible values	You can enter a range of values from 0000 to 9999. For a full listing of languages and associated codepage numbers supported by SAP, access SAP Note 7360.
Default	The default value for this property is conditionally determined by the value set for the Language code property.
Property type	Integer
Usage	The value assigned to the Codepage number defines the code page to be used and has a one-to-one relationship with the value set for the Language code property. The Codepage number establishes a connection to the appropriate language. Each language code value has a codepage number value associated with it. For example, the language code for English, is EN. If you selected EN (English) as your language code, the codepage number is automatically set to the numeric value associated with EN (English). The SAP code page number for EN (English) is 1100.
Example	If Language code is set to JA (Japanese), Codepage number is set to 8000.
Globalized	No
Bidi supported	No

Folder for RFC trace files

This property sets the fully qualified local path to the folder in which to write RFC trace files.

Table 58. Folder for RFC trace files details

Required	No
Default	No default value
Property type	String

Table 58. Folder for RFC trace files details (continued)

Usage	Identifies the fully qualified local path into which RFC trace files are written. If RFC trace on is set to False (not selected), you are not permitted to set a value in the Folder for RFC trace files property. This field cannot be edited if you are modifying existing artifacts
Example	c:\temp\rfcTraceDir
Globalized	Yes
Bidi supported	No

Host name

Specifies the IP address or the name of the application server host that the adapter logs on to.

Table 59. Host name details

Required	Yes (when load balancing is not used).
Default	No default value
Property type	String
Usage	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 on to.
Example	sapServer
Globalized	No
Bidi supported	No

Language code

SAP logon language code.

Table 60. Language code details

Required	Yes
Possible values	Each of the supported languages is preceded by a 2 character language code. The language itself is displayed in parentheses. The language codes that display in the list represent the SAP default set of 41 languages for non Unicode systems plus Arabic. For a full listing of supported language codes and languages, see the SAP documentation.
Default	The default language code will be your current locale. If your current locale is not listed as one of the supported language codes, then a default language code of EN (English) is used.
Property type	String
Usage	If you manually enter a language code, you do not need to enter the language in parentheses.
Example	If the system locale is English, the value for this property is EN (English)
Globalized	No
Bidi supported	No

Log file output location property

This property specifies the location of the log file for external service discovery.

Table 61. Log file output location details

Required	Yes
Default	The .metadata directory of the workspace.
Property type	String
Usage	Use this directory to hold the log file that will list the errors that occur during the discovery process. The type of discovery errors for which logging occurs is controlled by the Logging level property
Example	C:\IBM\wid6.0\workspace\.metadata\SAPMetadataDiscovery.log
Globalized	Yes
Bidi supported	No

Logging level property

This property specifies the type error for which logging will occur during external service.

Table 62. Logging level details

Required	No
Possible values	FATAL SEVERE WARNING AUDIT INFO CONFIG DETAIL
Default	SEVERE
Property type	String
Usage	Use this property to tailor tracing capabilities. By specifying an error type, you are indicating that trace operations will occur only for errors of the type specified.

Table 62. Logging level details (continued)

Example	<p>Accepting the default value of SEVERE will provide trace information on errors that fall into the SEVERE category. Severe errors mean that an operation cannot continue, though the adapter can still function. Severe errors also include error conditions that indicate an impending fatal error, such as reporting on situations that strongly suggest that resources are on the verge of being depleted.</p> <p>Other error descriptions are as follows:</p> <ul style="list-style-type: none"> • Fatal Adapter cannot continue. Adapter cannot function • Warning Potential error or impending error. This also includes conditions that indicate a progressive failure – for example, the potential leaking of resources. • Audit Significant event affecting adapter state or resources • Info General information outlining overall operation progress. • Config Configuration change or status. • Detail General information detailing operation progress
Globalized	Yes
Bidi supported	No

Password

This property is the password of the user account of the adapter on the SAP application server.

Table 63. Password details

Required	Yes
Default	No default value
Property type	String
Usage	<p>The restrictions on the password depend on the version of SAP Web Application Server.</p> <ul style="list-style-type: none"> • For SAP Web Application Server version 6.40 or earlier, the password: <ul style="list-style-type: none"> – Must be uppercase – Must be 8 characters in length • For versions of SAP Web Application Server later than 6.40, the password: <ul style="list-style-type: none"> – Is not case-sensitive – Can be up to 40 characters in length
Globalized	No
Bidi supported	Yes

RFC trace level

This property specifies the global trace level.

Table 64. RFC trace level details

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

Table 64. RFC trace level details (continued)

Possible values	0 - No error 1 - Errors and warnings 2 - Execution path, errors and warnings 3 - Full Execution path, errors and warnings 4 - Execution path, info messages, errors and warnings 6 - Full execution path, info messages, errors and warnings 7 - Debug messages, full execution path, info messages, errors and warnings 8 - Verbose debug messages, full execution path, info messages, errors and warnings
Default	1
Property type	Integer
Usage	If RFC trace on is set to False (not selected), you cannot set a value in the RFC trace level property.
Globalized	No
Bidi supported	No

RFC trace on

This property specifies whether to generate a text file detailing the RFC activity for each event listener.

Table 65. RFC trace on details

Required	No
Possible values	True False
Default	False
Property type	Boolean
Usage	A value of True activates tracing, which generates a text file. This file is created in the directory in which the adapter process was started. The file has a prefix of rfx and a file type of trc (for example, rfc03912_02220.trc). Use these text files in a development environment only, because the files can grow rapidly. If RFC trace on is set to False (not selected), you cannot set values in the Folder for RFC trace files or RFC trace level properties.
Example	Examples of the information in the file are RfcCall FUNCTION BAPI_CUSTOMER_GETLIST, followed by the information for the parameters in the interface, or RFC Info rftable, followed by the data from one of the interface tables. The trace file is created in the directory where the adapter process has been started. The trace file has a .trc file extension and the file name will start with the letters rfc followed by a unique identifier. For example, rfc03912_02220.trc.
Globalized	No
Bidi supported	No

SAP interface name

This property indicates whether you are creating business objects for the ALE, BAPI, Advanced event processing, or Query interface for SAP Software.

Table 66. SAP interface name details

Required	Yes
Possible values	<p>For outbound:</p> <ul style="list-style-type: none"> Advanced event processing (AEP) ALE ALE pass-through IDoc BAPI BAPI work unit BAPI result set Query interface for SAP Software (QSS) <p>For inbound:</p> <ul style="list-style-type: none"> Advanced event processing (AEP) ALE ALE pass-through IDoc BAPI
Default	<p>For outbound: BAPI</p> <p>For inbound: ALE</p>
Property type	String
Usage	<p>Specifies the interface used by the adapter.</p> <p>The adapter interacts with the interface to support outbound and or inbound processing by enabling the exchange of data in the form of business objects.</p> <p>This field cannot be edited if you are modifying existing artifacts</p>
Globalized	No
Bidi supported	No

System number

This property is the system number of the SAP application server.

Table 67. System number details

Required	Yes
Possible values	You can enter a range of values from 00 to 99.
Default	00
Property type	Integer
Usage	The system number further identifies the Gateway service.
Globalized	No
Bidi supported	No

User name

This property is the user account for the adapter on the SAP server.

Table 68. User name details

Required	Yes
Default	No default value

Table 68. User name details (continued)

Property type	String
Usage	Maximum length of 12 characters. The user name is not case sensitive. It is recommended that you set up a CPIC user account in the SAP application and that you give this account the necessary privileges to manipulate the data required by the business objects supported by the adapter. For example, if the adapter must perform certain SAP business transactions, the adapter's account in the SAP application must have the permissions set to allow it to perform these transactions.
Example	SapUser
Globalized	Yes
Bidi supported	Yes

Resource adapter properties

The resource adapter properties control the general operation of the adapter. You set the resource adapter properties using the external service wizard when you configure the adapter. After deploying the adapter, use the administrative console to change these properties.

The following properties for logging and tracing are deprecated:

- LogFileSize
- LogFileName
- LogNumberOfFiles
- TraceFileSize
- TraceFileName
- TraceNumberOfFiles

The following table lists and describes the resource adapter properties. A more detailed description of each property is provided in the sections that follow the table. For information on how to read the property detail tables in the sections that follow, see "Guide to information about properties" on page 321.

Table 69. Resource adapter properties for Adapter for SAP Software

Property name		Description
In the wizard	In the administrative console	
Adapter ID	AdapterID	Identifies the adapter instance for PMI events and for logging and tracing.
Disguise user data as "XXX" in log and trace files	HideConfidentialTrace	Specifies whether to disguise potentially sensitive information by writing X strings instead of user data in the log and trace files.
(Not available)	"Enable high availability support (enableHASupport)" on page 334	Do not change this property.
(Not available)	LogFileSize	Deprecated
(Not available)	LogFilename	Deprecated
(Not available)	LogNumberOfFiles	Deprecated
(Not available)	TraceFileSize	Deprecated
(Not available)	TraceFileName	Deprecated

Table 69. Resource adapter properties for Adapter for SAP Software (continued)

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

Adapter ID (AdapterID)

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

Table 70. Adapter ID details

Required	Yes
Default	001
Property type	String
Usage	<p>This property identifies the adapter instance in the log and trace files, and also helps identify the adapter instance while monitoring adapters. The adapter ID is used with an adapter-specific identifier, SAPRA, to form the component name used by the Log and Trace Analyzer tool. For example, if the adapter ID property is set to 001, the component ID is SAPRA001.</p> <p>If you run multiple instances of the same adapter, ensure that the first eight characters of the adapter ID property are unique for each instance so that you can correlate the log and trace information to a particular adapter instance. By making the first seven characters of an adapter ID property unique, the component ID for multiple instances of that adapter is also unique, allowing you to correlate the log and trace information to a particular instance of an adapter.</p> <p>For example, when you set the adapter ID property of two instances of WebSphere Adapter for SAP Software to 001 and 002. The component IDs for those instances, SAPRA001 and SAPRA002, are short enough to remain unique, enabling you to distinguish them as separate adapter instances. However, instances with longer adapter ID properties cannot be distinguished from each other. If you set the adapter ID properties of two instances to Instance01 and Instance02, you will not be able to examine the log and trace information for each adapter instance because the component ID for both instances is truncated to SAPRAInstance.</p> <p>For inbound processing, the value of this property is set at the resource adapter level. For outbound processing, the value can be set both at the resource adapter level and the managed connection factory level. After you use the external service wizard to configure the adapter for outbound processing, you can set the resource adapter and managed connection factory properties independently. If you use the IBM Integration Designer assembly editor or the administrative console to reset these properties, ensure that you set them consistently, to prevent inconsistent marking of the log and trace entries.</p>
Globalized	Yes
Bidi supported	No

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

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

Table 71. Disguise user data as "XXX" in log and trace files details

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

Table 71. Disguise user data as "XXX" in log and trace files details (continued)

Possible values	True False
Default	False
Property type	Boolean
Usage	<p>If you set this property to True, the adapter replaces user data with a string of X's when writing to log and trace files.</p> <p>For inbound processing, the value of this property is set at the resource adapter level. For outbound processing, the value can be set both at the resource adapter level and the managed connection factory level. After you use the external service wizard to configure the adapter for outbound processing, you can set the resource adapter and managed connection factory properties independently. If you use the IBM Integration Designer assembly editor or the administrative console to reset these properties, ensure that you set them consistently, to prevent inconsistent marking of the log and trace entries.</p>
Globalized	No
Bidi supported	No

Enable high availability support (enableHASupport)

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

Managed connection factory properties

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

The following property that was specified as a Managed connection factory property in version 6.2.x applies to the interaction specification property group in version 7.5.0.2.

- IgnoreBAPIReturn

You set the managed connection factory properties using the external service wizard and can change them using the IBM Integration Designer Assembly Editor, or after deployment through the IBM Business Process Manager or WebSphere Enterprise Service Bus Process Administrative Console.

The following table lists and describes the managed connection factory properties. A more detailed description of each property is provided in the sections that follow the table. For information on how to read the property detail tables in the sections that follow, see "Guide to information about properties" on page 321.

Note: The external service wizard refers to these properties as managed connection factory properties and IBM Business Process Manager or WebSphere Enterprise Service Bus Process Administrative Console refers to these as (J2C) connection factory properties.

Table 72. Managed connection factory properties for Adapter for SAP Software

Property name		Description
In the wizard	In the Process Administrative Console	
Adapter ID	AdapterID	Identifies the adapter instance for PMI events and for logging and tracing.

Table 72. Managed connection factory properties for Adapter for SAP Software (continued)

Property name		Description
In the wizard	In the Process Administrative Console	
"Client" on page 337	Client	The client number of the SAP system to which the adapter connects.
"Codepage number" on page 338	Codepage	Indicates the numeric identifier of the code page.
Disguise user data as "XXX" in log and trace files	HideConfidentialTrace	Specifies whether to disguise potentially sensitive information by writing X strings instead of user data in the log and trace files.
"Maximum Number of retries in case of system connection failure" on page 339	connectionRetryLimit	The adapter will try connecting to the Enterprise Information System (EIS) for a specified number of tries. Select only if you want to reduce the number of connection exceptions in the outbound operation. If selected, adapter will validate the connection for each outbound request.
"Enable Secure Network Connection" on page 344	SnCMode	Indicates whether secure network connection mode is used.
"Folder for RFC trace files" on page 339	RfcTracePath	Sets the fully qualified local path to the folder into which the RFC trace files are written.
"Gateway host" on page 340	GatewayHost	The host name of the SAP gateway.
"Gateway service" on page 340	GatewayService	The identifier of the gateway on the gateway host that carries out the RFC services.
"Host name" on page 341	ApplicationServerHost	Specifies the IP address or the name of the application server host that the adapter logs on to.
"Language code" on page 341	Language code	Specifies the Language code in which the adapter logs on to SAP.
"Load Balancing" on page 339	loadBalancing	Specifies if your SAP configuration uses load balancing
"Message server host" on page 341	MessageServerHost	Specifies the name of the host on which the message server is running.
"Partner character set" on page 342	PartnerCharset	Specifies PartnerCharset encoding.
"Password" on page 342	Password	The password of the user account of the adapter on the SAP application server.
"Reset JCo Client after closing Connection Handle" on page 342	resetClient	This property optionally invokes the reset method on the JCo client to make sure changes on the SAP EIS are reflected in the client during an outbound transaction
"RFC trace level" on page 343	RfcTraceLevel	Specifies the global trace level.
"RFC trace on" on page 343	RfcTraceOn	Specifies whether to generate a text file detailing the RFC activity for each event listener.
"SAP system ID" on page 344	SAPSystemID	Specifies the system ID of the SAP system for which logon load balancing is allowed.
"Secure Network Connection library path" on page 345	SnCLib	Specifies the path to the library that provides the secure network connection service.

Table 72. Managed connection factory properties for Adapter for SAP Software (continued)

Property name		Description
In the wizard	In the Process Administrative Console	
"Secure Network Connection name" on page 345	SnCMyname	Specifies the name of the secure network connection.
"Secure Network Connection partner" on page 345	SnCPartnername	Specifies the name of the secure network connection partner.
"Secure Network Connection security level" on page 346	SnCQop	Specifies the level of security for the secure network connection.
"Enable Single Sign On (SSO)" on page 348	SsoMode	Enables the adapter to use Single Sign On as the authentication method.
"SSO Logon Ticket" on page 348	SsoLogonTicket	Specifies the SSO logon certificate to be used as the logon ticket.
"System number" on page 346	SystemNumber	The system number of the SAP application server.
"Time between retries in case of system connection failure (milliseconds)" on page 346	connectionRetryInterval	Specifies the time interval between attempts to restart the event listeners.
"User name" on page 347	userName	The user account for the adapter on the SAP server.
"Wait until commit call to SAP database is completed and returned" on page 347	WaitOnCommit	Enables adapter to wait until all time critical updates to SAP database are completed before calling commit on it.
"X509 certificate" on page 347	X509cert	Specifies the X509 certificate to be used as the logon ticket.

Adapter ID (AdapterID)

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

Table 73. Adapter ID details

Required	Yes
Default	001
Property type	String

Table 73. Adapter ID details (continued)

Usage	<p>This property identifies the adapter instance in the log and trace files, and also helps identify the adapter instance while monitoring adapters. The adapter ID is used with an adapter-specific identifier, SAPRA, to form the component name used by the Log and Trace Analyzer tool. For example, if the adapter ID property is set to 001, the component ID is SAPRA001.</p> <p>If you run multiple instances of the same adapter, ensure that the first eight characters of the adapter ID property are unique for each instance so that you can correlate the log and trace information to a particular adapter instance. By making the first seven characters of an adapter ID property unique, the component ID for multiple instances of that adapter is also unique, allowing you to correlate the log and trace information to a particular instance of an adapter.</p> <p>For example, when you set the adapter ID property of two instances of WebSphere Adapter for SAP Software to 001 and 002. The component IDs for those instances, SAPRA001 and SAPRA002, are short enough to remain unique, enabling you to distinguish them as separate adapter instances. However, instances with longer adapter ID properties cannot be distinguished from each other. If you set the adapter ID properties of two instances to Instance01 and Instance02, you will not be able to examine the log and trace information for each adapter instance because the component ID for both instances is truncated to SAPRAInstance.</p> <p>For inbound processing, the value of this property is set at the resource adapter level. For outbound processing, the value can be set both at the resource adapter level and the managed connection factory level. After you use the external service wizard to configure the adapter for outbound processing, you can set the resource adapter and managed connection factory properties independently. If you use the IBM Integration Designer assembly editor or the administrative console to reset these properties, ensure that you set them consistently, to prevent inconsistent marking of the log and trace entries.</p>
Globalized	Yes
Bidi supported	No

Client

This property is the client number of the SAP system to which the adapter connects.

Table 74. Client details

Required	Yes
Possible values	You can enter a range of values from 000 to 999.
Default	100
Property type	Integer
Usage	When an application attempts to log on to the SAP server, the SAP server requires that the application have a Client number associated with it. The Client property value identifies the client (the adapter) that is attempting to log onto the SAP server.
Globalized	No
Bidi supported	No

Codepage number

The numeric identifier of the code page.

Table 75. Codepage number details

Required	No
Possible values	You can enter a range of values from 0000 to 9999. For a full listing of languages and associated codepage numbers supported by SAP, access SAP Note 7360.
Default	The default value for this property is conditionally determined by the value set for the Language code property.
Property type	Integer
Usage	The value assigned to the Codepage number defines the code page to be used and has a one-to-one relationship with the value set for the Language code property. The Codepage number establishes a connection to the appropriate language. Each language code value has a codepage number value associated with it. For example, the language code for English, is EN. If you selected EN (English) as your language code, the codepage number is automatically set to the numeric value associated with EN (English). The SAP code page number for EN (English) is 1100.
Example	If Language code is set to JA (Japanese), Codepage number is set to 8000.
Globalized	No
Bidi supported	No

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

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

Table 76. Disguise user data as "XXX" in log and trace files details

Required	No
Possible values	True False
Default	False
Property type	Boolean
Usage	If you set this property to True, the adapter replaces user data with a string of X's when writing to log and trace files. For inbound processing, the value of this property is set at the resource adapter level. For outbound processing, the value can be set both at the resource adapter level and the managed connection factory level. After you use the external service wizard to configure the adapter for outbound processing, you can set the resource adapter and managed connection factory properties independently. If you use the IBM Integration Designer assembly editor or the administrative console to reset these properties, ensure that you set them consistently, to prevent inconsistent marking of the log and trace entries.
Globalized	No
Bidi supported	No

Load Balancing

This property specifies if your SAP configuration uses load balancing

Table 77. Load balancing details

Required	Yes
Possible values	TrueFalse
Default	False
Property type	Boolean
Usage	This value should be set to true if the SAP configuration uses load balancing. If set to true, Message server host, Logon group and SAP System ID need to be specified.
Globalized	No
Bidi supported	No

Maximum Number of retries in case of system connection failure

This property specifies the number of times the adapter attempts to create a connection to the Enterprise Information System (EIS). The adapter will try connecting to the EIS for the specified number of times. Select only if you want to reduce the number of connection exceptions in the outbound operation. If selected, the adapter will validate the connection for each outbound request.

Table 78. Reset Client details

Required	No
Possible values	Integers
Default	0
Property type	Integer
Usage	<p>Only positive values are valid.</p> <p>When the adapter encounters an error related to the outbound connection, it retries to establish a physical connection (when physical connection is not established) for the number of times specified for this property with a time delay specified in the property "Time between retries in case of system connection failure (milliseconds)" on page 346.</p> <p>If the value is 0, the adapter does not perform any EIS connection validation and executes the outbound operation.</p> <p>if the value is > 0, then during each request the adapter validates if the EIS connection is active.</p> <ul style="list-style-type: none">• If the connection is valid the operation is completed.• if connection is invalid, the adapter invalidates the current managed connection and a new managed connection is created (new physical connection)
Globalized	No
Bidi supported	No

Folder for RFC trace files

This property sets the fully qualified local path to the folder in which to write RFC trace files.

Table 79. Folder for RFC trace files details

Required	No
Default	No default value
Property type	String
Usage	Identifies the fully qualified local path into which RFC trace files are written. If RFC trace on is set to False (not selected), you are not permitted to set a value in the Folder for RFC trace files property. This field cannot be edited if you are modifying existing artifacts
Example	c:\temp\rfcTraceDir
Globalized	Yes
Bidi supported	No

Gateway host

This property is the Gateway host name. Enter either the IP address or the name of the Gateway host. Consult with your SAP administrator for information on the Gateway host name.

Table 80. Gateway host details

Required	Yes
Default	No default value
Property type	String
Usage	This property is the host name of the SAP gateway. The gateway enables communication between work processes on the SAP system and external programs. The host identified is used as the gateway for the resource adapter. Maximum length of 20 characters. If the computer name is longer than 20 characters, define a symbolic name in the THOSTS table.
Globalized	No
Bidi supported	No

Gateway service

This property is the identifier of the gateway on the gateway host that carries out the RFC services.

