



IBM Software Group

WebSphere® Process Server V6.0
WebSphere Integration Developer V6.0
WebSphere Adapters

IBM WebSphere Adapter V6.0 for SAP



@business on demand.

© 2006 IBM Corporation
Updated May 9, 2006

This presentation covers the IBM WebSphere Adapter for SAP. Other presentations provide an overview of all the WebSphere Adapters, including installation and deployment of WebSphere Adapters and common function.

Agenda

- SAP Terminology
- IBM WebSphere Adapter for SAP – Overview and Architecture
- Enterprise Service (Metadata) Discovery
- Outbound Operations
- Inbound Operations
- Business Objects
- Transaction and Security
- Problem Determination
- Summary



The agenda for this presentation is shown here. Note that the installation and deployment of the SAP Adapter is covered in a separate presentation common for all WebSphere Adapters.

SAP terminology

This section will cover SAP terminology.

SAP terminology

▪ IDoc (SAP Intermediate Document)

- ▶ IDocs represent SAP business objects as flat structures
- ▶ Proprietary format defined by SAP for business data transmission
- ▶ IDocs are used for asynchronous batch data transmission

▪ SAPJCo:

- ▶ The SAP Java™ Connector (SAPJCo) is a toolkit that allows a Java application to communicate with any SAP System
- ▶ The package supports both, Java to SAP System as well as SAP System to Java calls



Shown on this slide and the next is some common SAP terminology that will help you to better understand a discussion of the SAP adapter.

SAP terminology (cont.)

- **Business Application Program Interfaces (BAPI)**

- ▶ BAPI's are programming interface to access SAP Database from within SAP or other development platforms external to R/3 that support the Remote Function Call (RFC) protocol
- ▶ The main objective of BAPI is to achieve integration between the SAP System and external applications, legacy systems etc.
- ▶ BAPIs are defined in the Business Object Repository (BOR) as methods of SAP Business Objects or SAP Interface Types and enable object-oriented access to Business Components (application components) in the SAP System

- **ALE (Application Link Enabling)**

- ▶ Application Link Enabling (ALE) is an integration interface in SAP's Business Framework Architecture (BFA), a component-based architecture enabling software components from SAP and from other software vendors to communicate and be integrated with each other
- ▶ ALE can integrate business processes between SAP Systems and external applications as well as between SAP Systems. Application systems are loosely coupled in an ALE integrated system and the Data is exchanged asynchronously
- ▶ ALE uses IDoc for data exchange

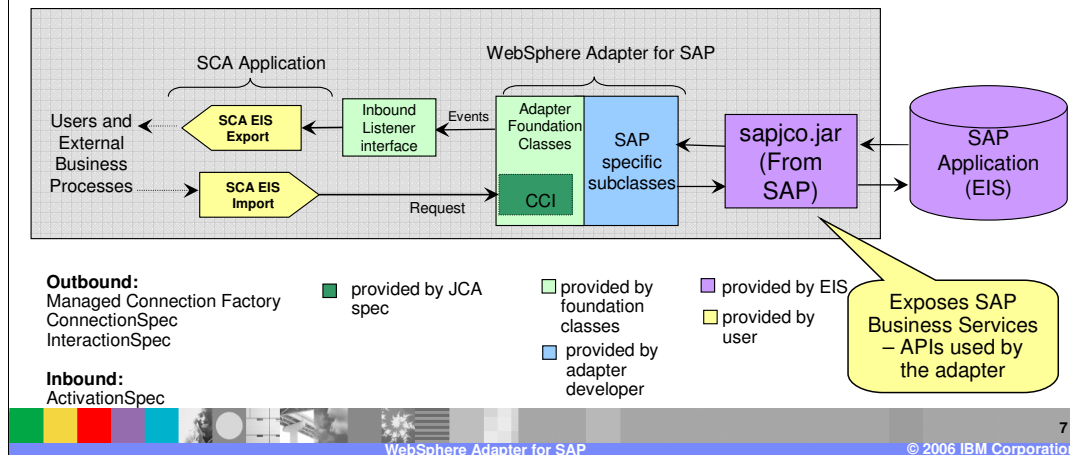
Overview and architecture of IBM WebSphere adapter for SAP



This section provides an overview of the WebSphere Adapter for SAP.

Adapter architecture

- IBM WebSphere Adapter for SAP implements the JCA version 1.5 specification, enabling bi-directional connectivity to and from SAP
- SAP resource adapter communicate with the SAP applications using SAP Java Connector (sapjco) API
- The adapter supports many SAP integration interfaces including BAPI (Business Applications Programming Interface), RFC (Remote Function Modules), and ALE (Application Link Enabling)



This diagram shows the high-level architecture view of the components that play a role in the end to end invocation of the outbound or the inbound request. Using the Enterprise Service Discovery tool in the WebSphere Integration Developer, the SCA EIS components are the associated Business Objects that are created. For the outbound request, the SCA EIS Export component is created, and for the inbound request, the SCA EIS Import component is created.