Table 81. Gateway service details

Required	Yes
Default	sapgw00
Property type	String
Usage	These services enable communication between work processes on the SAP server and external programs. The service typically has the format of sapgw00, where 00 is the SAP system number. Maximum of 20 characters.
Globalized	No
Bidi supported	No

Host name

Specifies the IP address or the name of the application server host that the adapter logs on to.

Table 82. Host name details

Required	Yes (when load balancing is not used).
Default	No default value
Property type	String
Usage	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 on to.
Example	sapServer
Globalized	No
Bidi supported	No

Language code

This property specifies the Language code in which the adapter logs on.

Table 83. Language code details

Required	Yes
Possible values	For a full listing of languages and associated codepage numbers supported by SAP, access SAP Note 7360.
Default	The default value for the Language code property is based on the system locale.
Property type	String
Usage	Each of the supported languages is preceded by a 2 character language code. The language itself is displayed in parentheses. The language codes that display in the list represent the SAP default set of 41 languages for non Unicode systems plus Arabic. The value you select determines the value of the Codepage number property. If you manually enter a language code, you do not need to enter the language in parentheses.
Example	If the system locale is English, the value for this property is EN (English).
Globalized	No
Bidi supported	No

Message server host

This property specifies the name of the host on which the message server is running.

Table 84. Message server host details

Required	Yes (if load balancing is used)
Default	No default value
Property type	String

Table 84. Message server host details (continued)

Usage	This property specifies the name of the host that will inform all the servers (instances) belonging to this SAP system of the existence of the other servers to be used for load balancing. The message server host contains the information about load balancing for RFC clients so that an RFC client can be directed to an appropriate application server.
Example	SAPERP05
Globalized	No
Bidi supported	No

Partner character set

This property specifies the partner character set encoding.

Table 85. Partner character set details

Required	No
Default	UTF-8
Property type	String
Usage	When an encoding is specified, it is used; otherwise the default encoding is used.
Globalized	No
Bidi supported	No

Password

This property is the password of the user account of the adapter on the SAP application server.

Table 86. Password details

Required	Yes
Default	No default value
Property type	String
Usage	The restrictions on the password depend on the version of SAP Web Application Server. <ul style="list-style-type: none"> • For SAP Web Application Server version 6.40 or earlier, the password: <ul style="list-style-type: none"> – Must be uppercase – Must be 8 characters in length • For versions of SAP Web Application Server later than 6.40, the password: <ul style="list-style-type: none"> – Is not case-sensitive – Can be up to 40 characters in length
Globalized	No
Bidi supported	Yes

Reset JCo Client after closing Connection Handle

This property optionally invokes the reset method on the JCo client to make sure changes on the SAP EIS are reflected in the client during an outbound transaction

Table 87. Reset Client details

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

Table 87. Reset Client details (continued)

Possible values	True False
Default	False
Property type	Boolean
Usage	<p>When the property is set to True, the adapter invokes the reset method on the JCo client to make sure changes on the SAP EIS are reflected in the client during an outbound transaction.</p> <p>When this property is set to False the adapter does not invoke the reset method on the JCo client, therefore any changes on the SAP EIS are not reflected in the client during an outbound transaction.</p> <p>This property is supported on the ALE interfaces - ALE and ALE Passthrough, and the BAPI interfaces - simple BAPI, BAPI work unit and BAPI ResultSet.</p>
Globalized	NA
Bidi supported	No

RFC trace level

This property specifies the global trace level.

Table 88. RFC trace level details

Required	No
Possible values	<p>0 - No error</p> <p>1 - Errors and warnings</p> <p>2 - Execution path, errors and warnings</p> <p>3 - Full Execution path, errors and warnings</p> <p>4 - Execution path, info messages, errors and warnings</p> <p>6 - Full execution path, info messages, errors and warnings</p> <p>7 - Debug messages, full execution path, info messages, errors and warnings</p> <p>8 - Verbose debug messages, full execution path, info messages, errors and warnings</p>
Default	1
Property type	Integer
Usage	If RFC trace on is set to False (not selected), you cannot set a value in the RFC trace level property.
Globalized	No
Bidi supported	No

RFC trace on

This property specifies whether to generate a text file detailing the RFC activity for each event listener.

Table 89. RFC trace on details

Required	No
Possible values	True False
Default	False
Property type	Boolean

Table 89. RFC trace on details (continued)

Usage	<p>A value of True activates tracing, which generates a text file.</p> <p>This file is created in the directory in which the adapter process was started. The file has a prefix of rfx and a file type of trc (for example, rfc03912_02220.trc).</p> <p>Use these text files in a development environment only, because the files can grow rapidly.</p> <p>If RFC trace on is set to False (not selected), you cannot set values in the Folder for RFC trace files or RFC trace level properties.</p>
Example	<p>Examples of the information in the file are RfcCall FUNCTION BAPI_CUSTOMER_GETLIST, followed by the information for the parameters in the interface, or RFC Info rftable, followed by the data from one of the interface tables.</p> <p>The trace file is created in the directory where the adapter process has been started. The trace file has a .trc file extension and the file name will start with the letters rfc followed by a unique identifier. For example, rfc03912_02220.trc.</p>
Globalized	No
Bidi supported	No

SAP system ID

This property specifies the system ID of the SAP system for which logon load balancing is allowed.

Table 90. SAP system ID details

Required	Yes (when load balancing is used)
Default	No default value
Property type	String
Usage	Value must be three characters
Example	DYL
Globalized	No
Bidi supported	No

Enable Secure Network Connection

This property indicates whether secure network connection mode is enabled.

Table 91. Enable Secure Network Connection details

Required	No
Possible values	0 (off) 1 (on)
Default	0
Property type	String
Usage	<p>Set the value to 1 (on) if you want to use secure network connection.</p> <p>If you set this value to 1, you must also set following properties:</p> <ul style="list-style-type: none"> • SncLib • SncMyname • SncPartnername • SncQop

Table 91. Enable Secure Network Connection details (continued)

Globalized	No
Bidi supported	No

Secure Network Connection library path

This property specifies the path to the library that provides the secure network connection service.

Table 92. Secure Network Connection library path details

Required	Yes, if SncMode is set to 1; no otherwise.
Default	No default value
Property type	String
Usage	If the SncMode property is set to 1 (indicating that you are using a secure network connection), specify the path to the library that provides the service.
Example	/WINDOWS/system32/gssapi32.dll
Globalized	No
Bidi supported	No

Secure Network Connection name

This property specifies the name of the secure network connection.

Table 93. Secure Network Connection name details

Required	Yes, if SncMode is set to 1; no otherwise.
Default	No default value
Property type	String
Usage	If the SncMode property is set to 1 (indicating that you are using a secure network connection), specify a name for the connection.
Example	DOMAINNAME/USERNAME
Globalized	No
Bidi supported	No

Secure Network Connection partner

This property specifies the name of the secure network connection partner.

Table 94. Secure Network Connection partner details

Required	Yes, if SncMode is set to 1; no otherwise.
Default	No default value
Property type	String
Usage	If the SncMode property is set to 1 (indicating that you are using a secure network connection), specify a name for the connection partner.
Example	CN=sap00.saperpdev, OU=Adapter, O=IBM, C=US
Globalized	No
Bidi supported	No

Secure Network Connection security level

This property specifies the level of security for the secure network connection.

Table 95. Secure Network Connection security level details

Required	Yes, if SncMode is set to 1; no otherwise.
Possible values	1 (Authentication only) 2 (Integrity protection) 3 (Privacy protection) 8 (Use the value from snc/data_protection/use on the application server) 9 (Use the value from snc/data_protection/max on the application server)
Default	3 (Privacy protection)
Property type	String
Usage	If the SncMode property is set to 1 (indicating that you are using a secure network connection), specify a value to indicate the level of security for the connection.
Globalized	No
Bidi supported	No

System number

This property is the system number of the SAP application server.

Table 96. System number details

Required	Yes
Possible values	You can enter a range of values from 00 to 99.
Default	00
Property type	Integer
Usage	The system number further identifies the Gateway service.
Globalized	No
Bidi supported	No

Time between retries in case of system connection failure (milliseconds)

This property specifies the time interval between attempts to connect to the Enterprise Information System (EIS).

Table 97. Time between retries in case of system connection failure details

Required	No
Possible Values	Positive Integers
Default	60000
Unit of measure	Milliseconds
Property type	Integer
Usage	When the adapter encounters an error related to the outbound connection, this property specifies the time interval that the adapter waits in between attempts to reestablish an outbound connection. It is disabled by default and is only enabled when the value of "Maximum Number of retries in case of system connection failure" on page 339 is greater than 0.

Table 97. Time between retries in case of system connection failure details (continued)

Globalized	No
Bidi supported	No

User name

This property is the user account for the adapter on the SAP server.

Table 98. User name details

Required	Yes
Default	No default value
Property type	String
Usage	Maximum length of 12 characters. The user name is not case sensitive. It is recommended that you set up a CPIC user account in the SAP application and that you give this account the necessary privileges to manipulate the data required by the business objects supported by the adapter. For example, if the adapter must perform certain SAP business transactions, the adapter's account in the SAP application must have the permissions set to allow it to perform these transactions.
Example	SapUser
Globalized	Yes
Bidi supported	Yes

Wait until commit call to SAP database is completed and returned

For BAPI outbound processing, this property specifies whether the commit call from the adapter should wait until all the time-critical (V1) updates to the SAP database have been completed. When this property is set to true, the commit to call to SAP waits until all updates are completed and returned. If it is set to false, the commit call returns immediately from the SAP database.

This property is only active when you use the CWYAP_SAPAdapter_Tx.rar file.

Table 99. WaitOnCommit property details

Required	No
Default	False
Property type	Boolean
Globalized	No
Bidi supported	No

X509 certificate

This property specifies the X509 certificate to be used as the logon ticket.

Table 100. X509 certificate details

Required	No.
Default	No default value
Property type	String

Table 100. X509 certificate details (continued)

Usage	If the SncMode property is set to 1 (indicating that you are using a secure network connection), you can provide a value for the X509 certificate.
Globalized	No
Bidi supported	No

SSO Logon Ticket

This property specifies the SSO logon certificate to be used as the logon ticket.

Table 101. Enable Single Sign On details

Required	Yes, if SSO is used for authentication; no otherwise.
Default	False
Property type	Boolean
Usage	If this property is set to true, the adapter authenticates with the SAP system using SSO tokens.
Globalized	No
Bidi supported	No

Enable Single Sign On (SSO)

This property enables Single Sign On as the authentication method.

Table 102. Enable Single Sign On details

Required	Yes, if SSO is used for authentication; no otherwise.
Default	False
Property type	Boolean
Usage	If this property is set to true, the adapter authenticates with the SAP system using SSO tokens.
Globalized	No
Bidi supported	No

Interaction specification properties

An interaction is an operation. Interaction specification properties control how the operation is run. The external service wizard sets the interaction specification properties when you configure the adapter.

Table 103 on page 349 lists and describes the interaction specification properties that you set. For information about how to read the property detail tables in the sections that follow, see “Guide to information about properties” on page 321.

Note: Typically, you do not need to change these properties. However, you can change some properties for outbound operations. For example, you might increase the value of the interaction specification property that specifies the maximum number of hits to be returned by a RetrieveAll operation, if your RetrieveAll operations do not return complete information. Use the assembly editor in IBM Integration Designer to change these properties, which reside in the method binding of the import.

Table 103. Interaction specification properties for Adapter for SAP Software

Property name		Description
In the wizard	In the assembly editor	
“Custom retrieve function name”	customFunctionName	Indicates the name of a custom function to be used by the Query interface to SAP Software to retrieve data from an SAP table.
Function name	functionName	Populates the function name for the specific SAP interface.
Ignore errors in BAPI return	IgnoreBAPIReturn	Indicates if errors in BAPI returns will be ignored.
“Maximum number of hits for the discovery” on page 351	ResultSetLimit	Maximum number of result sets to return during a RetrieveAll operation.
“Select the queue name” on page 351	QRFCQueueName	The name of a customer-defined queue on the SAP server.
“Wait until commit call to SAP database is completed and returned” on page 352	WaitOnCommit	Enables adapter to wait until all time critical updates to SAP database are completed before calling commit on it.

Custom retrieve function name

For the Query interface for SAP Software, this property specifies the name of a custom function that should be used to retrieve data from an SAP table.

Table 104. Custom retrieve function name details

Required	No
Default	No default value
Property type	String
Usage	<p>This property applies to Query interface for SAP Software only.</p> <p>On non-Unicode systems, the default function used to retrieve data from SAP tables (RFC_READ_TABLE) might produce an exception. To avoid the problem, you can create another function on the SAP server and then indicate, during configuration, that the adapter should use this custom function to retrieve data. This property specifies the name of the custom function.</p> <p>Note: You must create the function on the SAP server before you specify this property on the external service wizard. Follow the steps listed in SAP note 758278 to make a copy of RFC_READ_TABLE and modify the copy per the note.</p>
Globalized	No
Bidi supported	No

Function name

The functionName interaction specification property controls the interaction by associating operations with the proper interface.

Table 105. Function name details

Required	Yes
Possible values	True False
Default	Null

Table 105. Function name details (continued)

Property type	String
Usage	<p>The BAPI outbound and inbound interfaces support the following values for the functionName interaction specification property:</p> <p>WBIInteractionSpec.CREATE WBIInteractionSpec.UPDATE WBIInteractionSpec.RETRIEVE WBIInteractionSpec.DELETE</p> <p>The BAPI result set supports the following values for the functionName interaction specification property:</p> <p>WBIInteractionSpec.RETRIEVEALL</p> <p>The ALE outbound interface supports the following value for the functionName interaction specification property:</p> <p>WBIInteractionSpec.EXECUTE</p> <p>The ALE inbound interface supports the following values for the functionName interaction specification property:</p> <p>WBIInteractionSpec.CREATE WBIInteractionSpec.UPDATE WBIInteractionSpec.RETRIEVE WBIInteractionSpec.DELETE</p> <p>The Query interface for SAP software (QISS) interface supports the following values for the functionName interaction specification property:</p> <ul style="list-style-type: none"> • WBIInteractionSpec.EXISTS Throws exceptions NotExistsException and QISSQueryFailedException • WBIInteractionSpec.RETRIEVEALL Throws exceptions QISSQueryFailedException <p>The Advanced event processing interface for inbound processing supports the following values for the functionName interaction specification property:</p> <p>WBIInteractionSpec.CREATE WBIInteractionSpec.UPDATE WBIInteractionSpec.DELETE</p> <p>The Advanced event processing interface for outbound processing supports the following values for the functionName interaction specification property:</p> <p>WBIInteractionSpec.CREATE WBIInteractionSpec.UPDATE WBIInteractionSpec.RETRIEVE WBIInteractionSpec.DELETE</p>
Globalized	No
Bidi supported	No

Ignore errors in BAPI return

This property indicates whether to ignore errors specified in a BAPI return operation. The return structure can be data or a table.

Table 106. Ignore errors in BAPI return details

Required	No
Possible values	True False
Default	False
Property type	Boolean
Usage	This property applies to BAPI outbound synchronous RFC processing only. When set to True, the Adapter for SAP Software <i>ignores</i> the checking of error code in the BAPI RETURN structure after BAPI has run, and returns this structure to user as is. Note: RETURN structure is part of every BAPI and contains status of the BAPI execution. Accepting the default value of False results in the adapter processing the RETURN structure and throwing an exception if an error code is found.
Globalized	No
Bidi supported	No

Maximum number of hits for the discovery

For the Query interface for SAP Software, this property specifies the maximum number of result sets, which represents data for each row retrieved from a table through a RetrieveAll operation.

Table 107. Result set limit details

Required	Yes
Default	100
Property type	Integer
Usage	This property applies to Query interface for SAP Software only. If the number of hits in the table on the SAP server exceeds the value of the ResultSetLimit property, the adapter returns the error MatchesExceededLimitException. The adapter uses this property to help avoid out-of-memory issues.
Globalized	No
Bidi supported	No

Select the queue name

For BAPI outbound processing, when Asynchronous queued RFC is selected, this property specifies the name of a queue on the SAP server to which BAPIs will be delivered.

Table 108. Select the queue name details

Required	No
Default	The first queue defined on the SAP server. If no queue is defined on the SAP server, there is no default value.
Property type	String
Usage	This property applies to BAPI outbound asynchronous queued RFC processing only. When you want to deliver BAPI calls to a queue on the SAP server, you must specify the name of the queue. During configuration, you select, from a drop-down list, an existing queue. If no queues exist on the SAP server, you can type the name of a queue.

Table 108. Select the queue name details (continued)

Globalized	No
Bidi supported	No

Wait until commit call to SAP database is completed and returned

For BAPI outbound processing, this property specifies whether a commit call from the adapter should wait until all the time-critical (V1) updates to the SAP database have been completed. When this property is set to true, the commit to call to SAP waits until all updates are completed and returned. If it is set to false, the commit call returns immediately from the SAP database.

This property is only active when you use the CWYAP_SAPAdapter.rar file.

Table 109. WaitOnCommit property details

Required	No
Default	False
Property type	Boolean
Globalized	No
Bidi supported	No

Inbound configuration properties

WebSphere Adapter for SAP Software has several categories of inbound connection configuration properties, which you set with the external service wizard while generating or creating objects and services. You can change the resource adapter and activation specification properties after you deploy the module using IBM Integration Designer or the Process Administrative Console, but connection properties for the external service wizard cannot be changed after deployment.

Use this table if you are planning to configure a J2C activation specification manually, which is used by the resource adapter when configuring a specific endpoint instance:

Table 110. Message listener type

Interface	Message listener type
ALE	com.ibm.j2ca.base.ExtendedInboundListener supported by com.ibm.j2ca.sap.SAPActivationSpecwithXID
RFC / BAPI	com.ibm.j2ca.SAPRfcserverListener supported by com.ibm.j2ca.sap.SAPRFCActivationSpec
Transactional RFC / BAPI	com.ibm.j2ca.SAPtrfcserverListener supported by com.ibm.j2ca.sap.SAPtrRFCActivationSpec
AEP	com.ibm.j2ca.base.SAPAEPListner supported by com.ibm.j2ca.sap.aep.inbound.SAPAEPActivationSpecwithXID

Table 110. Message listener type (continued)

Interface	Message listener type
Pass - through	com.ibm.j2ca.base.SAPPassThroughListner supported by com.ibm.j2ca.sap.SAPPassThroughActivationSpecActivationS

Guide to information about properties

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

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

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

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

Connection properties for the wizard

External service connection properties establish a connection between the external service wizard of IBM Integration Designer, a tool that is used to create business objects, and the SAP server. The properties you configure in the external service wizard specify such things as connection configuration, bidi properties and tracing and logging options.

Once a connection between the external service wizard and the SAP server is established, the external service wizard is able to access the metadata it needs from the SAP server to create business objects.

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

The external service connection properties and their purpose are described in the following table. A complete description of each property is provided in the sections that follow the table. For information on how to read the property detail tables in the sections that follow, see “Guide to information about properties” on page 321.

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

Table 111. External service connection properties Adapter for SAP Software

Property name	Description
"Bidi direction " on page 356	The orientation component of the bidi format specification.
"Bidi ordering schema" on page 356	The ordering scheme of the bidi format specification.
"Bidi numeric shaping" on page 356	The numeric shaping component of the bid format specification.
"Bidi shaping" on page 357	The shaping component of the bidi format specification.
"Bidi symmetric swapping" on page 357	The symmetric swapping component of the bid format specification.
"Client" on page 357	The client number of the SAP system to which the adapter connects.
"Codepage number" on page 358	Indicates the numeric identifier of the code page.
"Folder for RFC trace files" on page 358	Sets the fully qualified local path to the folder into which the RFC trace files are written.
"Host name" on page 359	Specifies the IP address or the name of the application server host that the adapter logs on to.
"Language code" on page 359	Specifies the language in which the adapter logs on.
"Log file output location property" on page 360	Specifies the location of the log file for external service.
"Logging level property" on page 360	Specifies the type error for which logging will occur during external service.
"Password" on page 361	The password of the user account of the adapter on the SAP application server.
"RFC trace level" on page 361	Specifies the global trace level.
"RFC trace on" on page 362	Specifies whether to generate a text file detailing the RFC activity for each event listener.
"SAP interface name" on page 362	Indicates the SAP interface to be used.
"System number" on page 363	The system number of the SAP application server.
"User name" on page 363	The user account for the adapter on the SAP server.

The external service wizard uses the bidirectional connection properties to apply the proper bidirectional transformation on the data passed to the SAP server.

For more details on how to set the character code set on IBM Business Process Manager or WebSphere Enterprise Service Bus for processing multilingual data (including bidirectional data), see the technical article titled "Overview of Bidirectional script support in WebSphere Process Server".

The bidi properties specify the bidirectional format for data coming from an external application into the adapter in the form of any business object supported by this adapter.

You should accept the default values for the bidirectional formatting properties on the external service wizard providing SAP server bidirectional format specification. When combined, these bidirectional properties define one single bidirectional format.

The default values for bidirectional formatting properties listed below are based on Windows bidirectional formatting. If the enterprise information system supports a bidirectional format other than the Windows standard bidirectional format, you must make appropriate changes to the bidi properties listed below.

Bidi direction

This property specifies the orientation component of the bidi format specification.

Table 112. Bidi direction details

Required	No
Possible values	Possible values include: <ul style="list-style-type: none">• LTR The orientation is left-to-right• RTL The orientation is right-to-left• contextualLTR The orientation is left-to-right because of the context. A character that is not categorized as LTR that is located between two strong characters with a different writing direction, will inherit the main context's writing direction (in an LTR document the character will become LTR).• contextualRTL The orientation is right-to-left because of the context. A character that is not categorized as RTL that is located between two strong characters with a different writing direction, will inherit the main context's writing direction (in an RTL document the character will become RTL).
Default	LTR
Property type	String
Usage	Specifies the orientation component of the bidi format specification.
Globalized	Yes
Bidi supported	No

Bidi ordering schema

This property specifies the ordering scheme of the bidi format specification.

Table 113. Bidi ordering schema details

Required	No
Possible values	Implicit Visual
Default	Implicit
Property type	String
Usage	Specifies the ordering scheme of the bidi format specification.
Globalized	Yes
Bidi supported	No

Bidi numeric shaping

This property specifies the numeric shaping component of the bidi format specification.

Table 114. Bidi numeric details

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

Table 114. Bidi numeric details (continued)

Possible values	Nominal National Contextual
Default	Nominal
Property type	String
Usage	Specifies the numeric shaping component of the bidi format specification.
Globalized	Yes
Bidi supported	No

Bidi shaping

This property specifies the shaping component of the bidi format specification.

Table 115. Bidi shaping details

Required	No
Possible values	Nominal Shaped Initial Middle Final Isolated
Default	Nominal
Property type	String
Usage	Specifies the shaping component of the bidi format specification.
Globalized	Yes
Bidi supported	No

Bidi symmetric swapping

This property specifies the symmetric swapping component of the bidi format specification.

Table 116. Bidi symmetric swapping details

Required	No
Possible values	True False
Default	True
Property type	Boolean
Usage	This property specifies the symmetric swapping component of the bidi format specification.
Globalized	Yes
Bidi supported	No

Client

This property is the client number of the SAP system to which the adapter connects.

Table 117. Client details

Required	Yes
Possible values	You can enter a range of values from 000 to 999.
Default	100
Property type	Integer
Usage	When an application attempts to log on to the SAP server, the SAP server requires that the application have a Client number associated with it. The Client property value identifies the client (the adapter) that is attempting to log onto the SAP server.
Globalized	No
Bidi supported	No

Codepage number

The numeric identifier of the code page.

Table 118. Codepage number details

Required	No
Possible values	You can enter a range of values from 0000 to 9999. For a full listing of languages and associated codepage numbers supported by SAP, access SAP Note 7360.
Default	The default value for this property is conditionally determined by the value set for the Language code property.
Property type	Integer
Usage	The value assigned to the Codepage number defines the code page to be used and has a one-to-one relationship with the value set for the Language code property. The Codepage number establishes a connection to the appropriate language. Each language code value has a codepage number value associated with it. For example, the language code for English, is EN. If you selected EN (English) as your language code, the codepage number is automatically set to the numeric value associated with EN (English). The SAP code page number for EN (English) is 1100.
Example	If Language code is set to JA (Japanese), Codepage number is set to 8000.
Globalized	No
Bidi supported	No

Folder for RFC trace files

This property sets the fully qualified local path to the folder in which to write RFC trace files.

Table 119. Folder for RFC trace files details

Required	No
Default	No default value
Property type	String

Table 119. Folder for RFC trace files details (continued)

Usage	Identifies the fully qualified local path into which RFC trace files are written. If RFC trace on is set to False (not selected), you are not permitted to set a value in the Folder for RFC trace files property. This field cannot be edited if you are modifying existing artifacts
Example	c:\temp\rfcTraceDir
Globalized	Yes
Bidi supported	No

Host name

Specifies the IP address or the name of the application server host that the adapter logs on to.

Table 120. Host name details

Required	Yes (when load balancing is not used).
Default	No default value
Property type	String
Usage	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 on to.
Example	sapServer
Globalized	No
Bidi supported	No

Language code

SAP logon language code.

Table 121. Language code details

Required	Yes
Possible values	Each of the supported languages is preceded by a 2 character language code. The language itself is displayed in parentheses. The language codes that display in the list represent the SAP default set of 41 languages for non Unicode systems plus Arabic. For a full listing of supported language codes and languages, see the SAP documentation.
Default	The default language code will be your current locale. If your current locale is not listed as one of the supported language codes, then a default language code of EN (English) is used.
Property type	String
Usage	If you manually enter a language code, you do not need to enter the language in parentheses.
Example	If the system locale is English, the value for this property is EN (English)
Globalized	No
Bidi supported	No

Log file output location property

This property specifies the location of the log file for external service discovery.

Table 122. Log file output location details

Required	Yes
Default	The .metadata directory of the workspace.
Property type	String
Usage	Use this directory to hold the log file that will list the errors that occur during the discovery process. The type of discovery errors for which logging occurs is controlled by the Logging level property
Example	C:\IBM\wi6.0\workspace\.metadata\SAPMetadataDiscovery.log
Globalized	Yes
Bidi supported	No

Logging level property

This property specifies the type error for which logging will occur during external service.

Table 123. Logging level details

Required	No
Possible values	FATAL SEVERE WARNING AUDIT INFO CONFIG DETAIL
Default	SEVERE
Property type	String
Usage	Use this property to tailor tracing capabilities. By specifying an error type, you are indicating that trace operations will occur only for errors of the type specified.

Table 123. Logging level details (continued)

Example	<p>Accepting the default value of SEVERE will provide trace information on errors that fall into the SEVERE category. Severe errors mean that an operation cannot continue, though the adapter can still function. Severe errors also include error conditions that indicate an impending fatal error, such as reporting on situations that strongly suggest that resources are on the verge of being depleted.</p> <p>Other error descriptions are as follows:</p> <ul style="list-style-type: none"> • Fatal Adapter cannot continue. Adapter cannot function • Warning Potential error or impending error. This also includes conditions that indicate a progressive failure – for example, the potential leaking of resources. • Audit Significant event affecting adapter state or resources • Info General information outlining overall operation progress. • Config Configuration change or status. • Detail General information detailing operation progress
Globalized	Yes
Bidi supported	No

Password

This property is the password of the user account of the adapter on the SAP application server.

Table 124. Password details

Required	Yes
Default	No default value
Property type	String
Usage	<p>The restrictions on the password depend on the version of SAP Web Application Server.</p> <ul style="list-style-type: none"> • For SAP Web Application Server version 6.40 or earlier, the password: <ul style="list-style-type: none"> – Must be uppercase – Must be 8 characters in length • For versions of SAP Web Application Server later than 6.40, the password: <ul style="list-style-type: none"> – Is not case-sensitive – Can be up to 40 characters in length
Globalized	No
Bidi supported	Yes

RFC trace level

This property specifies the global trace level.

Table 125. RFC trace level details

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

Table 125. RFC trace level details (continued)

Possible values	0 - No error 1 - Errors and warnings 2 - Execution path, errors and warnings 3 - Full Execution path, errors and warnings 4 - Execution path, info messages, errors and warnings 6 - Full execution path, info messages, errors and warnings 7 - Debug messages, full execution path, info messages, errors and warnings 8 - Verbose debug messages, full execution path, info messages, errors and warnings
Default	1
Property type	Integer
Usage	If RFC trace on is set to False (not selected), you cannot set a value in the RFC trace level property.
Globalized	No
Bidi supported	No

RFC trace on

This property specifies whether to generate a text file detailing the RFC activity for each event listener.

Table 126. RFC trace on details

Required	No
Possible values	True False
Default	False
Property type	Boolean
Usage	A value of True activates tracing, which generates a text file. This file is created in the directory in which the adapter process was started. The file has a prefix of rfx and a file type of trc (for example, rfc03912_02220.trc). Use these text files in a development environment only, because the files can grow rapidly. If RFC trace on is set to False (not selected), you cannot set values in the Folder for RFC trace files or RFC trace level properties.
Example	Examples of the information in the file are RfcCall FUNCTION BAPI_CUSTOMER_GETLIST, followed by the information for the parameters in the interface, or RFC Info rftable, followed by the data from one of the interface tables. The trace file is created in the directory where the adapter process has been started. The trace file has a .trc file extension and the file name will start with the letters rfc followed by a unique identifier. For example, rfc03912_02220.trc.
Globalized	No
Bidi supported	No

SAP interface name

This property indicates whether you are creating business objects for the ALE, BAPI, Advanced event processing, or Query interface for SAP Software.

Table 127. SAP interface name details

Required	Yes
Possible values	<p>For outbound:</p> <ul style="list-style-type: none"> Advanced event processing (AEP) ALE ALE pass-through IDoc BAPI BAPI work unit BAPI result set Query interface for SAP Software (QSS) <p>For inbound:</p> <ul style="list-style-type: none"> Advanced event processing (AEP) ALE ALE pass-through IDoc BAPI
Default	<p>For outbound: BAPI</p> <p>For inbound: ALE</p>
Property type	String
Usage	<p>Specifies the interface used by the adapter.</p> <p>The adapter interacts with the interface to support outbound and or inbound processing by enabling the exchange of data in the form of business objects.</p> <p>This field cannot be edited if you are modifying existing artifacts</p>
Globalized	No
Bidi supported	No

System number

This property is the system number of the SAP application server.

Table 128. System number details

Required	Yes
Possible values	You can enter a range of values from 00 to 99.
Default	00
Property type	Integer
Usage	The system number further identifies the Gateway service.
Globalized	No
Bidi supported	No

User name

This property is the user account for the adapter on the SAP server.

Table 129. User name details

Required	Yes
Default	No default value

Table 129. User name details (continued)

Property type	String
Usage	Maximum length of 12 characters. The user name is not case sensitive. It is recommended that you set up a CPIC user account in the SAP application and that you give this account the necessary privileges to manipulate the data required by the business objects supported by the adapter. For example, if the adapter must perform certain SAP business transactions, the adapter's account in the SAP application must have the permissions set to allow it to perform these transactions.
Example	SapUser
Globalized	Yes
Bidi supported	Yes

Resource adapter properties

The resource adapter properties control the general operation of the adapter. You set the resource adapter properties using the external service wizard when you configure the adapter. After deploying the adapter, use the administrative console to change these properties.

The following properties for logging and tracing are deprecated:

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

The following table lists and describes the resource adapter properties. A more detailed description of each property is provided in the sections that follow the table. For information on how to read the property detail tables in the sections that follow, see "Guide to information about properties" on page 321.

Table 130. Resource adapter properties for Adapter for SAP Software

Property name		Description
In the wizard	In the administrative console	
"Adapter ID (AdapterID)" on page 365	AdapterID	Identifies the adapter instance for PMI events and for logging and tracing.
"Disguise user data as "XXX" in log and trace files (HideConfidentialTrace) " on page 365	HideConfidentialTrace	Specifies whether to disguise potentially sensitive information by writing X strings instead of user data in the log and trace files.
(Not available)	"Enable high availability support (enableHASupport)" on page 366	Specifies if only one instance of the adapter is active, or more than one adapter instance is active at a given time.
(Not available)	LogFileSize	Deprecated
(Not available)	LogFilename	Deprecated
(Not available)	LogNumberOfFiles	Deprecated
(Not available)	TraceFileSize	Deprecated

Table 130. Resource adapter properties for Adapter for SAP Software (continued)

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

Adapter ID (AdapterID)

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

Table 131. Adapter ID details

Required	Yes
Default	001
Property type	String
Usage	<p>This property identifies the adapter instance in the log and trace files, and also helps identify the adapter instance while monitoring adapters. The adapter ID is used with an adapter-specific identifier, SAPRA, to form the component name used by the Log and Trace Analyzer tool. For example, if the adapter ID property is set to 001, the component ID is SAPRA001.</p> <p>If you run multiple instances of the same adapter, ensure that the first eight characters of the adapter ID property are unique for each instance so that you can correlate the log and trace information to a particular adapter instance. By making the first seven characters of an adapter ID property unique, the component ID for multiple instances of that adapter is also unique, allowing you to correlate the log and trace information to a particular instance of an adapter.</p> <p>For example, when you set the adapter ID property of two instances of WebSphere Adapter for SAP Software to 001 and 002. The component IDs for those instances, SAPRA001 and SAPRA002, are short enough to remain unique, enabling you to distinguish them as separate adapter instances. However, instances with longer adapter ID properties cannot be distinguished from each other. If you set the adapter ID properties of two instances to Instance01 and Instance02, you will not be able to examine the log and trace information for each adapter instance because the component ID for both instances is truncated to SAPRAInstance.</p> <p>For inbound processing, the value of this property is set at the resource adapter level. For outbound processing, the value can be set both at the resource adapter level and the managed connection factory level. After you use the external service wizard to configure the adapter for outbound processing, you can set the resource adapter and managed connection factory properties independently. If you use the IBM Integration Designer assembly editor or the administrative console to reset these properties, ensure that you set them consistently, to prevent inconsistent marking of the log and trace entries.</p>
Globalized	Yes
Bidi supported	No

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

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

Table 132. Disguise user data as "XXX" in log and trace files details

Required	No
Possible values	True False
Default	False
Property type	Boolean
Usage	<p>If you set this property to True, the adapter replaces user data with a string of X's when writing to log and trace files.</p> <p>For inbound processing, the value of this property is set at the resource adapter level. For outbound processing, the value can be set both at the resource adapter level and the managed connection factory level. After you use the external service wizard to configure the adapter for outbound processing, you can set the resource adapter and managed connection factory properties independently. If you use the IBM Integration Designer assembly editor or the administrative console to reset these properties, ensure that you set them consistently, to prevent inconsistent marking of the log and trace entries.</p>
Globalized	No
Bidi supported	No

Enable high availability support (enableHASupport)

This property is true by default and appears in the Process Administrative Console. When it is true, all Inbound applications will be started on all the cluster members but only one application will actively receive the events. The rest will be on stand by for fail over. When the property is set to false all the applications deployed in a high availability environment will actively poll.

Note: For HA Active-Active configuration, this property must be set to false in the Process Administrative Console

Activation specification properties for BAPI inbound processing

Activation specification properties hold the inbound event processing configuration information for a message endpoint.

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

You set the activation specification properties using the external service wizard and can change them using the IBM Integration Designer Assembly Editor, or after deployment through the IBM Business Process Manager or WebSphere Enterprise Service Bus Process Administrative Console.

Table 133 on page 367 lists and describes the activation specification properties that apply to both synchronous RFC and asynchronous transactional RFC. Table 134 on page 368 applies only to asynchronous transaction RFC properties that are used for assured-once delivery.

A more detailed description of each property is provided in the sections that follow the tables. For information on how to read the property detail tables in the sections that follow, see "Guide to information about properties" on page 321.

Table 133. Activation specification properties for BAPI inbound processing

Property name		Description
In the wizard	In the administrative console	
"Client" on page 370	Client	The client number of the SAP system to which the adapter connects.
"Codepage number" on page 370	Codepage	Indicates the numeric identifier of the code page.
"Enable Secure Network Connection" on page 371	SnCMode	Indicates whether secure network connection mode is used.
Retry limit for failed events	FailedEventRetryLimit	The number of times the adapter attempts to redeliver an event before marking the event as failed.
"Folder for RFC trace files" on page 373	RfcTracePath	Sets the fully qualified local path to the folder into which the RFC trace files are written.
"Gateway host" on page 373	GatewayHost	The host name of the SAP gateway.
"Gateway service" on page 374	GatewayService	The identifier of the gateway on the gateway host that carries out the RFC services.
"Host name" on page 374	ApplicationServerHost	Specifies the IP address or the name of the application server host that the adapter logs on to.
"Language code" on page 374	Language code	Specifies the Language code in which the adapter logs on to SAP.
"Logon group name" on page 375	Group	An identifier of the name of the group of application server instances that have been defined in transaction SMLG and linked together for logon load balancing.
"Load Balancing" on page 375	loadBalancing	Specifies if your SAP configuration uses load balancing
"Maximum number of retries in case of system connection failure" on page 376	connectionRetryLimit	The adapter will try connecting to the Enterprise Information System (EIS) for a specified number of tries. Select only if you want to reduce the number of connection exceptions in the outbound operation. If selected, adapter will validate the connection for each outbound request.
"Message server host" on page 376	MessageServerHost	Specifies the name of the host on which the message server is running.
"Number of listeners" on page 376	NumberOfListeners	Specifies the number of event listeners that are to be started.
"Partner character set" on page 377	PartnerCharset	Specifies PartnerCharset encoding.
"Password" on page 377	Password	The password of the user account of the adapter on the SAP application server.
"Retry EIS connection on startup" on page 377	RetryConnectionOnStartup	Controls whether the adapter retries the connection to the EIS if it cannot connect at startup
"RFC program ID" on page 378	RfcProgramID	The remote function call identifier under which the adapter registers in the SAP gateway.
"RFC trace level" on page 379	RfcTraceLevel	Specifies the global trace level.

Table 133. Activation specification properties for BAPI inbound processing (continued)

Property name		Description
In the wizard	In the administrative console	
"RFC trace on" on page 379	RfcTraceOn	Specifies whether to generate a text file detailing the RFC activity for each event listener.
"SAP system ID" on page 380	SAPSystemID	Specifies the system ID of the SAP system for which logon load balancing is allowed.
"Secure Network Connection library path" on page 380	SncLib	Specifies the path to the library that provides the secure network connection service.
"Secure Network Connection name" on page 380	SncMyname	Specifies the name of the secure network connection.
"Secure Network Connection partner" on page 381	SncPartnername	Specifies the name of the secure network connection partner.
"Secure Network Connection security level" on page 381	SncQop	Specifies the level of security for the secure network connection.
"System number" on page 381	SystemNumber	The system number of the SAP application server.
"Time between retries in case of system connection failure (milliseconds)" on page 382	connectionRetryInterval	Specifies the time interval between attempts to restart the event listeners.
"User name" on page 382	userName	The user account for the adapter on the SAP server.
"X509 certificate" on page 383	X509cert	Specifies the X509 certificate to be used as the logon ticket.

The properties in the following table applies only to assured-once delivery. When you select assured-once delivery, the transaction ID sent from the SAP server is stored in a data source. You specify information about the data source with these properties.

Table 134. Additional activation specification properties for assured-once delivery

Property name		Description
In the wizard	In the administrative console	
"Assured once-only delivery" on page 369	AssuredOnceDelivery	Specifies whether to provide assured-once delivery for inbound events.
"Auto create event table" on page 369	EP_CreateTable	Indicates whether the adapter should create the event recovery table automatically if it does not already exist.
"Event recovery data source (JNDI) name" on page 371	EP_SchemaName	The schema used for automatically creating the event recovery table.
"Event recovery data source (JNDI) name" on page 371	EP_DataSource_JNDIName	The JNDI name of the data source configured for event recovery.
"Event recovery table name" on page 372	EP_TableName	The name of the event recovery table.
"Password used to connect to event data source" on page 378	EP_Password	The user password for connecting to the database.

Table 134. Additional activation specification properties for assured-once delivery (continued)

Property name		Description
In the wizard	In the administrative console	
"User name used to connect to event data source" on page 383	EP_UserName	The user name for connecting to the database.

Assured once-only delivery

This property specifies whether to provide assured once-only delivery for inbound events.

Table 135. Assured once-only delivery details

Required	No
Default	True
Property type	Boolean
Usage	<p>When this property is set to True, the adapter provides assured once event delivery. This means that each event will be delivered once and only once. A value of False does not provide assured once event delivery, but provides better performance.</p> <p>When this property is set to True, the adapter attempts to store transaction (XID) information in the event store. If it is set to False, the adapter does not attempt to store the information.</p> <p>This property is used only if the export component is transactional. If the export component is not transactional, no transaction can be used, regardless of the value of this property.</p>
Globalized	No
Bidi supported	No

Note: The **Assured once-only delivery** property applies only to asynchronous transactional RFC processing.

Auto create event table

Determines if the event table is created automatically.

Table 136. Auto create event table details

Required	Yes, if Assured once-only event delivery is set to True, No otherwise.
Possible values	True False
Default	True
Property type	Boolean
Usage	<p>This property indicates whether the adapter should create the event recovery table automatically if it does not already exist.</p> <p>In the administrative console, this property is listed as "EP_CreateTable".</p> <p>If you specify a value of True to automatically create the table, you must specify information about the event table (such as the event recovery table name).</p> <p>The value provided in the Event recovery table name property is used to create the table.</p>
Globalized	No

Table 136. Auto create event table details (continued)

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

Note: The **Auto create event table** property applies only to asynchronous transactional RFC processing.

Client

This property is the client number of the SAP system to which the adapter connects.

Table 137. Client details

Required	Yes
Possible values	You can enter a range of values from 000 to 999.
Default	100
Property type	Integer
Usage	When an application attempts to log on to the SAP server, the SAP server requires that the application have a Client number associated with it. The Client property value identifies the client (the adapter) that is attempting to log onto the SAP server.
Globalized	No
Bidi supported	No

Codepage number

The numeric identifier of the code page.

Table 138. Codepage number details

Required	No
Possible values	You can enter a range of values from 0000 to 9999. For a full listing of languages and associated codepage numbers supported by SAP, access SAP Note 7360.
Default	The default value for this property is conditionally determined by the value set for the Language code property.
Property type	Integer
Usage	The value assigned to the Codepage number defines the code page to be used and has a one-to-one relationship with the value set for the Language code property. The Codepage number establishes a connection to the appropriate language. Each language code value has a codepage number value associated with it. For example, the language code for English, is EN. If you selected EN (English) as your language code, the codepage number is automatically set to the numeric value associated with EN (English). The SAP code page number for EN (English) is 1100.
Example	If Language code is set to JA (Japanese), Codepage number is set to 8000.
Globalized	No
Bidi supported	No

Database schema name

This property is the schema used for automatically creating the event recovery table.

Note: In the administrative console, this property is listed as "EP_SchemaName".