The SCA Clients interact with the SCA EIS Export and Import components to drive an outbound request or receive an inbound request, as shown in the diagram. The Adapter contains the implementation of JCA specifications and has extensions provided by the Adapter foundation classes. The SCA Export component passes a Business Object wrapped in a J2C CCI Record object. The Adapter extracts the Business Object from the Record object and determines the SAP function to call along with its arguments. The Adapter uses the SAP Component Interface JAR file to communicate with the SAP application.

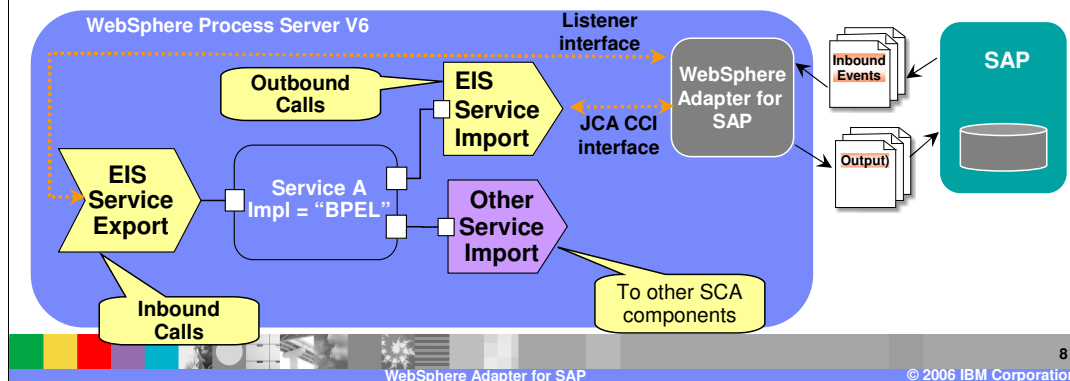
The two main interfaces to a JCA adapter are the Service Provider Interface (SPI) and the Common Client Interface (CCI). The SPI is the application server view of the adapter and contains the contracts necessary to work well with an application server, including Connection creation and matching, security, and work management.

The CCI is designed to provide a common view of data and interaction with the adapter.

The CCI defines the data model and provides a common mechanism to interact with the adapter.

EIS import and export services

- EIS Import (for outbound) and EIS Export (for inbound) SCA components provide a uniform view of the EIS services external to the module
- Business Objects are generated by EMD, and are used by SCA components and the adapter
- This allows components to communicate with the variety of external EIS systems using consistent SCA programming model
- Interacting with the adapters through the use of SCA components and BOs fit the goals and the vision of SOA solutions



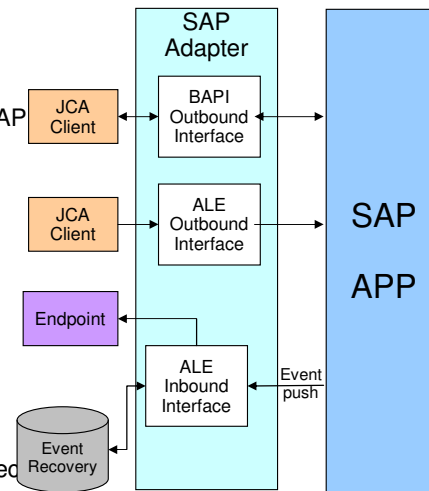
Enterprise Service Discovery tool in WebSphere Integration Developer creates an EIS Import SCA component for an outbound request, and creates EIS Export SCA component for an inbound event request. The Business Objects for the outbound or inbound requests are also created. Using the SCA components for the adapter, they can be wired with other SCA components to create a business application.

In the diagram, the SCA component representing "Service A" is wired with the Adapter EIS Export and EIS Import component. Also shown is the wiring from Service A to other SCA components through the Import. The implementation of Service A could be BPEL or any other support implementation, such as POJO or Human task.

For SCA clients, the adapter functionality is exposed through the EIS Import and EIS Export SCA components.

SAP adapter outbound and inbound support

- Outbound calls supported using the following:
 - BAPI interface
 - ▶ Single BAPI call
 - ▶ BAPI / RFC
 - ▶ Support multiple BAPI calls in a single interaction (BAP Transactions)
 - ALE interface
 - ▶ Single IDoc
 - ▶ IDoc packets (collection of IDOCs)
- Inbound calls with ALE only using Asynchronous event notification – Event Recovery table is used for inbound event management
- SAP functions are modeled as BO
 - ▶ SCA clients wrap the BO within J2C CCI Record object
 - ▶ Metadata in BO used to make SAP function calls and vice versa



The diagram on this page shows the high level flow and the components for the outbound and inbound calls to and from SAP.