Table 139. Database schema name details

Required	No
Default	No default value.
Property type	String
Usage	Specifies the schema name for the database used by the adapters event persistence feature.
Example	ALE_SCHEMA
Globalized	Yes
Bidi supported	No

Note: The **Database schema name** property applies only to asynchronous transactional RFC processing.

Enable Secure Network Connection

This property indicates whether secure network connection mode is enabled.

Table 140. Enable Secure Network Connection details

Required	No
Possible values	0 (off) 1 (on)
Default	0
Property type	String
Usage	Set the value to 1 (on) if you want to use secure network connection. If you set this value to 1, you must also set following properties: <ul style="list-style-type: none">• SncLib• SncMyname• SncPartername• SncQop
Globalized	No
Bidi supported	No

Event recovery data source (JNDI) name

This property is the JNDI name of the data source configured for event recovery.

Note: In the Process Administrative Console, this property is listed as "EP_DataSource_JNDIName".

Table 141. Event recovery data source (JNDI) name details

Required	Yes
Default	No default value.

Table 141. Event recovery data source (JNDI) name details (continued)

Property type	String
Usage	Used in event recovery processing. The data source must be created in Process Administrative Console. The adapter utilizes data source for <i>persisting</i> the event state.
Example	jdbc/DB2
Globalized	No
Bidi supported	No

Note: The **Event recovery data source (JNDI) name** property applies only to asynchronous transactional RFC processing.

Event recovery table name

This property is the name of the event recovery table.

Note: In the administrative console, this property is listed as "EP_TableName".

Table 142. Event recovery table name details

Required	Yes
Default	No default value.
Property type	String
Usage	Used in event recovery processing. Consult database documentation for information on naming conventions. It is recommended that a separate event recovery table is configured for each endpoint. The same data source can be used to hold all of the event recovery tables.
Example	EVENT_TABLE
Globalized	No
Bidi supported	No

Note: The **Event recovery table name** property applies only to asynchronous transactional RFC processing.

Retry limit for failed events (FailedEventRetryLimit)

This property specifies the number of times that the adapter attempts to redeliver an event before marking the event as failed.

Table 143. Retry limit for failed events details

Required	No
Possible values	Integers
Default	5
Property type	Integer

Table 143. Retry limit for failed events details (continued)

Usage	Use this property to control how many times the adapter tries to send an event before marking it as failed. It accepts the following values: Default If this property is not set, the adapter tries five additional times before marking the event as failed. 0 The adapter tries to deliver the event an infinite number of times. When the property is set to 0, the event remains in the event store and the event is never marked as failed. > 0 For integers greater than zero, the adapter retries the specified number of times before marking the event as failed. < 0 For negative integers, the adapter does not retry failed events.
Globalized	No
Bidi supported	No

Folder for RFC trace files

This property sets the fully qualified local path to the folder in which to write RFC trace files.

Table 144. Folder for RFC trace files details

Required	No
Default	No default value
Property type	String
Usage	Identifies the fully qualified local path into which RFC trace files are written. If RFC trace on is set to False (not selected), you are not permitted to set a value in the Folder for RFC trace files property. This field cannot be edited if you are modifying existing artifacts
Example	c:\temp\rfcTraceDir
Globalized	Yes
Bidi supported	No

Gateway host

This property is the Gateway host name. Enter either the IP address or the name of the Gateway host. Consult with your SAP administrator for information on the Gateway host name.

Table 145. Gateway host details

Required	Yes
Default	No default value
Property type	String

Table 145. Gateway host details (continued)

Usage	This property is the host name of the SAP gateway. The gateway enables communication between work processes on the SAP system and external programs. The host identified is used as the gateway for the resource adapter. Maximum length of 20 characters. If the computer name is longer than 20 characters, define a symbolic name in the THOSTS table.
Globalized	No
Bidi supported	No

Gateway service

This property is the identifier of the gateway on the gateway host that carries out the RFC services.

Table 146. Gateway service details

Required	Yes
Default	sapgw00
Property type	String
Usage	These services enable communication between work processes on the SAP server and external programs. The service typically has the format of sapgw00, where 00 is the SAP system number. Maximum of 20 characters.
Globalized	No
Bidi supported	No

Host name

Specifies the IP address or the name of the application server host that the adapter logs on to.

Table 147. Host name details

Required	Yes (when load balancing is not used).
Default	No default value
Property type	String
Usage	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 on to.
Example	sapServer
Globalized	No
Bidi supported	No

Language code

This property specifies the Language code in which the adapter logs on.

Table 148. Language code details

Required	Yes
Possible values	For a full listing of languages and associated codepage numbers supported by SAP, access SAP Note 7360.

Table 148. Language code details (continued)

Default	The default value for the Language code property is based on the system locale.
Property type	String
Usage	Each of the supported languages is preceded by a 2 character language code. The language itself is displayed in parentheses. The language codes that display in the list represent the SAP default set of 41 languages for non Unicode systems plus Arabic. The value you select determines the value of the Codepage number property. If you manually enter a language code, you do not need to enter the language in parentheses.
Example	If the system locale is English, the value for this property is EN (English).
Globalized	No
Bidi supported	No

Logon group name

This property is an identifier for the name of the group of application server instances that have been defined in transaction SMLG and linked together for logon load balancing.

Table 149. Logon group details

Required	Yes (if load balancing is used)
Possible values	Consult SAP documentation for information on creating Logon groups and on calling transaction SMLG.
Default	No default value
Property type	String
Usage	When the adapter is configured for load balancing, this property represents the name of the group of application server instances that have been defined in transaction SMLG and linked together for logon load balancing. Logon load balancing allows for the dynamic distribution of logon connections to application server instances. Maximum of 20 characters. On most SAP systems, the SPACE logon group is reserved by SAP.
Globalized	No
Bidi supported	No

Load Balancing

This property specifies if your SAP configuration uses load balancing

Table 150. Load balancing details

Required	Yes
Possible values	TrueFalse
Default	False
Property type	Boolean
Usage	This value should be set to true if the SAP configuration uses load balancing. If set to true, Message server host, Logon group and SAP System ID need to be specified.

Table 150. Load balancing details (continued)

Globalized	No
Bidi supported	No

Maximum number of retries in case of system connection failure

This property specifies the number of times the adapter tries to restart the event listeners. If the “Retry EIS connection on startup” on page 377 property is set to true it also indicates the maximum number of times the adapter will retry the connection to the EIS if it cannot connect at startup.

Table 151. Maximum number of retries in case of system failure details

Required	No
Possible values	Integers
Default	0
Property type	Integer
Usage	<p>Only positive values are valid.</p> <p>When the adapter encounters an error related to the inbound connection, this property specifies the number of times the adapter tries to restart the connection. A value of 0 indicates an infinite number of retries.</p> <p>Negative values indicate that the adapter will not attempt to establish the inbound connection</p>
Globalized	No
Bidi supported	No

Message server host

This property specifies the name of the host on which the message server is running.

Table 152. Message server host details

Required	Yes (if load balancing is used)
Default	No default value
Property type	String
Usage	<p>This property specifies the name of the host that will inform all the servers (instances) belonging to this SAP system of the existence of the other servers to be used for load balancing.</p> <p>The message server host contains the information about load balancing for RFC clients so that an RFC client can be directed to an appropriate application server.</p>
Example	SAPERP05
Globalized	No
Bidi supported	No

Number of listeners

This property specifies the number of listeners that are started by an event.

Table 153. Number of listeners details

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

Table 153. Number of listeners details (continued)

Default	1
Property type	Integer
Usage	For event sequencing, this property should be set to 1. To improve adapter performance, you can increase the number of listeners. Note: The adapter will not start if the number of listeners is 0
Globalized	No
Bidi supported	No

Partner character set

This property specifies the partner character set encoding.

Table 154. Partner character set details

Required	No
Default	UTF-8
Property type	String
Usage	When an encoding is specified, it is used; otherwise the default encoding is used.
Globalized	No
Bidi supported	No

Password

This property is the password of the user account of the adapter on the SAP application server.

Table 155. Password details

Required	Yes
Default	No default value
Property type	String
Usage	The restrictions on the password depend on the version of SAP Web Application Server. <ul style="list-style-type: none"> • For SAP Web Application Server version 6.40 or earlier, the password: <ul style="list-style-type: none"> – Must be uppercase – Must be 8 characters in length • For versions of SAP Web Application Server later than 6.40, the password: <ul style="list-style-type: none"> – Is not case-sensitive – Can be up to 40 characters in length
Globalized	No
Bidi supported	Yes

Retry EIS connection on startup

This property controls whether the adapter retries the connection to the EIS if it cannot connect at startup. This property is used in conjunction with “Maximum

number of retries in case of system connection failure” on page 376 and “Time between retries in case of system connection failure (milliseconds)” on page 382.

Table 156. Retry EIS connection on startup

Required	No
Possible Values	True False
Default	False
Property type	Boolean
Usage	If the value is true, it indicates that the adapter will retry the connection to EIS if it cannot connect at startup. The values for the following properties have to be specified: <ul style="list-style-type: none"> • “Maximum number of retries in case of system connection failure” on page 376 • “Time between retries in case of system connection failure (milliseconds)” on page 382 If the value is false, it indicates that the adapter will not retry the connection to EIS if it cannot connect at startup.
Globalized	No
Bidi supported	No

Password used to connect to event data source

This property is the user password for connecting to the database.

Note: In the administrative console, this property is listed as "EP_Password".

Table 157. Password to connect to event data source details

Required	Yes
Default	No default value.
Property type	String
Usage	This property specifies the password used by event persistence processing to obtain the database connection from the data source.
Globalized	Yes
Bidi supported	No

Note: The **Password used to connect to event data source** property applies only to asynchronous transactional RFC processing.

RFC program ID

This property is the program identifier under which the adapter registers in the SAP gateway.

Table 158. RFC program ID details

Required	Yes
Possible values	Use the SAP transaction SM59 (Display and Maintain RFC Destinations) to see a list of available RFC program IDs.
Default	No default value.
Property type	String

Table 158. RFC program ID details (continued)

Usage	The adapter registers with the gateway so that listener threads can process events from RFC-enabled functions. This value must match the program ID registered in the SAP application. The maximum length is 64 characters.
Globalized	No
Bidi supported	No

RFC trace level

This property specifies the global trace level.

Table 159. RFC trace level details

Required	No
Possible values	0 - No error 1 - Errors and warnings 2 - Execution path, errors and warnings 3 - Full Execution path, errors and warnings 4 - Execution path, info messages, errors and warnings 6 - Full execution path, info messages, errors and warnings 7 - Debug messages, full execution path, info messages, errors and warnings 8 - Verbose debug messages, full execution path, info messages, errors and warnings
Default	1
Property type	Integer
Usage	If RFC trace on is set to False (not selected), you cannot set a value in the RFC trace level property.
Globalized	No
Bidi supported	No

RFC trace on

This property specifies whether to generate a text file detailing the RFC activity for each event listener.

Table 160. RFC trace on details

Required	No
Possible values	True False
Default	False
Property type	Boolean
Usage	A value of True activates tracing, which generates a text file. This file is created in the directory in which the adapter process was started. The file has a prefix of rfx and a file type of trc (for example, rfx03912_02220.trc). Use these text files in a development environment only, because the files can grow rapidly. If RFC trace on is set to False (not selected), you cannot set values in the Folder for RFC trace files or RFC trace level properties.

Table 160. RFC trace on details (continued)

Example	<p>Examples of the information in the file are RfcCall FUNCTION BAPI_CUSTOMER_GETLIST, followed by the information for the parameters in the interface, or RFC Info rfctable, followed by the data from one of the interface tables.</p> <p>The trace file is created in the directory where the adapter process has been started. The trace file has a .trc file extension and the file name will start with the letters rfc followed by a unique identifier. For example, rfc03912_02220.trc.</p>
Globalized	No
Bidi supported	No

SAP system ID

This property specifies the system ID of the SAP system for which logon load balancing is allowed.

Table 161. SAP system ID details

Required	Yes (when load balancing is used)
Default	No default value
Property type	String
Usage	Value must be three characters
Example	DYL
Globalized	No
Bidi supported	No

Secure Network Connection library path

This property specifies the path to the library that provides the secure network connection service.

Table 162. Secure Network Connection library path details

Required	Yes, if SncMode is set to 1; no otherwise.
Default	No default value
Property type	String
Usage	If the SncMode property is set to 1 (indicating that you are using a secure network connection), specify the path to the library that provides the service.
Example	/WINDOWS/system32/gssapi32.dll
Globalized	No
Bidi supported	No

Secure Network Connection name

This property specifies the name of the secure network connection.

Table 163. Secure Network Connection name details

Required	Yes, if SncMode is set to 1; no otherwise.
Default	No default value
Property type	String

Table 163. Secure Network Connection name details (continued)

Usage	If the SncMode property is set to 1 (indicating that you are using a secure network connection), specify a name for the connection.
Example	DOMAINNAME/USERNAME
Globalized	No
Bidi supported	No

Secure Network Connection partner

This property specifies the name of the secure network connection partner.

Table 164. Secure Network Connection partner details

Required	Yes, if SncMode is set to 1; no otherwise.
Default	No default value
Property type	String
Usage	If the SncMode property is set to 1 (indicating that you are using a secure network connection), specify a name for the connection partner.
Example	CN=sap00.saperpdev, OU=Adapter, O=IBM, C=US
Globalized	No
Bidi supported	No

Secure Network Connection security level

This property specifies the level of security for the secure network connection.

Table 165. Secure Network Connection security level details

Required	Yes, if SncMode is set to 1; no otherwise.
Possible values	1 (Authentication only) 2 (Integrity protection) 3 (Privacy protection) 8 (Use the value from snc/data_protection/use on the application server) 9 (Use the value from snc/data_protection/max on the application server)
Default	3 (Privacy protection)
Property type	String
Usage	If the SncMode property is set to 1 (indicating that you are using a secure network connection), specify a value to indicate the level of security for the connection.
Globalized	No
Bidi supported	No

System number

This property is the system number of the SAP application server.

Table 166. System number details

Required	Yes
Possible values	You can enter a range of values from 00 to 99.

Table 166. System number details (continued)

Default	00
Property type	Integer
Usage	The system number further identifies the Gateway service.
Globalized	No
Bidi supported	No

Time between retries in case of system connection failure (milliseconds)

This property specifies the time interval between attempts to restart the event listeners.

If the “Retry EIS connection on startup” on page 377 property is set to true it also indicates the time interval that the adapter will wait in between attempts to retry the inbound connection to the EIS if it cannot connect at startup.

Table 167. Time between retries in case of system connection failure details

Required	No
Possible Values	Positive Integers
Default	60000
Unit of measure	Milliseconds
Property type	Integer
Usage	When the adapter encounters an error related to the inbound connection, this property specifies the time interval the adapter waits in between attempts to reestablish an inbound connection.
Globalized	No
Bidi supported	No

User name

This property is the user account for the adapter on the SAP server.

Table 168. User name details

Required	Yes
Default	No default value
Property type	String
Usage	<p>Maximum length of 12 characters. The user name is not case sensitive.</p> <p>It is recommended that you set up a CPIC user account in the SAP application and that you give this account the necessary privileges to manipulate the data required by the business objects supported by the adapter. For example, if the adapter must perform certain SAP business transactions, the adapter's account in the SAP application must have the permissions set to allow it to perform these transactions.</p>
Example	SapUser
Globalized	Yes
Bidi supported	Yes

User name used to connect to event data source

This property is the user name for connecting to the database.

Note: In the administrative console, this property is listed as "EP_UserName".

Table 169. User name to connect to event data source details

Required	Yes
Default	No default value.
Property type	String
Usage	User name used by event persistence for getting the database connection from the data source. Consult database documentation for information on naming conventions.
Globalized	Yes
Bidi supported	No

Note: The **User name used to connect to event data source** property applies only to asynchronous transactional RFC processing.

X509 certificate

This property specifies the X509 certificate to be used as the logon ticket.

Table 170. X509 certificate details

Required	No.
Default	No default value
Property type	String
Usage	If the SncMode property is set to 1 (indicating that you are using a secure network connection), you can provide a value for the X509 certificate.
Globalized	No
Bidi supported	No

Activation specification properties for ALE inbound processing

Activation specification properties hold the inbound event processing configuration information for a message endpoint.

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

You set the activation specification properties using the external service wizard and can change them using the IBM Integration Designer Assembly Editor, or after deployment through the IBM Business Process Manager or WebSphere Enterprise Service Bus Process Administrative Console.

The following table lists and describes the activation specification properties for ALE inbound processing. A more detailed description of each property is provided in the sections that follow the table. For information on how to read the property

detail tables in the sections that follow, see http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r5mx/topic/com.ibm.wsadapters.jca.sap.doc/shared/rsha_in_interpret_prop_details.html.

Table 171. Activation specification properties for ALE inbound processing

Property name		Description
In the wizard	In the administrative console	
"Failure code" on page 386	aleFailureCode	Specifies the status code for dispatch failure.
"Failure text" on page 386	aleFailureText	Specifies the descriptive text for dispatch failure.
"ALE packet audit" on page 387	alePacketUpdate	Specifies if the adapter should send ALEAUD per IDoc or per packet (TID)
"Selective update" on page 387	aleSelectiveUpdate	Specifies which IDoc Type and MessageType combinations are to be updated when the adapter is configured to update a standard SAP status code.
"Status message code" on page 388	aleStatusMsgCode	If required, specifies the message code to use when the adapter posts the ALEAUD Message IDoc (ALEAUD01).
"Success code" on page 388	aleSuccessCode	Specifies the success status code for Application Document Posted.
"Success text" on page 389	aleSuccessText	Specifies the descriptive text for successful Application Document Posted.
"ALE update status" on page 390	aleUpdateStatus	Specifies whether an audit trail is required for all message types.
"Assured once-only delivery" on page 390	AssuredOnceDelivery	Specifies whether to provide assured-once delivery for inbound events.
"Auto create event table" on page 391	EP_CreateTable	Indicates whether the adapter should create the event recovery table automatically if it does not already exist.
"Client" on page 391	Client	The client number of the SAP system to which the adapter connects.
"Codepage number" on page 391	Codepage	Indicates the numeric identifier of the code page.
"Event recovery data source (JNDI) name" on page 392	EP_SchemaName	The schema used for automatically creating the event recovery table.
"Enable Secure Network Connection" on page 389	SrcMode	Indicates whether secure network connection mode is used.
"Event recovery data source (JNDI) name" on page 392	EP_DataSource_JNDIName	The JNDI name of the data source configured for event recovery.
"Event recovery table name" on page 393	EP_TableName	The name of the event recovery table.
Retry limit for failed events	FailedEventRetryLimit	The number of times the adapter attempts to redeliver an event before marking the event as failed.
"Folder for RFC trace files" on page 394	RfcTracePath	Sets the fully qualified local path to the folder into which the RFC trace files are written.
"Gateway host" on page 394	GatewayHost	The host name of the SAP gateway.
"Gateway service" on page 394	GatewayService	The identifier of the gateway on the gateway host that carries out the RFC services.
"Host name" on page 395	ApplicationServerHost	Specifies the IP address or the name of the application server host that the adapter logs on to.

Table 171. Activation specification properties for ALE inbound processing (continued)

Property name		Description
In the wizard	In the administrative console	
"IDoc empty tags" on page 395	IDocEmptyTag	Includes empty tags to the unpopulated fields in the IDoc segment, which are sent to a configured endpoint, based on the option selected.
"Ignore IDoc packet errors" on page 396	IgnoreIDocPacketErrors	Determines what the adapter does when it encounters an error while processing the IDoc packet.
"Language code" on page 396	Language code	Specifies the Language code in which the adapter logs on to SAP.
"Logon group name" on page 396	Group	An identifier of the name of the group of application server instances that have been defined in transaction SMLG and linked together for logon load balancing.
"Load Balancing" on page 397	loadBalancing	Specifies if your SAP configuration uses load balancing
"Maximum Number of retries in case of system connection failure" on page 397	connectionRetryLimit	The adapter will try connecting to the Enterprise Information System (EIS) for a specified number of tries. Select only if you want to reduce the number of connection exceptions in the outbound operation. If selected, adapter will validate the connection for each outbound request.
"Message server host" on page 398	MessageServerHost	Specifies the name of the host on which the message server is running.
"Number of listeners" on page 398	NumberOfListeners	Specifies the number of event listeners that are to be started.
"Partner character set" on page 399	PartnerCharset	Specifies PartnerCharset encoding.
"Password" on page 399	Password	The password of the user account of the adapter on the SAP application server.
"Password used to connect to event data source" on page 399	EP_Password	The user password for connecting to the database.
"Retry EIS connection on startup" on page 400	RetryConnectionOnStartup	Controls whether the adapter retries the connection to the EIS if it cannot connect at startup
"RFC program ID" on page 400	RfcProgramID	The remote function call identifier under which the adapter registers in the SAP gateway.
"RFC trace level" on page 401	RfcTraceLevel	Specifies the global trace level.
"RFC trace on" on page 401	RfcTraceOn	Specifies whether to generate a text file detailing the RFC activity for each event listener.
"SAP system ID" on page 402	SAPSystemID	Specifies the system ID of the SAP system for which logon load balancing is allowed.
"Secure Network Connection library path" on page 402	SncLib	Specifies the path to the library that provides the secure network connection service.
"Secure Network Connection name" on page 402	SncMyname	Specifies the name of the secure network connection.
"Secure Network Connection partner" on page 403	SncPartnername	Specifies the name of the secure network connection partner.