The outbound calls are supported with the BAPI interface as well as the ALE interface.

- With BAPI, they can be simple BAPI calls, or BAPI using Remote function call (RFC) or can be multiple BAPI calls in a single interaction, referred to as BAPI transactions. BAPI outbound calls have request and response interaction style.
- The ALE interface supports passing single or multiple IDocs. These are only one-way calls where the IDoc(s) are passed to the SAP application.

The inbound calls are supported by the ALE interface only using Asynchronous event notification. For the inbound call, the Adapter acts as an RfcServer and listens for ALE events from SAP Application. The adapter uses an event recovery table to manage the inbound events.

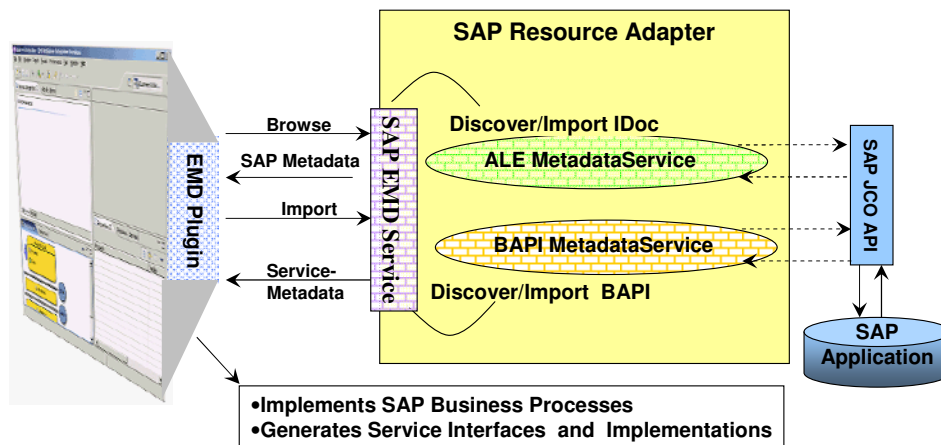
More details on outbound and inbound calls are discussed later in this presentation.

Similar to other WebSphere Adapters, SAP functions are modeled as Business Objects (BOs). The SCA clients wrap the BO with the J2C CCI Record object. The parameters and details of the SAP function call are contained as metadata within the BO.

Enterprise metadata (service) discovery

This section will cover Enterprise Service Discovery.

EMD tool and SAP EMD service interaction



This diagram shows the interaction between the Enterprise Metadata (Service) Discovery (EMD) tool in WebSphere Integration Developer and the EMD support classes within the SAP adapter. It also shows the end to end flow of interaction from the tool to the adapter to the SAP application to create the artifacts for the outbound and inbound requests to and from SAP.

The EMD tool working with the EMD classes in the adapter will do the following:

1. Discover and import SAP metadata definitions and other artifacts related to BAPI and ALE interfaces
2. Generate BO for each method in the business service selected:
 - BOs will contain properties that correspond to the method arguments
 - Application specific information is set in the BOs
3. Generate EIS Import (for outbound) or Export (for inbound) SCA components that can then be wired to other business components and create your business process.
4. Provide connection descriptions for the inbound and outbound calls.

Based on whether BAPI or ALE is selected, the appropriate functions with the adapter are invoked to introspect the SAP application for the available functions.

Using the EMD tool in WebSphere Integration Developer, the developer will introspect the SAP application through the metadata that the adapter will fetch from the SAP application. Once the necessary function and interaction style (outbound or inbound) is selected, the EMD tool will generate the artifacts for the outbound or inbound call, like the EIS SCA export or import components, the Business graph and business objects, the interfaces defined by WSDL and other artifacts needed for SCA clients to send an outbound or receive inbound requests.

SAP outbound operations

This section covers the outbound operations.

Outbound – high level flow for BAPI and ALE

- A SCA component invokes an interaction with SAP
- Business Object (BO) representing the SAP function is passed from the SCA to the Adapter
 - ▶ BO is wrapped in the JCA CCI Record object
 - ▶ BO metadata contains properties that correspond to the method arguments and other application specific information
- Adapter extracts the elements from the BO
 - ▶ From the BO metadata, the Adapter recognizes the SAP interface to use (BAPI or ALE) and the function to call
- Adapter converts the BO data to the appropriate SAP function call
- Adapter makes the SAP function call using the SAPJCo APIs on the destination SAP application, sending the data to SAP



The high level flow of an outbound call using either the BAPI or ALE interface is described here.

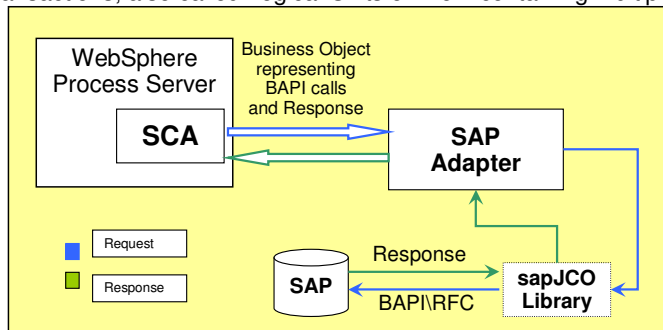
The SAP function call and its parameters and other attributes are modeled using a Business Object (BO). The BOs were generated by the EMD wizard within WebSphere Integration Developer. The SCA client under the covers wraps the BO in a JCA CCI Record object. The Adapter extracts the BO and its metadata from the CCI Record object. It then determines the type of call, BAPI or ALE, and the SAP function name and the function attributes. Using the sapjco APIS the adapter makes the SAP function call to the target SAP application.

SAP outbound operations using BAPI interface

This section covers the outbound operations using the BAPI interface.

Outbound operations using BAPI interface

- BAPIs are SAP standardized Business Application Programming Interfaces that enable third party systems to interact with SAP systems
- BAPI is implemented as an RFC-enabled function
 - Hence, the adapter's BAPI interface can support any RFC-enabled function
- BAPI/RFC interface supports after-image updates only
- Adapter supports the following BAPI calls:
 - Single BAPI calls
 - BAPI transactions, also called Logical Units of Work containing multiple BAPI calls



As indicated earlier when discussing the SAP terminology, the BAPI interface APIs allow the third party systems to interact with SAP.

BAPI/RFC interface supports after-image updates only, meaning that change-summary is not supported. Some adapters support delta objects for create or update whereas the SAP adapter will create the entire business object representing the after image update.

Adapter provides local transaction support for the BAPI interface using the BAPI calls, BAPI_TRANSACTION_COMMIT and BAPI_TRANSACTION_ROLLBACK.

Support for single BAPI call

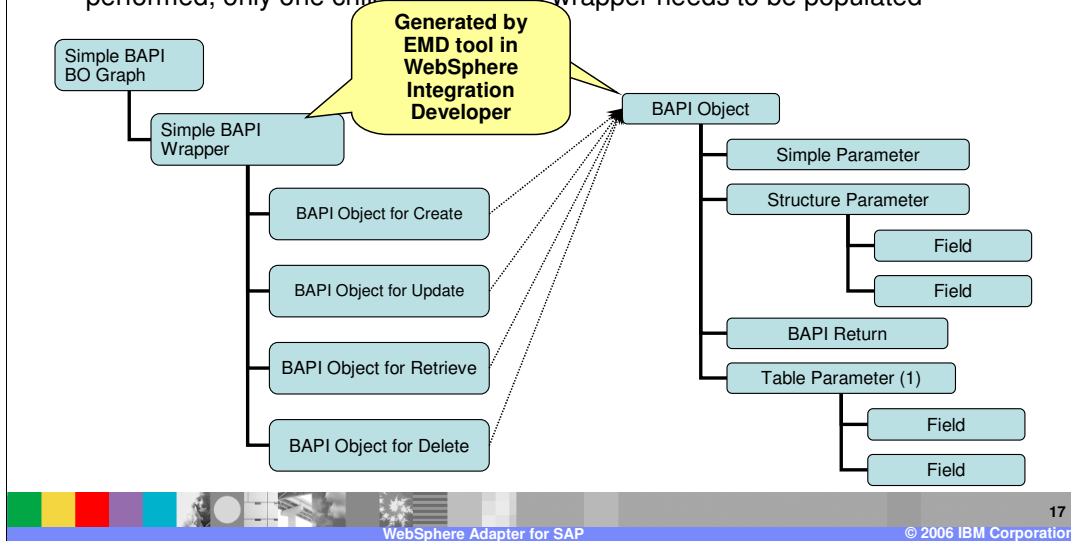
- A single BAPI call is a synchronous blocking call
- Adapter supports single BAPI calls by representing each call with a single Business Object (BO)
- Adapter looks on the metadata and properties in the BO to make the appropriate SAP API function call through the sapjco.jar file
- SAP application executes the function call and returns response data to the adapter
- Adapter converts the response into a BO to be send back to the SCA client component



A single BAPI call is a synchronous blocking call, meaning that the call is blocked until the response is received by the SAP through the adapter. The single BAPI call is represented with a single Business object (BO). As discussed before, the SCA client wraps the BO within the J2C CCI Record object. The adapter extracts the BO and looks at the metadata of the BO to determine the SAP function call. The SAP application executes the function call and returns response data to the adapter. The adapter converts the response to a BO to be send back to the SCA client.