Table 171. Activation specification properties for ALE inbound processing (continued)

Property name		Description
In the wizard	In the administrative console	
"Secure Network Connection security level" on page 403	SncQop	Specifies the level of security for the secure network connection.
"System number" on page 403	SystemNumber	The system number of the SAP application server.
"Time between retries in case of system connection failure (milliseconds)" on page 404	connectionRetryInterval	Specifies the time interval between attempts to restart the event listeners.
"Trim ALE Idoc field data" on page 404	trimAleData	Specifies if the leading white spaces are to be trimmed by the adapter before sending it to endpoint.
"User name" on page 404	userName	The user account for the adapter on the SAP server.
"User name used to connect to event data source" on page 405	EP_UserName	The user name for connecting to the database.
"X509 certificate" on page 405	X509cert	Specifies the X509 certificate to be used as the logon ticket.

Failure code

The value entered determines how the adapter updates the SAP failure status code after the ALE module has retrieved an IDoc object for event processing.

Table 172. ALE failure code details

Required	Yes if AleUpdateStatus is set to True; no otherwise
Possible values	68 58
Default	40, 51, 68
Property type	Integer
Usage	<p>Set a value for this property only if you set the value for AleUpdateStatus to True.</p> <p>Specify a value 68 for this property to cause the adapter to update the SAP failure status code after the ALE module has retrieved an IDoc object for event processing. SAP converts this value to 40 (Application Document not created in receiving system).</p> <p>When you set the AleUpdateStatus property to True, the adapter updates a standard SAP status code after the adapter retrieves an IDoc object for event processing. An IDoc that is not successfully sent to the endpoint is considered a failure. You use the ALE failure code property to specify the code used to signify this failure.</p>
Globalized	No
Bidi supported	No

Failure text

The text that displays in the event that an IDoc is not successfully sent to the endpoint.

Table 173. ALE failure text details

Required	Yes if AleUpdateStatus is set to True; no otherwise.
----------	--

Table 173. ALE failure text details (continued)

Possible values	40, 51, 68
Default	68 Error - no further processing. The values in the text boxes change in accordance with the failure codes.
Property type	String
Usage	Use this property only if you set the AleUpdateStatus property to True. The length of the text string cannot exceed 70 characters. When you set the AleUpdateStatus property to True, the adapter updates a standard SAP status code after the adapter retrieves an IDoc object for event processing. An IDoc that is not successfully sent to the endpoint is considered a failure. You use the ALE failure text property to specify the descriptive text used to signify this failure.
Example	ALE Dispatch Failed
Globalized	Yes
Bidi supported	No

ALE packet audit

ALE update per packet indicates whether the adapter should send ALEAUD audit IDocs per packet or per IDoc.

Table 174. ALE packet audit details

Required	No
Default	False (send ALEAUD per IDoc).
Property type	Boolean
Usage	You can enable/disable this property only if AleUpdateStatus is set to True. When you set this property to true, the adapter sends one ALEAUD per IDoc packet which contains confirmations for all IDocs in the packet. When you set this property to false, the adapter sends one ALEAUD for each IDoc received at the adapter.
Globalized	No
Bidi supported	No

Selective update

Specifies which IDoc Type and MessageType combinations are to be updated.

Table 175. ALE selective update details

Required	No
Default	No default value
Property type	String

Table 175. ALE selective update details (continued)

Usage	<p>You can set values for this property only if AleUpdateStatus has been set to True.</p> <p>When you set the AleUpdateStatus property to True, the adapter updates a standard SAP status code after the adapter retrieves an IDoc object for event processing. You use the ALE selective update property to specify which IDoc Type and MessageType combinations are to be updated.</p> <p>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.</p>
Example	<p>The following example illustrates two sets. In the example, MATMAS03 and DEBMAS03 are the IDocs, and MATMAS and DEBMAS are the message types:</p> <p>MATMAS03/MATMAS;DEBMAS03/DEBMAS</p>
Globalized	No
Bidi supported	No

Status message code

This property specifies the message code to use when the adapter posts the ALEAUD01 IDoc with message type ALEAUD.

Table 176. ALE status message code details

Required	No
Possible values	For list of available codes, refer to the SAP table TEDS1.
Default	No default value.
Property type	String
Usage	<ul style="list-style-type: none"> You can set a value for this property only if AleUpdateStatus has been set to True. You must configure this message code in the receiving partner profile on SAP.
Globalized	No
Bidi supported	No

Success code

ALE success code for the successful posting of an IDoc.

Table 177. ALE success code details

Required	Yes if AleUpdateStatus is set to True; no otherwise
Possible values	30, 41, 55
Default	55 - Application document posted. The values in the text boxes change in accordance with the success codes
Property type	Integer

Table 177. ALE success code details (continued)

Usage	<p>Use this property only if you set the AleUpdateStatus property to True.</p> <p>When you set the AleUpdateStatus property to True, the adapter updates a standard SAP status code after the adapter retrieves an IDoc object for event processing. You use the ALE success code property to specify the code for IDoc posted as 53.</p> <p>After the IDoc is sent to the endpoint, the IDoc status remains as 03 (IDoc posted to port) in SAP. After posting the IDoc, the adapter posts the audit IDoc with the current IDoc number and status as 53. SAP converts the current IDoc status to 41 (Application Document Created in Receiving System).</p>
Globalized	No
Bidi supported	No

Enable Secure Network Connection

This property indicates whether secure network connection mode is enabled.

Table 178. Enable Secure Network Connection details

Required	No
Possible values	<p>0 (off)</p> <p>1 (on)</p>
Default	0
Property type	String
Usage	<p>Set the value to 1 (on) if you want to use secure network connection.</p> <p>If you set this value to 1, you must also set following properties:</p> <ul style="list-style-type: none"> • SncLib • SncMyname • SncPartnername • SncQop.
Globalized	No
Bidi supported	No

Success text

Indicates the text that displays when an application document is posted successfully.

Table 179. ALE success text details

Required	Yes if AleUpdateStatus is set to True; no otherwise.
Possible values	30, 41, 55
Default	55 - Application document posted. The values in the text boxes change in accordance with the success codes
Property type	String

Table 179. ALE success text details (continued)

Usage	Use this property only if you set the AleUpdateStatus property to True. The length of the text string cannot exceed 70 characters. When you set the AleUpdateStatus property to True, the adapter updates a standard SAP status code after the adapter retrieves an IDoc object for event processing. You use the ALE success text property to specify the descriptive text used to signify Application Document Posted.
Example	ALE Dispatch OK
Globalized	Yes
Bidi supported	No

ALE update status

This property specifies whether an audit trail is required for all message types.

Table 180. ALE update status details

Required	Yes
Possible values	True False
Default	False
Property type	Boolean
Usage	Set this property to True if you want the adapter to update a standard SAP status code after the ALE module has retrieved an IDoc object for event processing. If you set this value to True, you must also set following properties: <ul style="list-style-type: none"> • AleFailureCode • AleSuccessCode • AleFailureText • AleSuccessText.
Globalized	No
Bidi supported	No

Assured once-only delivery

This property specifies whether to provide assured once-only delivery for inbound events.

Table 181. Assured once-only delivery details

Required	No
Default	True
Property type	Boolean
Usage	When this property is set to True, the adapter provides assured once event delivery. This means that each event will be delivered once and only once . A value of False does not provide assured once event delivery, but provides better performance. When this property is set to True, the adapter attempts to store transaction (XID) information in the event store. If it is set to False, the adapter does not attempt to store the information. This property is used only if the export component is transactional. If the export component is not transactional, no transaction can be used, regardless of the value of this property.

Table 181. Assured once-only delivery details (continued)

Globalized	No
Bidi supported	No

Auto create event table

Determines if the event table is created automatically.

Table 182. Auto create event table details

Required	Yes, if Assured once-only event delivery is set to True, No otherwise.
Possible values	True False
Default	True
Property type	Boolean
Usage	<p>This property indicates whether the adapter should create the event recovery table automatically if it does not already exist.</p> <p>In the administrative console, this property is listed as "EP_CreateTable".</p> <p>If you specify a value of True to automatically create the table, you must specify information about the event table (such as the event recovery table name).</p> <p>The value provided in the Event recovery table name property is used to create the table.</p>
Globalized	No
Bidi supported	No

Client

This property is the client number of the SAP system to which the adapter connects.

Table 183. Client details

Required	Yes
Possible values	You can enter a range of values from 000 to 999.
Default	100
Property type	Integer
Usage	When an application attempts to log on to the SAP server, the SAP server requires that the application have a Client number associated with it. The Client property value identifies the client (the adapter) that is attempting to log onto the SAP server.
Globalized	No
Bidi supported	No

Codepage number

The numeric identifier of the code page.

Table 184. Codepage number details

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

Table 184. Codepage number details (continued)

Possible values	You can enter a range of values from 0000 to 9999. For a full listing of languages and associated codepage numbers supported by SAP, access SAP Note 7360.
Default	The default value for this property is conditionally determined by the value set for the Language code property.
Property type	Integer
Usage	The value assigned to the Codepage number defines the code page to be used and has a one-to-one relationship with the value set for the Language code property. The Codepage number establishes a connection to the appropriate language. Each language code value has a codepage number value associated with it. For example, the language code for English, is EN. If you selected EN (English) as your language code, the codepage number is automatically set to the numeric value associated with EN (English). The SAP code page number for EN (English) is 1100.
Example	If Language code is set to JA (Japanese), Codepage number is set to 8000.
Globalized	No
Bidi supported	No

Database schema name

This property is the schema used for automatically creating the event recovery table.

Note: In the administrative console, this property is listed as "EP_SchemaName".

Table 185. Database schema name details

Required	No
Default	No default value.
Property type	String
Usage	Specifies the schema name for the database used by the adapters event persistence feature.
Example	ALE_SCHEMA
Globalized	Yes
Bidi supported	No

Event recovery data source (JNDI) name

This property is the JNDI name of the data source configured for event recovery.

Note: In the Process Administrative Console, this property is listed as "EP_DataSource_JNDIName".

Table 186. Event recovery data source (JNDI) name details

Required	Yes
Default	No default value.
Property type	String
Usage	Used in event recovery processing. The data source must be created in Process Administrative Console. The adapter utilizes data source for <i>persisting</i> the event state.
Example	jdbc/DB2

Table 186. Event recovery data source (JNDI) name details (continued)

Globalized	No
Bidi supported	No

Event recovery table name

This property is the name of the event recovery table.

Note: In the administrative console, this property is listed as "EP_TableName".

Table 187. Event recovery table name details

Required	Yes
Default	No default value.
Property type	String
Usage	Used in event recovery processing. Consult database documentation for information on naming conventions. It is recommended that a separate event recovery table is configured for each endpoint. The same data source can be used to hold all of the event recovery tables.
Example	EVENT_TABLE
Globalized	No
Bidi supported	No

Retry limit for failed events (FailedEventRetryLimit)

This property specifies the number of times that the adapter attempts to redeliver an event before marking the event as failed.

Table 188. Retry limit for failed events details

Required	No
Possible values	Integers
Default	5
Property type	Integer
Usage	Use this property to control how many times the adapter tries to send an event before marking it as failed. It accepts the following values: Default If this property is not set, the adapter tries five additional times before marking the event as failed. 0 The adapter tries to deliver the event an infinite number of times. When the property is set to 0, the event remains in the event store and the event is never marked as failed. > 0 For integers greater than zero, the adapter retries the specified number of times before marking the event as failed. < 0 For negative integers, the adapter does not retry failed events.
Globalized	No
Bidi supported	No

Folder for RFC trace files

This property sets the fully qualified local path to the folder in which to write RFC trace files.

Table 189. Folder for RFC trace files details

Required	No
Default	No default value
Property type	String
Usage	Identifies the fully qualified local path into which RFC trace files are written. If RFC trace on is set to False (not selected), you are not permitted to set a value in the Folder for RFC trace files property. This field cannot be edited if you are modifying existing artifacts
Example	c:\temp\rfcTraceDir
Globalized	Yes
Bidi supported	No

Gateway host

This property is the Gateway host name. Enter either the IP address or the name of the Gateway host. Consult with your SAP administrator for information on the Gateway host name.

Table 190. Gateway host details

Required	Yes
Default	No default value
Property type	String
Usage	This property is the host name of the SAP gateway. The gateway enables communication between work processes on the SAP system and external programs. The host identified is used as the gateway for the resource adapter. Maximum length of 20 characters. If the computer name is longer than 20 characters, define a symbolic name in the THOSTS table.
Globalized	No
Bidi supported	No

Gateway service

This property is the identifier of the gateway on the gateway host that carries out the RFC services.

Table 191. Gateway service details

Required	Yes
Default	sapgw00
Property type	String

Table 191. Gateway service details (continued)

Usage	These services enable communication between work processes on the SAP server and external programs. The service typically has the format of sapgw00, where 00 is the SAP system number. Maximum of 20 characters.
Globalized	No
Bidi supported	No

Host name

Specifies the IP address or the name of the application server host that the adapter logs on to.

Table 192. Host name details

Required	Yes (when load balancing is not used).
Default	No default value
Property type	String
Usage	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 on to.
Example	sapServer
Globalized	No
Bidi supported	No

IDoc empty tags

This property includes empty tags to the unpopulated fields in the IDoc segment, which are sent to a configured endpoint, based on the option selected.

Table 193. IDoc empty tags

Required	No
Possible Values	BEFORE_AND_AFTER ONLY_BEFORE ALL_UNPOPULATED_FIELDS_SEGMENTS
Default	ONLY_BEFORE
Property type	String
Usage	Use this property to select the following IDoc empty tag options: <ul style="list-style-type: none"> • BEFORE_AND_AFTER - Include empty tags to the unpopulated fields before and after the populated fields within the IDoc segments. • ONLY_BEFORE - Include empty data for the unpopulated fields within an IDoc segment before the populated field. • ALL_UNPOPULATED_FIELDS_SEGMENTS - Include empty tags to the unpopulated fields in all the IDoc segments.
Globalized	No
Bidi supported	No

Ignore IDoc packet errors

Determines whether or not IDoc packet errors are to be ignored.

Table 194. Ignore IDOC packet errors details

Required	No
Possible values	True False
Default	False
Property type	Boolean
Usage	<p>If the adapter encounters an error while processing the IDoc packet, it can behave in two different ways.</p> <ul style="list-style-type: none">• 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. <p>The status of the transaction is marked as INPROGRESS. The adapter log would display the IDoc numbers that failed and you need to resubmit those individual IDocs separately. You need to manually maintain these records in the event recovery table.</p> <p>This property is not used for single IDocs and for non-split IDoc packets.</p>
Globalized	No
Bidi supported	No

Language code

This property specifies the Language code in which the adapter logs on.

Table 195. Language code details

Required	Yes
Possible values	For a full listing of languages and associated codepage numbers supported by SAP, access SAP Note 7360.
Default	The default value for the Language code property is based on the system locale.
Property type	String
Usage	<p>Each of the supported languages is preceded by a 2 character language code. The language itself is displayed in parentheses.</p> <p>The language codes that display in the list represent the SAP default set of 41 languages for non Unicode systems plus Arabic.</p> <p>The value you select determines the value of the Codepage number property.</p> <p>If you manually enter a language code, you do not need to enter the language in parentheses.</p>
Example	If the system locale is English, the value for this property is EN (English).
Globalized	No
Bidi supported	No

Logon group name

This property is an identifier for the name of the group of application server instances that have been defined in transaction SMLG and linked together for logon load balancing.

Table 196. Logon group details

Required	Yes (if load balancing is used)
Possible values	Consult SAP documentation for information on creating Logon groups and on calling transaction SMLG.
Default	No default value
Property type	String
Usage	<p>When the adapter is configured for load balancing, this property represents the name of the group of application server instances that have been defined in transaction SMLG and linked together for logon load balancing.</p> <p>Logon load balancing allows for the dynamic distribution of logon connections to application server instances.</p> <p>Maximum of 20 characters. On most SAP systems, the SPACE logon group is reserved by SAP.</p>
Globalized	No
Bidi supported	No

Load Balancing

This property specifies if your SAP configuration uses load balancing

Table 197. Load balancing details

Required	Yes
Possible values	TrueFalse
Default	False
Property type	Boolean
Usage	This value should be set to true if the SAP configuration uses load balancing. If set to true, Message server host, Logon group and SAP System ID need to be specified.
Globalized	No
Bidi supported	No

Maximum Number of retries in case of system connection failure

This property specifies the number of times the adapter attempts to create a connection to the Enterprise Information System (EIS). The adapter will try connecting to the EIS for the specified number of times. Select only if you want to reduce the number of connection exceptions in the outbound operation. If selected, the adapter will validate the connection for each outbound request.

Table 198. Reset Client details

Required	No
Possible values	Integers
Default	0
Property type	Integer

Table 198. Reset Client details (continued)

Usage	<p>Only positive values are valid.</p> <p>When the adapter encounters an error related to the outbound connection, it retries to establish a physical connection (when physical connection is not established) for the number of times specified for this property with a time delay specified in the property "Time between retries in case of system connection failure (milliseconds)" on page 346.</p> <p>If the value is 0, the adapter does not perform any EIS connection validation and executes the outbound operation.</p> <p>if the value is > 0, then during each request the adapter validates if the EIS connection is active.</p> <ul style="list-style-type: none"> • If the connection is valid the operation is completed. • if connection is invalid, the adapter invalidates the current managed connection and a new managed connection is created (new physical connection)
Globalized	No
Bidi supported	No

Message server host

This property specifies the name of the host on which the message server is running.

Table 199. Message server host details

Required	Yes (if load balancing is used)
Default	No default value
Property type	String
Usage	<p>This property specifies the name of the host that will inform all the servers (instances) belonging to this SAP system of the existence of the other servers to be used for load balancing.</p> <p>The message server host contains the information about load balancing for RFC clients so that an RFC client can be directed to an appropriate application server.</p>
Example	SAPER05
Globalized	No
Bidi supported	No

Number of listeners

This property specifies the number of listeners that are started by an event.

Table 200. Number of listeners details

Required	No
Default	1
Property type	Integer
Usage	<p>For event sequencing, this property should be set to 1.</p> <p>To improve adapter performance, you can increase the number of listeners.</p> <p>Note: The adapter will not start if the number of listeners is 0</p>
Globalized	No

Table 200. Number of listeners details (continued)

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

Partner character set

This property specifies the partner character set encoding.

Table 201. Partner character set details

Required	No
Default	UTF-8
Property type	String
Usage	When an encoding is specified, it is used; otherwise the default encoding is used.
Globalized	No
Bidi supported	No

Password

This property is the password of the user account of the adapter on the SAP application server.

Table 202. Password details

Required	Yes
Default	No default value
Property type	String
Usage	<p>The restrictions on the password depend on the version of SAP Web Application Server.</p> <ul style="list-style-type: none"> • For SAP Web Application Server version 6.40 or earlier, the password: <ul style="list-style-type: none"> – Must be uppercase – Must be 8 characters in length • For versions of SAP Web Application Server later than 6.40, the password: <ul style="list-style-type: none"> – Is not case-sensitive – Can be up to 40 characters in length
Globalized	No
Bidi supported	Yes

Password used to connect to event data source

This property is the user password for connecting to the database.

Note: In the administrative console, this property is listed as "EP_Password".

Table 203. Password to connect to event data source details

Required	Yes
Default	No default value.
Property type	String
Usage	This property specifies the password used by event persistence processing to obtain the database connection from the data source.
Globalized	Yes

Table 203. Password to connect to event data source details (continued)

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

Retry EIS connection on startup

This property controls whether the adapter retries the connection to the EIS if it cannot connect at startup. This property is used in conjunction with “Maximum Number of retries in case of system connection failure” on page 397 and “Time between retries in case of system connection failure (milliseconds)” on page 404.

Table 204. Retry EIS connection on startup

Required	No
Possible Values	True False
Default	False
Property type	Boolean
Usage	If the value is true, it indicates that the adapter will retry the connection to EIS if it cannot connect at startup. The values for the following properties have to be specified: <ul style="list-style-type: none"> • “Maximum Number of retries in case of system connection failure” on page 397 • “Time between retries in case of system connection failure (milliseconds)” on page 404 If the value is false, it indicates that the adapter will not retry the connection to EIS if it cannot connect at startup.
Globalized	No
Bidi supported	No

RFC program ID

This property is the program identifier under which the adapter registers in the SAP gateway.

Table 205. RFC program ID details

Required	Yes
Possible values	Use the SAP transaction SM59 (Display and Maintain RFC Destinations) to see a list of available RFC program IDs.
Default	No default value.
Property type	String
Usage	The adapter registers with the gateway so that listener threads can process events from RFC-enabled functions. This value must match the program ID registered in the SAP application. The maximum length is 64 characters.
Globalized	No
Bidi supported	No

RFC trace level

This property specifies the global trace level.

Table 206. RFC trace level details

Required	No
Possible values	0 - No error 1 - Errors and warnings 2 - Execution path, errors and warnings 3 - Full Execution path, errors and warnings 4 - Execution path, info messages, errors and warnings 6 - Full execution path, info messages, errors and warnings 7 - Debug messages, full execution path, info messages, errors and warnings 8 - Verbose debug messages, full execution path, info messages, errors and warnings
Default	1
Property type	Integer
Usage	If RFC trace on is set to <code>False</code> (not selected), you cannot set a value in the RFC trace level property.
Globalized	No
Bidi supported	No

RFC trace on

This property specifies whether to generate a text file detailing the RFC activity for each event listener.

Table 207. RFC trace on details

Required	No
Possible values	True False
Default	False
Property type	Boolean
Usage	A value of <code>True</code> activates tracing, which generates a text file. This file is created in the directory in which the adapter process was started. The file has a prefix of <code>rfx</code> and a file type of <code>trc</code> (for example, <code>rfc03912_02220.trc</code>). Use these text files in a development environment only, because the files can grow rapidly. If RFC trace on is set to <code>False</code> (not selected), you cannot set values in the Folder for RFC trace files or RFC trace level properties.
Example	Examples of the information in the file are <code>RfcCall FUNCTION BAPI_CUSTOMER_GETLIST</code> , followed by the information for the parameters in the interface, or <code>RFC Info rfctable</code> , followed by the data from one of the interface tables. The trace file is created in the directory where the adapter process has been started. The trace file has a <code>.trc</code> file extension and the file name will start with the letters <code>rfc</code> followed by a unique identifier. For example, <code>rfc03912_02220.trc</code> .
Globalized	No
Bidi supported	No

SAP system ID

This property specifies the system ID of the SAP system for which logon load balancing is allowed.