Business object structure for single BAPI call

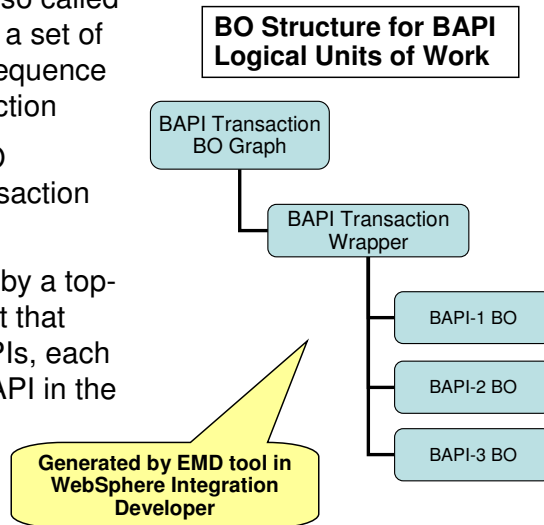
- BAPIs that represent Create, Update, Retrieve, and Delete operations are grouped with a wrapper BO
- BAPIs are children of the wrapper, and, depending on the operation to be performed, only one child object in this wrapper needs to be populated



This diagram shows the Business Object structure for a single BAPI call. These are generated by the EMD tool with the WebSphere Integration Developer. The BAPI calls that represent the operations, namely, create, update, retrieve and delete are grouped and wrapped within a wrapper BO. The BO representing the operation have a structure containing the attributes of the function call, as shown on the right side of the page.

Support of BAPI logical units of work (transaction)

- BAPI Logical Unit of Work (also called BAPI transaction) consists of a set of BAPIs that are executed in sequence to complete the entire transaction
- BAPI transaction wrapper BO represents the complete transaction (sequence of BAPI calls)
- The transaction is supported by a top-level wrapper business object that consists of multiple child BAPIs, each one representing a simple BAPI in the sequence



This page shows the BAPI logical units of work or transactions, where there may be one or more BAPI calls. Each BAPI call is represented by its BO, shown in the diagram as BAPI-1 BO, BAPI-2 BO and so on.

BAPI transaction wrapper object represents the complete transaction. Each second-level child business object represents a structure parameter or table parameter of the method. Simple attributes correspond to simple parameters of the method. The adapter uses the operation (or verb) application specific information (ASI) to determine the sequence of the BAPI calls.

Support of BAPI logical units of work – transaction rollback

- BAPI interface in SAP does not provide a rollback mechanism for transactions
 - ▶ So there is no corresponding BO for BAPI_TRANSACTION_COMMIT
 - ▶ Most BAPI calls commit changes to the SAP system - COMMIT is automatically invoked at the end of every BAPI call (single BAPI or multiple BAPIs)
- Rollback needs to be handled by the business process logic as per one of the following ways:
 - ▶ Terminate all subsequent BAPI calls, thus terminating the transaction before the COMMIT occurs
 - The adapter terminates the transaction when an error is detected
 - If there is no intrinsic COMMIT in the BAPIs that are already invoked, no further steps are required
 - ▶ 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

This page outlines support for BAPI logical units of work.

BAPI outbound supported verbs and operations

- An operation reflects the operation to be performed on the BO by the adapter
- A verb of a BO reflects its state and is definable for after-image objects only

- Supported Verbs
 - ▶ Create
 - ▶ UpdateWithDelete
 - ▶ Delete
- Supported Operations
 - ▶ After-Image Create
 - ▶ After-Image UpdateWithDelete
 - ▶ Retrieve
 - ▶ After-Image Delete



The supported verbs and operations for BAPI outbound calls are listed on this page.

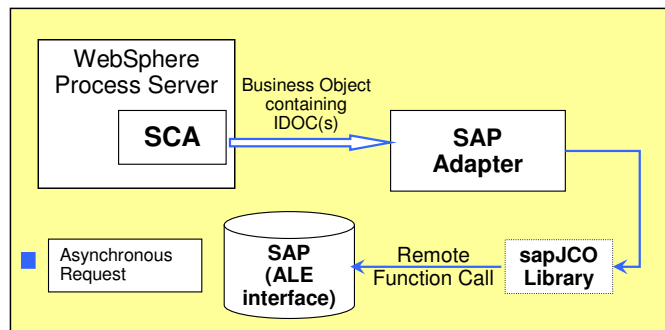
SAP outbound operations using ALE interface



This section covers the outbound operations using the ALE interface.

Outbound operations using ALE interface

- Provides asynchronous interface for sending batch data to SAP applications
 - ▶ Only return code (success or failure) is returned - No return data send back to the adapter
- ALE interface uses IDoc (Intermediate Document) data structures for message exchange
 - ▶ BOs are modeled after the IDoc structure.



The ALE interface provides asynchronous interface for sending batch data represented to SAP applications

The SCA client wraps the SAP IDoc within a Business Object (BO) and then wraps the BO in a J2C CCI Record object.