Table 208. SAP system ID details

Required	Yes (when load balancing is used)
Default	No default value
Property type	String
Usage	Value must be three characters
Example	DYL
Globalized	No
Bidi supported	No

Secure Network Connection library path

This property specifies the path to the library that provides the secure network connection service.

Table 209. Secure Network Connection library path details

Required	Yes, if SncMode is set to 1; no otherwise.
Default	No default value
Property type	String
Usage	If the SncMode property is set to 1 (indicating that you are using a secure network connection), specify the path to the library that provides the service.
Example	/WINDOWS/system32/gssapi32.dll
Globalized	No
Bidi supported	No

Secure Network Connection name

This property specifies the name of the secure network connection.

Table 210. Secure Network Connection name details

Required	Yes, if SncMode is set to 1; no otherwise.
Default	No default value
Property type	String
Usage	If the SncMode property is set to 1 (indicating that you are using a secure network connection), specify a name for the connection.
Example	DOMAINNAME/USERNAME
Globalized	No
Bidi supported	No

Secure Network Connection partner

This property specifies the name of the secure network connection partner.

Table 211. Secure Network Connection partner details

Required	Yes, if SncMode is set to 1; no otherwise.
Default	No default value
Property type	String
Usage	If the SncMode property is set to 1 (indicating that you are using a secure network connection), specify a name for the connection partner.
Example	CN=sap00.saperpdev, OU=Adapter, O=IBM, C=US
Globalized	No
Bidi supported	No

Secure Network Connection security level

This property specifies the level of security for the secure network connection.

Table 212. Secure Network Connection security level details

Required	Yes, if SncMode is set to 1; no otherwise.
Possible values	1 (Authentication only) 2 (Integrity protection) 3 (Privacy protection) 8 (Use the value from snc/data_protection/use on the application server) 9 (Use the value from snc/data_protection/max on the application server)
Default	3 (Privacy protection)
Property type	String
Usage	If the SncMode property is set to 1 (indicating that you are using a secure network connection), specify a value to indicate the level of security for the connection.
Globalized	No
Bidi supported	No

System number

This property is the system number of the SAP application server.

Table 213. System number details

Required	Yes
Possible values	You can enter a range of values from 00 to 99.
Default	00
Property type	Integer
Usage	The system number further identifies the Gateway service.
Globalized	No
Bidi supported	No

Time between retries in case of system connection failure (milliseconds)

This property specifies the time interval between attempts to connect to the Enterprise Information System (EIS).

Table 214. Time between retries in case of system connection failure details

Required	No
Possible Values	Positive Integers
Default	60000
Unit of measure	Milliseconds
Property type	Integer
Usage	When the adapter encounters an error related to the outbound connection, this property specifies the time interval that the adapter waits in between attempts to reestablish an outbound connection. It is disabled by default and is only enabled when the value of "Maximum Number of retries in case of system connection failure" on page 339 is greater than 0.
Globalized	No
Bidi supported	No

Trim ALE Idoc field data

This property specifies if the leading white spaces are trimmed by the adapter before sending it to endpoint.

Table 215. Trim ALE Idoc field data

Required	No
Possible Values	True False
Default	True
Property type	Boolean
Usage	Set the value to True, if you want the leading white spaces to be trimmed by the adapter before sending it to endpoint. By default, the value is set to True. Set the value to False, if you do not want the leading white spaces to be trimmed by the adapter.
Globalized	No
Bidi supported	No

User name

This property is the user account for the adapter on the SAP server.

Table 216. User name details

Required	Yes
Default	No default value
Property type	String

Table 216. User name details (continued)

Usage	Maximum length of 12 characters. The user name is not case sensitive. It is recommended that you set up a CPIC user account in the SAP application and that you give this account the necessary privileges to manipulate the data required by the business objects supported by the adapter. For example, if the adapter must perform certain SAP business transactions, the adapter's account in the SAP application must have the permissions set to allow it to perform these transactions.
Example	SapUser
Globalized	Yes
Bidi supported	Yes

User name used to connect to event data source

This property is the user name for connecting to the database.

Note: In the administrative console, this property is listed as "EP_UserName".

Table 217. User name to connect to event data source details

Required	Yes
Default	No default value.
Property type	String
Usage	User name used by event persistence for getting the database connection from the data source. Consult database documentation for information on naming conventions.
Globalized	Yes
Bidi supported	No

X509 certificate

This property specifies the X509 certificate to be used as the logon ticket.

Table 218. X509 certificate details

Required	No.
Default	No default value
Property type	String
Usage	If the SncMode property is set to 1 (indicating that you are using a secure network connection), you can provide a value for the X509 certificate.
Globalized	No
Bidi supported	No

Activation specification properties for Advanced event processing

Activation specification properties are properties that hold the inbound event processing configuration information for a message endpoint.

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

You set the activation specification properties using the external service wizard and can change them using the IBM Integration Designer Assembly Editor, or after deployment through the IBM Business Process Manager or WebSphere Enterprise Service Bus Process Administrative Console.

The following table lists the activation specification properties for Advanced event inbound processing. A complete description of each property is provided in the sections that follow the table. For information on how to read the property detail tables in the sections that follow, see http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r5mx/topic/com.ibm.wsadapters.jca.sap.doc/shared/rsha_in_interpret_prop_details.html.

Table 219. Activation specification properties for Advanced event processing

Property name		Purpose
In enterprise service wizard	In administrative console	
"Adapter Instance for event filtering (AdapterInstanceEventFilter)" on page 408	AdapterInstanceEventFilter	Identifier that determines whether this adapter instance processes specific events in the event store.
"Assured once-only delivery" on page 409	AssuredOnceDelivery	Specifies whether to provide assured-once delivery for inbound events.
"Client" on page 410	Client	The client number of the SAP system to which the adapter connects.
"Codepage number" on page 410	Codepage	Indicates the numeric identifier of the code page.
"Enable Secure Network Connection" on page 411	SnCMode	Indicates whether secure network connection mode is used.
"Delivery type (DeliveryType)" on page 410	DeliveryType	Determines the order in which events are delivered by the adapter to the export.
"Event types to process (EventTypeFilter)" on page 411	EventTypeFilter	A delimited list of event types that indicates to the adapter which events it should deliver.
Retry limit for failed events	FailedEventRetryLimit	The number of times the adapter attempts to redeliver an event before marking the event as failed.
"Folder for RFC trace files" on page 412	RfcTracePath	Sets the fully qualified local path to the folder into which the RFC trace files are written.
"Gateway host" on page 413	GatewayHost	The host name of the SAP gateway.
"Gateway service" on page 413	GatewayService	The identifier of the gateway on the gateway host that carries out the RFC services.
"Host name" on page 413	ApplicationServerHost	Specifies the IP address or the name of the application server host that the adapter logs on to.
"IDoc empty tags" on page 414	IDocEmptyTag	Includes empty tags to the unpopulated fields in the IDoc segment, which are sent to a configured endpoint, based on the option selected.
"Language code" on page 414	Language code	Specifies the Language code in which the adapter logs on to SAP.

Table 219. Activation specification properties for Advanced event processing (continued)

Property name		Purpose
In enterprise service wizard	In administrative console	
"Logon group name" on page 415	Group	An identifier of the name of the group of application server instances that have been defined in transaction SMLG and linked together for logon load balancing.
"Load Balancing" on page 415	loadBalancing	Specifies if your SAP configuration uses load balancing
"Maximum number of events collected during each poll" on page 415	PollQuantity	The number of events that the adapter delivers to the export during each poll period
"Maximum number of retries in case of system connection failure" on page 416	RetryLimit	The number of times SAP JCo tries to restart its server after an error.
"Message server host" on page 416	MessageServerHost	Specifies the name of the host on which the message server is running.
"Partner character set" on page 416	PartnerCharset	Specifies PartnerCharset encoding.
"Password" on page 417	Password	The password of the user account of the adapter on the SAP application server.
"Retry EIS connection on startup" on page 417	RetryConnectionOnStartup	Controls whether the adapter retries the connection to the EIS if it cannot connect at startup
"RFC trace level" on page 418	RfcTraceLevel	Specifies the global trace level.
"RFC trace on" on page 418	RfcTraceOn	Specifies whether to generate a text file detailing the RFC activity for each event listener.
"SAP system ID" on page 419	SAPSystemID	Specifies the system ID of the SAP system for which logon load balancing is allowed.
"Secure Network Connection library path" on page 419	SncLib	Specifies the path to the library that provides the secure network connection service.
"Secure Network Connection name" on page 420	SncMyname	Specifies the name of the secure network connection.
"Secure Network Connection partner" on page 420	SncPartnername	Specifies the name of the secure network connection partner.
"Secure Network Connection security level" on page 421	SncQop	Specifies the level of security for the secure network connection.
"Stop the adapter when an error is encountered while polling (StopPollingOnError)" on page 421	StopPollingOnError	Specifies whether the adapter stops polling for events when it encounters an error during polling.
"System number" on page 421	SystemNumber	The system number of the SAP application server.
"Time between polling for events (milliseconds)" on page 422	PollPeriod	The length of time that the adapter waits between polling periods
"Time between retries in case of system connection failure (milliseconds)" on page 422	RetryInterval	This property is used by the SAP JCo server for the number of retry attempts made.

Table 219. Activation specification properties for Advanced event processing (continued)

Property name		Purpose
In enterprise service wizard	In administrative console	
"User name" on page 423	userName	The user account for the adapter on the SAP server.
"X509 certificate" on page 423	X509cert	Specifies the X509 certificate to be used as the logon ticket.

Adapter Instance for event filtering (AdapterInstanceEvent Filter)

This property controls whether the adapter instance processes specific events in the event store.

Table 220. Adapter Instance for event filtering details

Required	No
Default	null
Property type	String
Usage	<p>This property helps you migrate from WebSphere Business Integration Adapter for SAP to WebSphere Adapter for SAP Software. WebSphere Business Integration Adapter for SAP allows you to perform load balancing on high-volume event types by allowing multiple adapter instances to process events of the same type. When load balancing is not required, a single adapter instance processes all events of a given type. This property is to enable seamless migration for WBIA customers to JCA for customers who are currently taking advantage of the connectorID filtering.</p> <p>WebSphere Adapter for SAP Software typically does not require load balancing in this way, but supports it so that you can migrate without modifying the database triggers or other mechanisms that write events to the event store.</p> <p>The AdapterInstanceEvent Filter property corresponds to the ConnectorID property of the WebSphere Business Integration Adapter for SAP.</p> <p>To use this feature, the database triggers or other mechanisms that create events in the event store must assign the appropriate value to the ConnectorId column.</p> <p>Table 221 on page 409 shows the interaction between the AdapterInstanceEvent Filter property and the value in the ConnectorId column in the event store.</p> <p>If the EventTypeFilter and AdapterInstanceEvent Filter properties are both set, the adapter processes only events that meet both criteria. That is, it processes only those events whose type is specified in the EventTypeFilter property and whose ConnectorId column matches the AdapterInstanceEvent Filter property.</p>
Example	See Table 221 on page 409.
Globalized	Yes
Bidi supported	Yes

Table 221. Interaction of the AdapterInstanceEvent
Filter property with the ConnectorId column in the event store

AdapterInstanceEvent Filter property	ConnectorId column of an event	Result
null	null	The adapter processes the event.
null	Instance1	The adapter processes the event, because the ConnectorId column is not checked.
Instance1	Instance1	The adapter processes the event.
Instance1	Instance2	The adapter does not process the event, because the instance IDs do not match.
Instance1	null	The adapter does not process the event, because the instance IDs do not match.

Assured once-only delivery

This property specifies whether to provide assured once-only delivery for inbound events.

Table 222. Assured once-only delivery details

Required	Yes
Default	True
Property type	Boolean
Usage	<p>When this property is set to True, the adapter provides assured once event delivery. This means that each event will be delivered once and only once. A value of False does not provide assured once event delivery, but provides better performance.</p> <p>When this property is set to True, the adapter attempts to store transaction (XID) information in the event store. If it is set to False, the adapter does not attempt to store the information.</p> <p>This property is used only if the export component is transactional. If the export component is not transactional, no transaction can be used, regardless of the value of this property.</p>
Globalized	No
Bidi supported	No

Client

This property is the client number of the SAP system to which the adapter connects.

Table 223. Client details

Required	Yes
Possible values	You can enter a range of values from 000 to 999.
Default	100
Property type	Integer
Usage	When an application attempts to log on to the SAP server, the SAP server requires that the application have a Client number associated with it. The Client property value identifies the client (the adapter) that is attempting to log onto the SAP server.
Globalized	No
Bidi supported	No

Client

This property is the client number of the SAP system to which the adapter connects.

Table 224. Client details

Required	Yes
Possible values	You can enter a range of values from 000 to 999.
Default	100
Property type	Integer
Usage	When an application attempts to log on to the SAP server, the SAP server requires that the application have a Client number associated with it. The Client property value identifies the client (the adapter) that is attempting to log onto the SAP server.
Globalized	No
Bidi supported	No

Codepage number

The numeric identifier of the code page.

Table 225. Codepage number details

Required	No
Possible values	You can enter a range of values from 0000 to 9999. For a full listing of languages and associated codepage numbers supported by SAP, access SAP Note 7360.
Default	The default value for this property is conditionally determined by the value set for the Language code property.
Property type	Integer
Usage	The value assigned to the Codepage number defines the code page to be used and has a one-to-one relationship with the value set for the Language code property. The Codepage number establishes a connection to the appropriate language. Each language code value has a codepage number value associated with it. For example, the language code for English, is EN. If you selected EN (English) as your language code, the codepage number is automatically set to the numeric value associated with EN (English). The SAP code page number for EN (English) is 1100.
Example	If Language code is set to JA (Japanese), Codepage number is set to 8000.
Globalized	No
Bidi supported	No

Delivery type (DeliveryType)

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

Table 226. Delivery type details

Required	No
Possible values	ORDERED UNORDERED
Default	ORDERED

Table 226. Delivery type details (continued)

Property type	String
Usage	The following values are supported: <ul style="list-style-type: none"> • ORDERED: The adapter delivers events to the export one at a time. • UNORDERED: The adapter delivers all events to the export at once.
Globalized	No
Bidi supported	No

Enable Secure Network Connection

This property indicates whether secure network connection mode is enabled.

Table 227. Enable Secure Network Connection details

Required	No
Possible values	0 (off) 1 (on)
Default	0
Property type	String
Usage	Set the value to 1 (on) if you want to use secure network connection. If you set this value to 1, you must also set following properties: <ul style="list-style-type: none"> • SncLib • SncMyname • SncPartnername • SncQop
Globalized	No
Bidi supported	No

Event types to process (EventTypeFilter)

This property contains a delimited list of event types that indicates to the adapter which events it should deliver.

Table 228. Event types to process details

Required	No
Possible values	A comma-delimited (,) list of business object types
Default	null
Property type	String
Usage	Events are filtered by business object type . If the property is set, the adapter delivers only those events that are in the list. A value of null indicates that no filter will be applied and that all events will be delivered to the export.

Table 228. Event types to process details (continued)

Example	To receive events related to the Customer and Order business objects only, specify this value: Customer,Order If the EventTypeFilter and AdapterInstanceEvent Filter properties are both set, the adapter processes only events that meet both criteria. That is, it processes only those events whose type is specified in the EventTypeFilter property and whose ConnectorId column matches the AdapterInstanceEvent Filter property.
Globalized	No
Bidi supported	No

Retry limit for failed events (FailedEventRetryLimit)

This property specifies the number of times that the adapter attempts to redeliver an event before marking the event as failed.

Table 229. Retry limit for failed events details

Required	No
Possible values	Integers
Default	5
Property type	Integer
Usage	Use this property to control how many times the adapter tries to send an event before marking it as failed. It accepts the following values: Default If this property is not set, the adapter tries five additional times before marking the event as failed. 0 The adapter tries to deliver the event an infinite number of times. When the property is set to 0, the event remains in the event store and the event is never marked as failed. > 0 For integers greater than zero, the adapter retries the specified number of times before marking the event as failed. < 0 For negative integers, the adapter does not retry failed events.
Globalized	No
Bidi supported	No

Folder for RFC trace files

This property sets the fully qualified local path to the folder in which to write RFC trace files.

Table 230. Folder for RFC trace files details

Required	No
Default	No default value
Property type	String

Table 230. Folder for RFC trace files details (continued)

Usage	Identifies the fully qualified local path into which RFC trace files are written. If RFC trace on is set to False (not selected), you are not permitted to set a value in the Folder for RFC trace files property. This field cannot be edited if you are modifying existing artifacts
Example	c:\temp\rfcTraceDir
Globalized	Yes
Bidi supported	No

Gateway host

This property is the Gateway host name. Enter either the IP address or the name of the Gateway host. Consult with your SAP administrator for information on the Gateway host name.

Table 231. Gateway host details

Required	Yes
Default	No default value
Property type	String
Usage	This property is the host name of the SAP gateway. The gateway enables communication between work processes on the SAP system and external programs. The host identified is used as the gateway for the resource adapter. Maximum length of 20 characters. If the computer name is longer than 20 characters, define a symbolic name in the THOSTS table.
Globalized	No
Bidi supported	No

Gateway service

This property is the identifier of the gateway on the gateway host that carries out the RFC services.

Table 232. Gateway service details

Required	Yes
Default	sapgw00
Property type	String
Usage	These services enable communication between work processes on the SAP server and external programs. The service typically has the format of sapgw00, where 00 is the SAP system number. Maximum of 20 characters.
Globalized	No
Bidi supported	No

Host name

Specifies the IP address or the name of the application server host that the adapter logs on to.

Table 233. Host name details

Required	Yes (when load balancing is not used).
Default	No default value
Property type	String
Usage	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 on to.
Example	sapServer
Globalized	No
Bidi supported	No

IDoc empty tags

This property includes empty tags to the unpopulated fields in the IDoc segment, which are sent to a configured endpoint, based on the option selected.

Table 234. IDoc empty tags

Required	No
Possible Values	BEFORE_AND_AFTER ONLY_POPULATED_FIELDS
Default	ONLY_POPULATED_FIELDS
Property type	String
Usage	Use this property to select the following IDoc empty tag options: <ul style="list-style-type: none"> • BEFORE_AND_AFTER - Include empty tags to the unpopulated fields before and after the populated fields within the IDoc segments. • ONLY_POPULATED_FIELDS - Do not include any empty tags to the unpopulated fields.
Globalized	No
Bidi supported	No

Language code

This property specifies the Language code in which the adapter logs on.

Table 235. Language code details

Required	Yes
Possible values	For a full listing of languages and associated codepage numbers supported by SAP, access SAP Note 7360.
Default	The default value for the Language code property is based on the system locale.
Property type	String
Usage	Each of the supported languages is preceded by a 2 character language code. The language itself is displayed in parentheses. The language codes that display in the list represent the SAP default set of 41 languages for non Unicode systems plus Arabic. The value you select determines the value of the Codepage number property. If you manually enter a language code, you do not need to enter the language in parentheses.
Example	If the system locale is English, the value for this property is EN (English).

Table 235. Language code details (continued)

Globalized	No
Bidi supported	No

Logon group name

This property is an identifier for the name of the group of application server instances that have been defined in transaction SMLG and linked together for logon load balancing.

Table 236. Logon group details

Required	Yes (if load balancing is used)
Possible values	Consult SAP documentation for information on creating Logon groups and on calling transaction SMLG.
Default	No default value
Property type	String
Usage	<p>When the adapter is configured for load balancing, this property represents the name of the group of application server instances that have been defined in transaction SMLG and linked together for logon load balancing.</p> <p>Logon load balancing allows for the dynamic distribution of logon connections to application server instances.</p> <p>Maximum of 20 characters. On most SAP systems, the SPACE logon group is reserved by SAP.</p>
Globalized	No
Bidi supported	No

Load Balancing

This property specifies if your SAP configuration uses load balancing

Table 237. Load balancing details

Required	Yes
Possible values	TrueFalse
Default	False
Property type	Boolean
Usage	This value should be set to true if the SAP configuration uses load balancing. If set to true, Message server host, Logon group and SAP System ID need to be specified.
Globalized	No
Bidi supported	No

Maximum number of events collected during each poll

This property specifies the number of events that the adapter delivers to the export during each poll period.

Table 238. Maximum number of events collected during each poll details

Required	Yes
Default	10

Table 238. Maximum number of events collected during each poll details (continued)

Property type	Integer
Usage	The value must be greater than 0
Globalized	No
Bidi supported	No

Maximum number of retries in case of system connection failure

This property specifies the number of times the SAP JCo tries to restart the server. If the Retry EIS connection on startup property is set to true it also indicates the maximum number of times the adapter will retry the inbound connection to the EIS if it cannot connect at startup.

Table 239. Maximum number of retries in case of system connection failure details

Required	No
Possible values	Integers
Default	0
Property type	Integer
Usage	Only positive values are valid. When the adapter encounters an error related to the inbound connection, this property specifies the number of times the SAP JCo tries to restart the server.
Globalized	No
Bidi supported	No

Message server host

This property specifies the name of the host on which the message server is running.

Table 240. Message server host details

Required	Yes (if load balancing is used)
Default	No default value
Property type	String
Usage	This property specifies the name of the host that will inform all the servers (instances) belonging to this SAP system of the existence of the other servers to be used for load balancing. The message server host contains the information about load balancing for RFC clients so that an RFC client can be directed to an appropriate application server.
Example	SAPERP05
Globalized	No
Bidi supported	No

Partner character set

This property specifies the partner character set encoding.