The adapters extracts the IDoc from the BO and then follows the steps specified below:

1. Using SAP JCo function calls, the adapter converts an IDoc business object to a table format that is compatible with the IDoc format.
2. The adapter then uses Remote Function Calls (RFCs) in the SAP RFC library to establish an RFC connection to the ALE interface and pass the IDoc data to the SAP system.
3. After passing the data to SAP, the adapter releases the connection to SAP.
4. Because the ALE interface is asynchronous, SAP returns a return code only and a null object to the caller. When no exceptions are raised, the outbound transaction is considered successful. The success of the transaction can be verified by inspecting the IDocs that have been generated in SAP.

ALE outbound supported verbs and operations

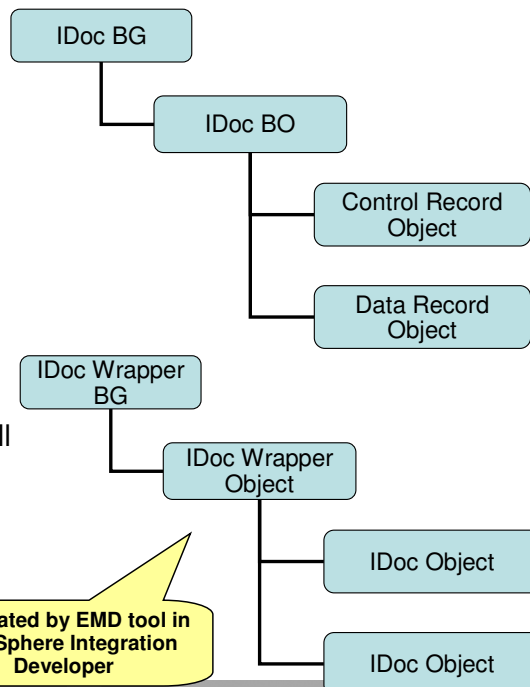
- Supported Verbs - None
- Supported Operations – Execute
 - ▶ CUD operations do not make sense for ALE as this is a batch oriented asynchronous interface
 - ▶ Receiving SAP application processes the data and performs appropriate operations based on the data contained in the control record child object of the IDoc



For the ALE outbound call, all the information is in the IDoc, so there is no supported verb, and the only supported operation is execute on the IDoc.

BO structure

- Control Record Child Object contains the metadata required by the resource adapter to process the business object
- Data Record Child Object contains the actual business object data to be processed by the SAP application, and the metadata required for the resource adapter to convert it to an IDoc structure for the RFC call
- Adapter provides support for IDoc packets using a wrapper business object



Generated by EMD tool in
WebSphere Integration
Developer

24

WebSphere Adapter for SAP

© 2006 IBM Corporation

The diagram on the top left of the page shows the Business graph and Business object generated for single IDoc. The IDoc BO contains the Control Record and Data Record objects. The Control Record object contains the metadata required by the SAP adapter to process the business object.

The diagram object on the bottom left of the page shows the support for multiple IDoc packets, where each IDoc is represented by the IDoc BO.

All these Business graph and business objects are generated by the EMD tool in WebSphere Integration Developer.

***SAP inbound operations
(supported in ALE interface only)***



This section covers the inbound operations, which are supported only in the ALE interface.

Inbound operations: High level flow (ALE only)

- Adapter acts as a Remote Function Call (RFC) Server by listening for ALE events from an SAP application
- Adapter spawns listener threads to the SAP application
 - ▶ Adapter uses these listener threads to poll for events at regular intervals
- Whenever an event occurs in SAP, the event is pushed to the adapter via the event listeners
- Adapter uses the Event Recovery Table to track the events
- Adapter forwards the event to the endpoint (message driven bean)
 - ▶ Adapter receives the event from SAP as data in IDoc table format and then converts the data into a BO before sending it to the message endpoint



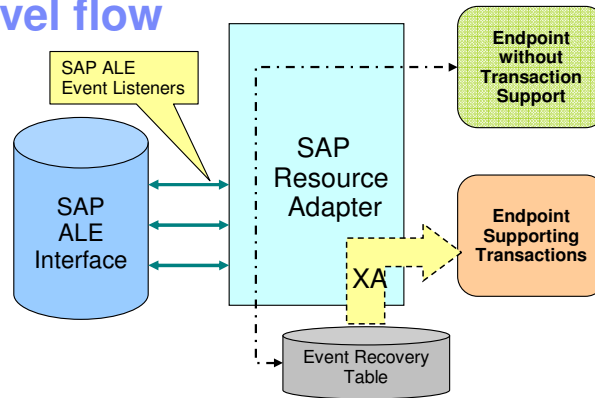
The adapter spawns a listener thread to the SAP system. The listener threads regularly poll for events. So, in a way, the adapter acts as a Remote function call server to the SAP application.