Table 241. Partner character set details

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

Table 241. Partner character set details (continued)

Default	UTF-8
Property type	String
Usage	When an encoding is specified, it is used; otherwise the default encoding is used.
Globalized	No
Bidi supported	No

Password

This property is the password of the user account of the adapter on the SAP application server.

Table 242. Password details

Required	Yes
Default	No default value
Property type	String
Usage	<p>The restrictions on the password depend on the version of SAP Web Application Server.</p> <ul style="list-style-type: none"> • For SAP Web Application Server version 6.40 or earlier, the password: <ul style="list-style-type: none"> – Must be uppercase – Must be 8 characters in length • For versions of SAP Web Application Server later than 6.40, the password: <ul style="list-style-type: none"> – Is not case-sensitive – Can be up to 40 characters in length
Globalized	No
Bidi supported	Yes

Retry EIS connection on startup

This property controls whether the adapter retries the connection to the EIS if it cannot connect at startup. This property is used in conjunction with “Maximum number of retries in case of system connection failure” on page 416 and “Time between retries in case of system connection failure (milliseconds)” on page 422.

Table 243. Retry EIS connection on startup

Required	No
Possible Values	<p>True</p> <p>False</p>
Default	False
Property type	Boolean
Usage	<p>If the value is true, it indicates that the adapter will retry the connection to EIS if it cannot connect at startup. The values for the following properties have to be specified:</p> <ul style="list-style-type: none"> • “Maximum number of retries in case of system connection failure” on page 416 • “Time between retries in case of system connection failure (milliseconds)” on page 422 <p>If the value is false, it indicates that the adapter will not retry the connection to EIS if it cannot connect at startup.</p>
Globalized	No

Table 243. Retry EIS connection on startup (continued)

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

RFC trace level

This property specifies the global trace level.

Table 244. RFC trace level details

Required	No
Possible values	<ul style="list-style-type: none"> 0 - No error 1 - Errors and warnings 2 - Execution path, errors and warnings 3 - Full Execution path, errors and warnings 4 - Execution path, info messages, errors and warnings 6 - Full execution path, info messages, errors and warnings 7 - Debug messages, full execution path, info messages, errors and warnings 8 - Verbose debug messages, full execution path, info messages, errors and warnings
Default	1
Property type	Integer
Usage	If RFC trace on is set to False (not selected), you cannot set a value in the RFC trace level property.
Globalized	No
Bidi supported	No

RFC trace on

This property specifies whether to generate a text file detailing the RFC activity for each event listener.

Table 245. RFC trace on details

Required	No
Possible values	<ul style="list-style-type: none"> True False
Default	False
Property type	Boolean
Usage	<p>A value of True activates tracing, which generates a text file.</p> <p>This file is created in the directory in which the adapter process was started. The file has a prefix of rfx and a file type of trc (for example, rfc03912_02220.trc).</p> <p>Use these text files in a development environment only, because the files can grow rapidly.</p> <p>If RFC trace on is set to False (not selected), you cannot set values in the Folder for RFC trace files or RFC trace level properties.</p>
Example	<p>Examples of the information in the file are RfcCall FUNCTION BAPI_CUSTOMER_GETLIST, followed by the information for the parameters in the interface, or RFC Info rftable, followed by the data from one of the interface tables.</p> <p>The trace file is created in the directory where the adapter process has been started. The trace file has a .trc file extension and the file name will start with the letters rfc followed by a unique identifier. For example, rfc03912_02220.trc.</p>
Globalized	No

Table 245. RFC trace on details (continued)

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

SAP system ID

This property specifies the system ID of the SAP system for which logon load balancing is allowed.

Table 246. SAP system ID details

Required	Yes (when load balancing is used)
Default	No default value
Property type	String
Usage	Value must be three characters
Example	DYL
Globalized	No
Bidi supported	No

Secure Network Connection library path

This property specifies the path to the library that provides the secure network connection service.

Table 247. Secure Network Connection library path details

Required	Yes, if SncMode is set to 1; no otherwise.
Default	No default value
Property type	String
Usage	If the SncMode property is set to 1 (indicating that you are using a secure network connection), specify the path to the library that provides the service.
Example	/WINDOWS/system32/gssapi32.dll
Globalized	No
Bidi supported	No

Secure Network Connection library path

This property specifies the path to the library that provides the secure network connection service.

Table 248. Secure Network Connection library path details

Required	Yes, if SncMode is set to 1; no otherwise.
Default	No default value
Property type	String
Usage	If the SncMode property is set to 1 (indicating that you are using a secure network connection), specify the path to the library that provides the service.
Example	/WINDOWS/system32/gssapi32.dll
Globalized	No
Bidi supported	No

Secure Network Connection name

This property specifies the name of the secure network connection.

Table 249. Secure Network Connection name details

Required	Yes, if SncMode is set to 1; no otherwise.
Default	No default value
Property type	String
Usage	If the SncMode property is set to 1 (indicating that you are using a secure network connection), specify a name for the connection.
Example	DOMAINNAME/USERNAME
Globalized	No
Bidi supported	No

Secure Network Connection partner

This property specifies the name of the secure network connection partner.

Table 250. Secure Network Connection partner details

Required	Yes, if SncMode is set to 1; no otherwise.
Default	No default value
Property type	String
Usage	If the SncMode property is set to 1 (indicating that you are using a secure network connection), specify a name for the connection partner.
Example	CN=sap00.saperpdev, OU=Adapter, O=IBM, C=US
Globalized	No
Bidi supported	No

Secure Network Connection security level

This property specifies the level of security for the secure network connection.

Table 251. Secure Network Connection security level details

Required	Yes, if SncMode is set to 1; no otherwise.
Possible values	1 (Authentication only) 2 (Integrity protection) 3 (Privacy protection) 8 (Use the value from snc/data_protection/use on the application server) 9 (Use the value from snc/data_protection/max on the application server)
Default	3 (Privacy protection)
Property type	String
Usage	If the SncMode property is set to 1 (indicating that you are using a secure network connection), specify a value to indicate the level of security for the connection.
Globalized	No
Bidi supported	No

Secure Network Connection security level

This property specifies the level of security for the secure network connection.

Table 252. Secure Network Connection security level details

Required	Yes, if SncMode is set to 1; no otherwise.
Possible values	1 (Authentication only) 2 (Integrity protection) 3 (Privacy protection) 8 (Use the value from snc/data_protection/use on the application server) 9 (Use the value from snc/data_protection/max on the application server)
Default	3 (Privacy protection)
Property type	String
Usage	If the SncMode property is set to 1 (indicating that you are using a secure network connection), specify a value to indicate the level of security for the connection.
Globalized	No
Bidi supported	No

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

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

Table 253. Stop the adapter when an error is encountered while polling details

Required	No
Possible values	True False
Default	False
Property type	Boolean
Usage	If this property is set to True, the adapter stops polling when it encounters an error. If this property is set to False, the adapter logs an exception when it encounters an error during polling and continues polling.
Globalized	No
Bidi supported	No

System number

This property is the system number of the SAP application server.

Table 254. System number details

Required	Yes
Possible values	You can enter a range of values from 00 to 99.
Default	00
Property type	Integer
Usage	The system number further identifies the Gateway service.

Table 254. System number details (continued)

Globalized	No
Bidi supported	No

Time between polling for events (milliseconds)

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

Table 255. Time between polling for events (milliseconds)

Required	Yes
Possible values	Integers greater than or equal to 0.
Default	2000
Unit of measure	Milliseconds
Property type	Integer
Usage	The time interval between polling events is established at a fixed rate, which means that if running the poll cycle is delayed for any reason (for example if a prior poll cycle takes longer than expected to complete) the next poll cycle will occur immediately to make up for the lost time caused by the delay.
Globalized	No
Bidi supported	No

Time between retries in case of system connection failure (milliseconds)

This property is used by the SAP JCo server for the number of retry attempts made. If the "Retry EIS connection on startup" on page 417 property is set to true it also indicates the time interval that the adapter will wait in between attempts to retry the inbound connection to the EIS if it cannot connect at startup.

Table 256. Time between retries in case of system connection failure details

Required	No
Possible Values	Positive Integers
Default	60000
Unit of measure	Milliseconds
Property type	Integer
Usage	When the adapter encounters an error related to the inbound connection, this property is used by SAP JCo server.
Globalized	No
Bidi supported	No

User name

This property is the user account for the adapter on the SAP server.

Table 257. User name details

Required	Yes
Default	No default value

Table 257. User name details (continued)

Property type	String
Usage	Maximum length of 12 characters. The user name is not case sensitive. It is recommended that you set up a CPIC user account in the SAP application and that you give this account the necessary privileges to manipulate the data required by the business objects supported by the adapter. For example, if the adapter must perform certain SAP business transactions, the adapter's account in the SAP application must have the permissions set to allow it to perform these transactions.
Example	SapUser
Globalized	Yes
Bidi supported	Yes

User name

This property is the user account for the adapter on the SAP server.

Table 258. User name details

Required	Yes
Default	No default value
Property type	String
Usage	Maximum length of 12 characters. The user name is not case sensitive. It is recommended that you set up a CPIC user account in the SAP application and that you give this account the necessary privileges to manipulate the data required by the business objects supported by the adapter. For example, if the adapter must perform certain SAP business transactions, the adapter's account in the SAP application must have the permissions set to allow it to perform these transactions.
Example	SapUser
Globalized	Yes
Bidi supported	Yes

X509 certificate

This property specifies the X509 certificate to be used as the logon ticket.

Table 259. X509 certificate details

Required	No.
Default	No default value
Property type	String
Usage	If the SncMode property is set to 1 (indicating that you are using a secure network connection), you can provide a value for the X509 certificate.
Globalized	No
Bidi supported	No

Globalization

WebSphere Adapter for SAP Software is a globalized application that can be used in multiple linguistic and cultural environments. Based on character set support and the locale of the host server, the adapter delivers message text in the appropriate language. The adapter supports bidirectional script data transformation between integration components.

Globalization and bidirectional transformation

The adapter is globalized to support single- and multi-byte character sets and deliver message text in the specified language. The adapter also performs bidirectional script data transformation, which refers to the task of processing data that contains both right-to-left (Hebrew or Arabic, for example) and left-to-right (a URL or file path, for example) semantic content within the same file.

Globalization

Globalized software applications are designed and developed for use within multiple linguistic and cultural environments rather than a single environment. WebSphere Adapters, IBM Integration Designer, and IBM Business Process Manager or WebSphere Enterprise Service Bus are written in Java. 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). Therefore, when data is transferred between these integration system components, there is no need for character conversion.

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

Bidirectional script data transformation

Languages such as Arabic and Hebrew are written from right to left, yet they contain embedded segments of text that are written left to right, resulting in bidirectional script. When software applications handle bidirectional script data, standards are used to display and process it. Bidirectional script data transformation applies only to string type data. IBM Business Process Manager or WebSphere Enterprise Service Bus uses the Windows standard format, but applications or file systems exchanging data with the server might use a different format. The adapter transforms bidirectional script data passed between the two systems so that it is accurately processed and displayed on both sides of a transaction. It transforms the script data by using a set of properties that defines the format of script data, as well as properties that identify content or metadata to which transformation applies.

Bidirectional data formats

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. IBM Business Process Manager or WebSphere Enterprise Service Bus use the Windows standard format, but an enterprise information system exchanging data with IBM Business Process Manager 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.

Bidirectional format

IBM Business Process Manager or WebSphere Enterprise Service Bus use the bidirectional format of ILYNN (implicit, left-to-right, on, off, nominal). This is the format used by Windows. If an enterprise information system uses a different format, the adapter converts the format prior to introducing the data to IBM Business Process Manager or WebSphere Enterprise Service Bus.

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

Table 260. Bidirectional format attributes

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

Bidirectional properties that identify data for transformation

To identify business data subject to transformation, set the BiDiContextEIS property. Do this by specifying values for each of the five bidirectional format attributes (listed in the table in the previous section) for the property. The BiDiContextEIS property can be set for the managed connection factory and the activation specification.

To identify event persistence data subject to transformation, set the BiDiFormatEP property. Do this by specifying values for each of the five bidirectional format attributes (listed in the table in the previous section) for the property. The BiDiFormatEP property can be set for the activation specification.

To identify application-specific data for transformation, annotate the BiDiContextEIS property and the BiDiMetadata property within a business object. Do this by using the business object editor within IBM Integration Designer to add the properties as application-specific elements of a business object.

Properties enabled for bidirectional data transformation

Bidirectional data transformation properties enforce the correct format of bidirectional script data exchanged between an application or file system and integration tools and runtime environments. Once these properties are set, bidirectional script data is correctly processed and displayed in IBM Integration Designer and IBM Business Process Manager or WebSphere Enterprise Service Bus.

Enterprise service discovery connection properties

The following enterprise service discovery connection properties control bidirectional script data transformation.

- UserName
- Password

Managed connection factory properties

The following managed connection properties control bidirectional script data transformation.

- UserName
- Password

Activation specification properties

The following activation specification properties control bidirectional script data transformation.

- UserName
- Password

Fault business objects

The adapter supports business faults, which are exceptions that are anticipated and declared in the outbound service description, or import. Business faults occur at predictable points in a business process, and are caused by a business rule violation or a constraint violation.

Although IBM Business Process Manager or WebSphere Enterprise Service Bus support other types of faults, the adapter generates only business faults, which are called simply *faults* in this documentation. Not all exceptions become faults. Faults are generated for errors that are actionable (that is, errors that can have a recovery action that does not require the termination of the application). For example, the adapter generates a fault when it receives a business object for outbound processing that does not contain the required data or when the adapter encounters certain errors during outbound processing.

Fault business objects

The external service wizard creates a business object for each fault that the adapter can generate. In addition, the wizard creates a WBIFault superset business object, which has the message, errorCode, and primarySetKey attributes as shown in

Figure 121.

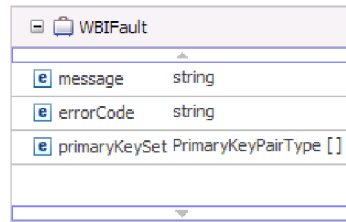


Figure 121. The structure of the WBIFault business object

The WebSphere Adapter for SAP Software enables faults for you. Manual configuration of faults is not required. The adapter provides the following fault business objects that the wizard creates:

- InvalidRequestFault

For a given scenario for one of the SAP outbound interfaces, if the SAP server is unable to process the request, and the SAP server throws errors, the adapter throws this fault. This fault is supported by all outbound interfaces.
- MissingDataFault

If incomplete data is provided, the adapter throws this fault. For example, if the ALE outbound interface has incomplete data to send an IDoc to the SAP server, the adapter throws the MissingDataFault.
- RecordNotFoundFault

During a Retrieve operation, if the record is not found in the SAP server for the input values specified, the adapter throws this fault. For example, for the Query interface for SAP Software Exists and RetrieveAll operations, if no record is found for the provided input, the adapter throws this fault. This fault is supported for the query interface.

The following table shows the faults that are associated with each SAP interface and describes the situation in which each fault is generated.

Table 261. Interfaces and associated faults

Interface	Fault	Cause
Query interface for SAP Software	RecordNotFoundFault	If the adapter does not find any data in SAP for the query, the adapter generates the RecordNotFoundFault.
	InvalidRequestFault	If the SAP server throws a JCo exception, the adapter generates this fault.
BAPI , BAPI work unit, and BAPI result set	InvalidRequestFault	If the SAP server throws a JCo exception, the adapter generates this fault.
Advanced event processing outbound	InvalidRequestFault	If the SAP server throws a JCo exception, the adapter generates this fault.
ALE outbound	MissingDataFault	If incomplete data is provided for a scenario, the adapter generates this fault.
	InvalidRequestFault	If the SAP server throws a JCo exception, the adapter generates this fault.

Adapter messages

View the messages issued by WebSphere Adapter for SAP Software at the following location.

Link to messages: http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r5m1/topic/com.ibm.wbpm.ref.doc/adapt_messages.html

The displayed Web page shows a list of message prefixes. Click a message prefix to see all the messages with that prefix:

- Messages with the prefix CWYAP are issued by WebSphere Adapter for SAP Software
- Messages with the prefix CWYBS are issued by the adapter foundation classes, which are used by all the adapters

Related information

The following information centers, IBM Redbooks, and web pages contain related information for WebSphere Adapter for SAP Software.

Information resources

- The WebSphere Business Process Management information resources web page includes links to articles, Redbooks, documentation, and educational offerings to help you learn about WebSphere Adapters: <http://www-01.ibm.com/software/integration/integration-designer/>.
- The WebSphere Adapters library page includes links to all versions of the documentation: <http://www.ibm.com/software/integration/wbiadapters/library/infocenter/>.

Information about related products

- IBM Business Process Manager, Version 8.0.1, information center, which includes IBM Business Process Manager, and IBM Integration Designer information: <http://pic.dhe.ibm.com/infocenter/dmndhelp/v8r0m1/index.jsp>.
- IBM Business Process Manager, Version 8.0, information center, which includes IBM Business Process Manager, and IBM Integration Designer information: <http://pic.dhe.ibm.com/infocenter/dmndhelp/v8r0mx/index.jsp>.

developerWorks® resources

- WebSphere Adapter Toolkit
- WebSphere business integration zone

Video samples

To help you use WebSphere Adapters, sample videos are available to detail some of the scenarios.

You can find several WebSphere Adapters video demonstrations at this FTP location <ftp://ftp.software.ibm.com/software/websphere/integration/wsa/library/videos/>.

To view these video samples:

1. Download the zip file of the required sample from the FTP location.
2. Extract the downloaded zip file to your system.
3. From the extracted folder, double-click the HTML file.

4. The video sample opens in your default Web browser.

Note: Some of the video samples has a built-in audio narration. If you do not want to listen to the audio, you can lower its tab to the minimum, and slide the tab to an audio level, whenever you want the audio back.

The following sample videos for IBM WebSphere Message Broker Version 7.0 are currently available at this location: <ftp://ftp.software.ibm.com/software/websphere/integration/wsa/library/videos/WMB70>

- Applying Adapter fix for a project
- Applying Adapter fix for IBM WebSphere Message Broker Toolkit
- Applying fix for IBM WebSphere Message Broker runtime
- Checking version in IBM WebSphere Message Broker
- Checking version in IBM WebSphere Message Broker Toolkit
- Enabling service trace in IBM WebSphere Message Broker:

The following sample video for WebSphere Adapters Version 7.5.0.2 is currently available at this location: <ftp://ftp.software.ibm.com/software/websphere/integration/wsa/library/videos/7502/SAP>

- Optional Segment

The following sample videos for IBM Integration Designer are currently available at this location: <ftp://ftp.software.ibm.com/software/websphere/integration/wsa/library/videos/IID>

- Applying Adapter fix for IBM Integration Designer toolkit
- Applying Adapter fix for IBM Process Server at embedded level
- Applying Adapter fix for IBM Process Server at node level
- Checking version in IBM Integration Designer toolkit
- Checking version in IBM Process Server after enabling traces

The following sample videos for WebSphere Adapters version 7.5.0 are currently available at this location: <ftp://ftp.software.ibm.com/software/websphere/integration/wsa/library/videos/750>

- AFC: Scheduling Calendar based polling
- AFC: Overcoming an AFC version conflict situation
- SAP: Monitoring Tivoli ITCAM for a BPM Portfolio requirement
- SAP: Adding random numbers to the end of duplicate field names

The following sample videos for WebSphere Adapters Version 7.0 are currently available at this location: <ftp://ftp.software.ibm.com/software/websphere/integration/wsa/library/videos/700>

- SAP: Discovering an IDoc which has unreleased segments
- SAP: Processing incoming IDoc packets using the ALE passthrough IDoc interface
- SAP: Support for multiple BAPI outbound processing

Notices

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

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

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

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

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

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

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

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

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

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

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

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

IBM Corporation
Department 2Z4A/SOM1
294 Route 100
Somers, NY 10589-0100
U.S.A.

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

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

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

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

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

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

COPYRIGHT LICENSE:

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

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows: (c) (your company name) (year). Portions of

this code are derived from IBM Corp. Sample Programs. (c) Copyright IBM Corp. _enter the year or years_. All rights reserved.

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

Programming interface information

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

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

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

Warning:

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

Trademarks and service marks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com) are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A complete and current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

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

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

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

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

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

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

Index

A

- ABAP handlers
 - creation 78
 - overview 77
- access permissions 295
- activation specification 245, 247
- activation specification properties
 - list of 366, 383, 405
 - setting in administrative console 258, 262
 - setting with external service wizard 184, 200, 209, 217
- Active-Passive 19
- adapter 89
 - configure 89
 - information resources 428
 - messages 428
 - package files 251
 - performance 272
 - project, create 105
 - related information 428
 - samples and tutorials 428
 - support and assistance 428
 - technotes 428
- adapter application
 - starting 264, 296
 - stopping 264, 297
- Adapter for SAP Software module
 - exporting as EAR file 241
 - installing EAR file on server 241
 - starting 264, 296
 - stopping 264, 297
- adapter implementation 13
- adapter log file
 - configuring 269
 - displaying 270
 - truncating 270
- administrative console 243, 245, 247
- Advanced event processing (AEP)
 - interface
 - ABAP handlers 77, 78
 - batch programs 99
 - business objects 84
 - business workflows 101
 - Call Transaction Recorder wizard 79
 - change pointers 102
 - custom triggers 97
 - inbound processing
 - configuring business objects 216
 - overview 80
 - selecting business objects 214
 - setting deployment properties 217
 - managing 265
 - outbound processing
 - configuring business objects 172
 - overview 77
 - selecting business objects 171
 - setting deployment properties 173
 - overview 3, 5, 76

- Advanced event processing (AEP)
 - interface (*continued*)
 - transport files 96
 - WebSphere BI Station tool 265
- Advanced event processing business objects
 - application-specific information 309
 - business-object-level metadata 309
 - metadata 309
 - naming conventions 320
 - operation-level metadata 311
 - operations 314
 - parameters 310
 - property-level metadata 310
- ALE business objects
 - application-specific information 303
 - business-object-level metadata 303
 - IDoc status codes 59, 73
 - metadata 303
 - operation-level metadata 305
 - operations 312, 313
 - parameters 305
 - property-level metadata 305
- ALE failure code property 59, 73, 386
- ALE failure text property 386
- ALE interface
 - business objects
 - metadata 303
 - naming conventions 318
 - structure 60
 - inbound processing
 - configuring business objects 198
 - creating data source 94
 - discovering IDocs from file 193
 - discovering IDocs from system 189
 - error handling 55, 71
 - overview 53, 69
 - selecting business objects 189, 206
 - setting deployment properties 200, 209
 - outbound processing
 - configuring business objects 148
 - discovering IDocs from file 144
 - discovering IDocs from system 141
 - overview 52, 68
 - selecting business objects 141, 154
 - setting deployment properties 149, 155
 - overview 3, 5, 51, 66
- ALE outbound processing
 - configure module 140
- ALE packet audit property 387
- ALE pass-through IDoc interface
 - business objects
 - structure 74
- ALE selective update property 387
- ALE status message code property 388
- ALE success code property 59, 73, 388
- ALE success text property 59, 73, 389

- ALE update status property 59, 73, 390
- ALEAUD IDoc 59, 73
- alias, authentication 104
- application-specific information
 - Advanced event processing business objects 309
 - ALE business objects 303
 - BAPI business objects 299
 - Query interface for SAP Software business objects 306
- Application-specific information (ASI) 299
- archive table 267
- archived events
 - deleting 268
 - displaying 267
 - resubmitting 268
- artifacts 29
- Assured once-only delivery property 42, 56, 72, 369, 390, 409
- authentication 89
 - connection specification
 - properties 10, 110
 - description 14
 - in the wizard 14
 - runtime environment 14
 - authentication alias 89, 104
 - J2C 14
- Auto create event table property
 - description 369, 391
 - prerequisite 94

B

- BAPI business objects
 - business-object-level metadata 299
 - naming conventions 315
 - nested 44
 - operation-level metadata 302
 - operations 311
 - parameters 301
 - property-level metadata 301
 - result set 49
 - simple 42
 - work units 47
- BAPI inbound interface
 - configuring business objects 181
 - overview 5
 - selecting business objects 179
 - setting deployment properties 184
- BAPI interface
 - configuring result set business objects 133
 - configuring simple business objects 115
 - inbound processing 38
 - multiple BAPI 45
 - overview 3, 35, 48
- BAPI outbound interface
 - configuring work unit business objects 124

- BAPI outbound interface *(continued)*
 - outbound processing 36
 - selecting business objects 113
 - setting deployment properties 117, 126, 135
- BAPI result set interface
 - selecting business objects 132
- BAPI result set outbound interface
 - outbound processing 49
- BAPI result sets
 - business object structure 49
 - overview 3, 48
- BAPI work unit interface
 - overview 46
 - selecting business objects 123
- BAPI work unit outbound interface
 - outbound processing 47
- BAPI work unit processing
 - configure module 122
- BAPI work units
 - business-object structure 47
 - overview 3
 - rollback mechanism 48
- Batch Processing 19
- batch programs 99
- BI Station tool 265
- BQPROC field 42, 56, 73
- BQTOTAL field 42, 56, 73
- business faults 426
- business integration adapters to
 - JCA-compliant adapters 26
- business integration concepts 13
- business object 89
- business object information 299
- business objects
 - Advanced event processing interface
 - business object-level metadata 309
 - metadata 309
 - naming conventions 320
 - operation-level metadata 311
 - operations 314
 - property-level metadata 310
 - structure 84
 - ALE interface
 - IDoc status codes 59, 73
 - metadata 303
 - naming conventions 318
 - operations 312, 313
 - structure 60
 - ALE pass-through IDoc interface
 - structure 74
 - BAPI
 - result set 49
 - simple 42
 - work unit 47
 - BAPI interface
 - business-object-level metadata 299
 - metadata 299
 - naming conventions 315
 - operation-level metadata 302
 - operations 311
 - property-level metadata 301
 - fault 426
 - overview 8

- business objects *(continued)*
 - Query interface for SAP Software
 - business-object-level metadata 306
 - metadata 306
 - naming conventions 319
 - operations 313
 - property-level metadata 306
- business workflows 101
- business-object-level metadata
 - Advanced event processing business objects 309
 - ALE business objects 303
 - BAPI business objects 299
 - Query interface for SAP Software
 - business objects 306

C

- Call Transaction Recorder wizard 79
- CEI (Common Event Infrastructure) 275
- chain BAPI
 - BAPI interface 45
- change pointers 102
- changes after migration
 - to the export file 32
 - to the import file 32
 - to the wsdl file 32
- CL (control language) 89
- Client property 325, 337, 357, 370, 391, 409, 410
- cluster level 245, 247
- clustered environment 243, 245, 247
 - adapters version conflict 19
 - deployment 19
 - inbound process 19
 - inbound processes 20
 - load balancing 19
 - outbound process 19
 - outbound processes 20
- Codepage number property 326, 338, 358, 370, 391, 410
- Common Event Infrastructure (CEI) 275
- compatibility matrix 2
- confidential data, disguising 13
- confidential tracing 13
- configuration overview 90
- configuration properties
 - inbound 352
- configure J2C bean and Java data
 - bindings, BAPI interface 113
- configure module 113, 131, 153, 159, 179, 189, 206, 214
- configure module for advanced event processing (inbound) 214
- configure module for advanced event processing (outbound) 170
- configure module for ALE inbound processing 189
- configure module for ALE pass-through IDoc (inbound) 206
- configure module, ALE pass-through IDoc (outbound) 153
- configure module, BAPI interface 113
- configure module, BAPI result set interface 131
- configure module, query interface 159

- configure the SAP system 91
- configuring
 - logging properties 251
 - Performance Monitoring Infrastructure (PMI) 272
- connection properties 111
- connection properties, external service wizard 107
- connector project 105
- control record, IDoc 61
- Create operation 313, 314
- current events queue 265
- custom properties
 - activation specification 258, 262
 - managed connection factory 256, 261
 - resource adapter 254, 260
- Custom retrieve function name property 349
- custom triggers 97

D

- data record, IDoc 61
- data source
 - creating 94
 - JNDI name 94
 - overview 41, 56, 72
 - troubleshooting 94
- database connection, testing 94
- database drivers, location 94
- Database schema name property 371, 392
- debugging
 - self-help resources 279, 298
- definition file, IDoc 95
- Delete operation 313, 314
- deploy module 243, 245
- deployment 241
 - production environment 238
 - test environment 233
- deployment environment 89, 233
- deployment options 15
- developerWorks 428
- disguising confidential data 13
- distribution model 92

E

- EAR file
 - exporting 241
 - installing on server 241
- edit binding
 - import component 225, 227, 229, 230
 - rediscover and reconfigure objects 231
- embedded adapter 243
 - activation specification properties, setting 258
 - considerations for using 17
 - managed connection factory properties, setting 256
 - resource adapter properties, setting 254
 - usage considerations 15
- embedded adapters 243, 245

- embedded adapters (*continued*)
 - changing configuration
 - properties 254
 - setting activation specification
 - properties 258
 - setting managed (J2C) connection
 - factory properties 256
 - setting resource adapter
 - properties 254
- embedded deployment 235
- Enable Secure Network Connection
 - property 344, 371, 389, 411
- enableHASupport property 20
- endpoint applicaiton
 - troubleshoot 294
- endpoints, multiple 54, 70
- EP_CreateTable property
 - description 41, 56, 72, 369, 391
 - prerequisite for using 94
- EP_DataSource_JNDIName
 - property 371, 392
- EP_Password property 378, 399
- EP_SchemaName property 371, 392
- EP_TableName property 372, 393
- EP_UserName property 383, 405
- error handling, event 55, 71
- ErrorCode, setting 281
- ErrorConfiguration, setting 281
- ErrorDetail, setting 281
- ErrorParameter, setting 281
- errors
 - JCo exception errors 292, 293, 294
 - JCo Server could not unmarshall
 - tables 289
 - out-of-memory 289
 - sapxxnn 292, 293, 294
- event delivery 411
- event detection 81
- event processing
 - parsed IDoc packets 57
 - unparsed IDoc packets 58
- event queue
 - current 265
 - future 266
- event recovery 53, 69
- Event recovery data source (JNDI) name
 - property 371, 392
- Event recovery table name property 372, 393
- event recovery table, ALE 56, 72
- event recovery table, BAPI 41
- event restriction 84
- event triggers 82
- event-detection mechanism 96
- EVNTDATA field 42, 56, 73
- EVNTID field 41, 56, 72
- EVNTSTAT field 41, 56, 72
- exception message 291, 293
- Execute operation 313
- Exists operation 314
- export file 8, 89
- exporting module as EAR file 241
- external dependencies, adding 106, 233, 235, 238
- external dependenciesconnector
 - dependencies, adding 233

- external service wizard
 - overview 8
 - properties, connection 322, 354
 - setting connection properties 107

F

- FAQ 295
- faults
 - business objects 426
 - description 426
 - InvalidRequestFault 427
 - MissingDataFault 427
 - RecordNotFoundFault 427
- FFDC (first-failure data capture) 289
- files
 - IDoc definition 95
 - SystemOut.log log file 253
 - trace.log trace file 253
- first-failure data capture (FFDC) 289
- Folders for RFC trace files 326, 339, 358, 373, 394, 412
- Frequently Asked Questions 295
 - WebSphere Adapter for SAP Software 295
- Function name property 349
- future events queue 266

G

- gateway connections, monitoring 271
- Gateway host property 340, 373, 394, 413
- Gateway service property 340, 374, 394, 413
- globalization and bidirectional
 - transformation 424
- globalization and bidirectional
 - transformation, properties 426

H

- HA Active-Active 19
- hardware and software requirements 2
- hardware requirements 2
- High Availability (HA)
 - clustered environments 428
- high-availability environment 19
 - Active-Active 19
 - Active-Passive 19
 - deployment 19
 - inbound processes 20
 - outbound processes 20
- Host name property 327, 341, 359, 374, 395, 413

I

- IBM Business Process Manager
 - information 428
- IBM Business Process Manager or
 - WebSphere Enterprise Service Bus
 - deploying to 238
- IBM Business Process Manager, version
 - 7.5.0.2, information 428

- IBM Integration Designer
 - information 428
 - test environment 233
- IBM WebSphere Adapter for SAP
 - Software
 - administering 251
- IBM WebSphere Adapter Toolkit 428
 - developerWorks resources 428
- IBM WebSphere Enterprise Service Bus
 - information 428
- IDoc definition file 95
- IDoc empty tags 395, 414
- IDoc packets
 - parsed 57
 - unparsed 58
- IDocs
 - control record 61
 - data record 61
 - definition 51, 66
 - inbound processing 53, 69
 - outbound processing 52, 68
 - status codes 59, 73
- Ignore errors in BAPI return
 - property 350
- Ignore IDoc packet errors property 396
- implementation, Java 236
- import component 225, 227, 229, 230
- import file 8, 89
- inbound
 - configuration properties 352
- inbound processing 89
 - Advanced event processing
 - interface 80
 - ALE 53, 69
 - BAPI 38
 - BAPI interface 38
 - overview 3
- installing EAR file 241
- interaction specification properties
 - changing 223
 - Custom retrieve function name 349
 - Function name 349
 - Ignore errors in BAPI return 350
 - Maximum number of hits for the
 - discovery 351
 - Select the queue name 351
 - Use wait parameter before calling
 - BAPI commit 347, 352
- interaction specification property
 - description 348
- interface 89
- InvalidRequestFault 427
- iterative development
 - connection-based editing 225, 227, 229, 230
 - Edit Binding 225, 227, 229, 230
 - import component 225, 227, 229, 230

J

- J2C local transactions 7
- JAR file, adding external 106, 233, 235, 238
- Java 2 security 14
- Java implementation 236
- JCo exception errors 292, 293, 294

JCo Server could not unmarshal tables
error 289
JDBC provider 94

K

knowledge base 296

L

Language code property 327, 341, 359,
374, 396, 414
load balancing 19
Load Balancing 339, 375, 397, 415
local event directory 295
local transactions 7
Log Analyzer 251
log and trace
configure 251
Log and Trace Analyzer, support for 11,
281
log and trace files 11, 281
Log file output location property 328,
360
log files
changing file name 253
disabling 251
enabling 251
level of detail 251
location 253
SystemOut.log 253
logging
configuring properties with
administrative console 251
logging level 251
Logging level property 328, 360
logging options 269
logical system 92
Logon group name property 375, 396,
415

M

manage adapter log file 269
managed (J2C) connection factory
properties
list of 334
setting in administrative console 256,
261
setting with external service
wizard 117, 126, 135, 149, 155, 165,
173
matrix, compatibility 2
Maximum number of events collected
during each poll property 415
Maximum number of events collected
property 415
Maximum number of hits for the
discovery property 351
Maximum number of retries in case of
system connection failure
property 339, 376, 397, 416
Maximum number of retries
property 339, 376, 397, 416
memory-related errors 289

Message server host property 341, 376,
398, 416
messages, adapter 428
metadata

business-object level
Advanced event processing 309
ALE 303
BAPI 299
Query interface for SAP
Software 306
operation-level
Advanced event processing 311
ALE 305
BAPI 302
property-object level
Advanced event processing 310
ALE 305
BAPI 301
Query interface for SAP
Software 306
migration 26
adapter-specific artifacts 29
artifacts
adapter-specific 29
considerations
compatibility with earlier
versions 21
migrating applications from
WebSphere InterChange Server
roadmap 27
overview 27
performing migration 23
upgrading a project 25
WebSphere InterChange Server
migration
migration roadmap 27
WebSphere InterChange Server
migration wizard 29
MissingDataFault 427
modify objects and services
regenerate export component 228
modify service export, BAPI inbound
processing 226
module 89
adding to the server 236
configuring for deployment
overview 89
configuring inbound processing 179
configuring outbound processing 110
deploy for testing 233
monitoring performance 272
multiple connection 411

N

naming business objects 315
naming conventions
Advanced event processing business
objects 320
ALE business objects 318
BAPI business objects 315
Query interface for SAP Software
business objects 319
nested BAPI 44
node level 245, 247
Number of listeners property 376, 398

O

operation-level metadata
Advanced event processing business
objects 311
ALE business objects 305
BAPI business objects 302
operations, supported
Advanced event processing
inbound 314
Advanced event processing
outbound 314
ALE inbound 313
ALE outbound 312
BAPI interface 311
Query interface for SAP
Software 313
out-of-memory errors 289
outbound configuration properties 321
outbound processing 89
Advanced event processing 77
ALE 52, 68
BAPI interface 36
BAPI result set interface 49
BAPI work unit interface 47
connection methods
connection properties 10, 110
overview 3
pass connection properties
dynamically 111
testing the module 237

P

package files for adapters 252
Partner character set property 342, 377,
399, 416
partner profile 93
passive adapter 294
Password property 329, 342, 361, 377,
399, 417
Password to connect to event data source
property 378, 399
performance monitoring
infrastructure 272
configuring 272
performance statistics 274
Performance Monitoring Infrastructure
(PMI)
configuring 272
description 272
viewing performance statistics 274
performance statistics 274
PMI (Performance Monitoring
Infrastructure)
configuring 272
viewing performance statistics 274
prerequisite tasks 91
problem determination
self-help resources 279, 298
process business objects 299
program ID, RFC 92
project 89
project interchange (PI) file
project interchange files 25
projects 25
updating without migrating 25

- properties
 - activation specification 258, 262
 - list of 366, 383, 405
 - setting with external service wizard 184, 200, 209, 217
- configuration properties
 - inbound 352
 - outbound 321
- external service connection 322, 354
- inbound configuration 352
- JNDI 247
- managed (J2C) connection
 - factory 256, 261
 - list of 334
 - setting with external service wizard 117, 126, 135, 149, 155, 165, 173
- outbound configuration 321
- resource adapter 243, 245, 247, 254, 260
 - list of 332, 364
- properties information
 - guide 321, 353
- property-level metadata
 - Advanced event processing business objects 310
 - ALE business objects 305
 - BAPI business objects 301
 - Query interface for SAP Software business objects 306

Q

- qRFC protocol 51, 66
- Query interface for SAP Software
 - configuring business objects 165
 - overview 3
 - selecting business objects 160
 - setting deployment properties 165
- Query interface for SAP Software business objects
 - business-object-level metadata 306
 - naming conventions 319
 - operations 313
 - parameters 306
 - property-level metadata 306
- query interface, errors 283

R

- RAR (resource adapter archive) file
 - description 239
 - installing on server 239
 - versions of 7
- receiver port 92
- recommended fixes 279, 298
- RecordNotFoundFault 427
- Redbooks, IBM WebSphere Adapters 428
- related products, information 428
- remote archive directory 295
- requirements
 - hardware 2
 - software 2
- Reset Client property 342
- resolving selector exception message 291

- resource adapter archive (RAR) file
 - description 239
 - installing on server 239
 - versions of 7
- resource adapter properties
 - list of 332, 364
 - setting in administrative console 254, 260
- result sets, BAPI
 - business object structure 49
 - overview 48
- Retrieve operation 314
- RetrieveAll operation 313
- Retry EIS connection on startup 377, 400, 417
- Retry Interval property 55, 71
- Retry Limit property 55, 71
- RFC program ID
 - description 378, 400
 - registering 92
- RFC trace level 329, 343, 361, 379, 401, 418
- RFC trace on 330, 343, 362, 379, 401, 418
- RFC trace path folder 326, 339, 358, 373, 394, 412
- roadmap
 - configuring the module 89
- roadmap for migrating WebSphere InterChange Server applications 27
- runtime environment
 - deploying EAR file 238

S

- samples 87, 428
- SAP dependencies 287
- SAP dependencies, AEP interface 287
- SAP gateway connections, monitoring 271
- SAP Interface name property 330, 362
- SAP system ID property 344, 380, 402, 419
- sapjco3.jar file 106, 233, 238
- sapxxnn 292, 293, 294
- Secure Network Connection library path
 - property
 - SncLib property 345, 380, 402, 419
- Secure Network Connection name
 - property
 - SncMyname property 345, 380, 402, 420
- Secure Network Connection partner
 - property
 - SncPartnername property 345, 381, 403, 420
- Secure Network Connection security level
 - property
 - SncQop property 346, 381, 403, 420, 421
- security
 - disguising sensitive data 13
 - user authentication 14
- Security
 - access levels 91
- security, Java 2 14
- Select the queue name 351

- self-help resources 279, 298
- sensitive data, disguising 13
- sequence file 295
- service interface queue 91
- setting connection properties 107
- simple BAPI
 - business object structure 42
 - description 35
- Single Sign On (SSO)
 - Enable Single Sign On 348
- Single Sign On Logon ticket 348
- SncMode property 344, 371, 389, 411
- software dependencies, adding
 - external 106, 233, 238
- software requirements 2
- stale connection problem 290
- stand-alone adapter 262
 - considerations for using 17
 - managed connection factory
 - properties, setting 261
 - resource adapter properties, setting 260
 - usage considerations 15
- stand-alone adapters
 - changing configuration
 - properties 260
 - setting activation specification
 - properties 262
 - setting managed (J2C) connection
 - factory properties 261
 - setting resource adapter
 - properties 260
- starting adapter applications 264, 296
- status codes, IDocs 59, 73
- stopping adapter applications 264, 297
- support
 - overview 279
 - plug-in for IBM support
 - assistant 279, 298
 - self-help resources 279, 298
 - web site 279, 298
- supported codepages 289
- supported data operations 311
- System number property 331, 346, 363, 381, 403, 421
- SystemOut.log file 253

T

- target component 235
- technotes 2, 279, 298
- test environment 233
 - adding module to 236
 - deploying to 236
 - testing modules 237
- TID (transaction identifier) 51, 66
- Time between retries in case of system connection failure 346, 382, 404, 422
- Time between retries property 346, 382, 404, 422
- trace files
 - changing file name 253
 - disabling 251
 - enabling 251
 - level of detail 251
 - location 253
 - trace.log 253

- tracing
 - configuring properties with administrative console 251
- transaction identifier (TID) 51, 66
- transport files 96
- tRFC protocol 41, 51, 56, 66, 72
- triggers, event 82
- Trim ALE Idoc field data 404
- troubleshooting
 - data source creation 94
 - overview 279
 - self-help resources 279, 298
- tutorials 87, 428

U

- UNORDERED 411
- Update operation 313, 314
- User name property 331, 347, 363, 382, 404, 422, 423
- User name used to connect to event data source property 383, 405

V

- version conflict exception message 293

W

- WaitOnCommit 347, 352
- WebSphere Adapter for SAP Software
 - overview 1
 - SAP interfaces 35
- WebSphere Adapters, version 6.0,
 - information 428
- WebSphere Adapters, version 6.2.x,
 - information 428
- WebSphere Application Server
 - information 428
- WebSphere business integration adapters 26
- WebSphere Business Integration Adapters
 - information 428
- WebSphere Extended Deployment 19
- wiring components 235
- work units, BAPI
 - business object structure 47
 - overview 46
- wrapper, business object
 - Advanced event processing interface 84
 - ALE 60
 - BAPI 42, 43
 - BAPI result set 49
 - BAPI work unit 47

X

- X509 certificate property 347, 383, 405, 423
- XID field 42, 56, 72



Printed in USA