Whenever an event occurs in SAP, the event is pushed to the adapter via the event listeners. There is no event store within the SAP application. The adapter uses a persistent event recovery table to track the events.

Adapter receives the event from SAP as data in IDoc table format and then converts the data into a BO before sending it to the endpoint.

ALE inbound high level flow

- The adapter can track and recover events in case of abrupt termination using the data source for persisting the event state in an event recovery table



Event Recovery Table Fields

Table Field Name	Description
TID	Transaction ID for tRFC protocol
Status	Event processing status. Possible values are "CREATED", "EXECUTED", "PARTIAL" and "ROLLBACK"
NumIDocs	Total number of IDocs in the packet (if IDoc packet is send)
NumIDocsProcessed	Total number of IDocs from a packet that are successfully processed.
CurrIDoc	Sequence number of the IDoc in the packet that the adapter is currently processing

This diagram shows the components involved in end to end handling of the events. The SAP ALE event listeners listen for the events. The new event is pushed to the adapter and the adapter stores it in the persistent event recovery table. The persistent table provides failure recovery to reprocess the events.

The schema of the event recovery table is shown. The TID is the unique transaction id that SAP application uses for events. SAP will send the same TID for subsequent processing of the same event. An event may represent an IDoc packet. The table has fields for the total number of IDocs in the packet, the total number of IDocs that are successfully processed and the current IDoc that is being processed.

Event processing for single IDoc

From SAP	Within Adapter
SAP system pushes the Transaction ID (TID) to the adapter	Adapter checks the status of the event in the Event Recovery Table: If new event, an entry is created in the table with status=CREATED If event exists and status=ROLLBACK, update status to CREATED If event exists and status=EXECUTED, return SUCCESS to SAP
SAP system pushes the single IDoc to adapter	Parse IDOC and create BO
SAP sends COMMIT call to adapter	Adapter sends BO to the endpoint (under XA transaction or no transaction). If BO delivered and if XA transaction is committed, status changed to EXECUTED. If error occurs, status changed to ROLLBACK
SAP sends CONFIRM call to adapter	adapter deletes the records from the event recovery table and logs a common event infrastructure (CEI) event



This table shows the actions that occur on the SAP side and within the adapter when processing a Single IDoc.

The description is self explanatory.

Event processing for IDoc packet

From SAP	Within Adapter
SAP system pushes the Transaction ID (TID) to the adapter	Adapter checks the status of the event in the Event Recovery Table: If new event, an entry is created in the table with status=CREATED If event exists and status=ROLLBACK, update status to CREATED If event exists and status=EXECUTED, return SUCCESS to SAP
SAP system pushes the single IDoc to adapter	Parse IDOC packet and create multiple BOs Adapter updates NumIDocs in the Event Recovery table If error occurs during parsing and IgnoreIDocPacketErrors=true, the adapter logs an error and continues processing the rest of the IDocs in the packet, and marks status=PARTIAL If error occurs during parsing and IgnoreIDocPacketErrors=false, adapter stops processing any further IDocs in the packet and reports errors to the SAP system
SAP sends COMMIT call to adapter	Adapter sends BO to the endpoint, one after another and updates the CurrIDoc to the sequence # of IDoc being processed. Delivery of BO may be part of a XA transaction (if supported by endpoint) When all BOs are send, status is updated to EXECUTED. If error occurs, status changed to ROLLBACK
SAP sends CONFIRM call to adapter	Adapter deletes the records from the event recovery table and logs a CEI event

This table shows the actions that occur on the SAP side and within the adapter when processing a IDoc packet.

The description is self explanatory.

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 CurrIDoc property, even if IgnoreIDocPacketErrors is set to true. This is required in case the user terminates the adapter manually while processing an IDoc.

ALE inbound business objects supported verbs

- Adapter uses IDoc control record field to determine the verb
- The verb is set on the Business Object, before sending it to the Endpoint
- Following fields in the IDoc control record are used to determine the verb:
 - ▶ Logical_message_type (MESTYP)
 - ▶ Logical_message_code (MESCOD)
 - ▶ Logical_message_function (MESFCT)

Verb	Definition
Create	The top-level business object and all contained children has been created
UpdateWithDelete	The top level BO has been modified. Can include adding and deleting child objects. Any and all deleted child objects are known and reflected
Delete	The top level BO and any contained child objects have been deleted

The supported verbs for ALE inbound operations are create, updatewithdelete and Delete. The adapter uses the IDoc control record field to determine the verb. The verb is then set on the Business Object, before sending it to the Endpoint

ALE inbound events – additional details

- Adapter supports sending of ALE events without parsing the data and converting it into a BO, based on the Adapter's J2C ActivationSpec "*ParseIDocs*" property
 - ▶ BO would contain the IDoc as byte array. Endpoint would have to handle the parsing and converting it to a BO
 - ▶ If "*ParseIDocs*" is false, Adapter will not parse the IDoc
- Updating IDoc status
 - ▶ Adapter can be configured to update the IDoc status back to SAP for monitoring purposes
 - ▶ Done by setting **ALEUpdateStatus** property to true in the J2C ActivationSpec
 - ▶ Update is accomplished by updating a status IDoc called ALEAUD that the adapter sends to the SAP system, after the event is sent to the message endpoint
 - ▶ The Adapter ActivationSpec to indicate success and failure are ALESuccessCode, ALESuccessText, ALEFailureCode, and ALEFailureText



Some additional details are included here.

If the end point needs the incoming IDoc without having the adapter parse the IDoc and creating a BO representing the parsed IDoc, the developer can set the "**ParseIDocs**" attribute on the ActivationSpec to false. If ParseIDocs is set to false, the BO would contain the IDoc as byte array.

Another attribute is **ALEUpdateStatus**. If this is set to true in the ActivationSpec, the adapter will update the IDoc status back to SAP application for monitoring purposes. The update is accomplished by updating a status IDoc called ALEAUD that the adapter sends to the SAP system, after the event is sent to the message endpoint. The ActivationSpec contains additional attributes to indicate the success code, success text, failure code and failure text to be sent to the SAP application.

Transaction and security

This section covers the transaction and security support.

Transaction and security

- Transaction
 - ▶ SAP application support transaction for BAPI calls only
 - ▶ Adapter provides local transaction support for the BAPI interface using the following BAPI calls
 - BAPI_TRANSACTION_COMMIT
 - BAPI_TRANSACTION_ROLLBACK

- Security
 - ▶ Adapter supports container managed sign-on and basic authentication. It does not support re-authentication
 - ▶ For outbound, use the WebSphere Process Server J2C Authentication Alias to specify the user id and password to connect to the EIS
 - ▶ Inbound does not use J2C Authentication Alias – instead uses user id and password on the ActivationSpec



SAP application supports transaction for BAPI calls only through the use of BAPI_TRANSACTION_COMMIT and BAPI_TRANSACTION_ROLLBACK calls.

On the security front, for the outbound request, the J2C Authentication Alias within the WebSphere Process Server is used. The authentication alias name can be specified in WebSphere Integration Developer. The administrator must have that J2C Authentication Alias defined within the Process Server with its user id and the password to authenticate the SAP application. Pre-defined J2C Authentication Alias “**SCA_Auth_Alias**” can be used for authentication.

For the inbound request, the user id and password properties specified on the inbound ActivationSpec is used. J2C Authentication Alias is not used for inbound operations.

***Configuration:
SAP properties for RAR deployment
descriptor, managed connection and
activation spec***



Common properties were covered In the WebSphere Adapter common details presentation. This section covers the SAP specific attributes.

RAR custom properties

Property	Description
PartnerCharset	This property is used to specify a PartnerCharset encoding. When populated with an encoding, it will be used. Otherwise the default encoding will be used. To use the default encoding, leave this property value blank

Log and Trace Custom Properties are common for all WebSphere Adapters and are covered in the separate common presentation

The PartnerCharset custom property in the SAP adapter is used to specify the encoding to be used. If no value is specified, the default encoding will be used.

Managed connection factory properties (for outbound)

Configuration Property	Details
Client	SAP Client number under which the resource adapter logs in, often 100
Username	User account on the SAP application.
Password	Password
Language	The default is E , for English.
ApplicationServerHost	When configuring the resource adapter to run without load balancing, specifies the IP address or the name of the application server that the resource adapter logs in to. In both cases, the resource adapter assumes that the name of the gateway host is the same as the value specified for this property.
SystemNumber	System number of the application server. The value is a two-digit number, often 00 - Default is 00.
GatewayHost	Host where the gateway service is running.
GatewayService	Gateway server identifier; often sapgw00. The 00 is the system number of the server running the SAP Gateway (usually an application server) and may not be 00 if you have more than one. The default is sapgw00.



This table shows the managed connection factory configuration properties for outbound request.

The details in the table are self explanatory.

Managed connection factory properties (for outbound) – cont.

Configuration Property	Details
Group	When configuring the resource adapter for load balancing, specifies the name of the logon group that represents a group of application servers.
MessageServerHost	When configuring the resource adapter for load balancing, specifies the name of the message server.
SAPSystemID	When configuring the resource adapter for load balancing, specifies the logical name of the SAP system, which is also known as R3name.
ABAPDebug	This property specifies whether the resource adapter invokes the ABAP Debugger for the appropriate function module when the resource adapter begins processing a business object. When this property is set to true, the resource adapter opens the ABAP Debugger. A dialog user with proper user authorizations is required for debugging. You can add breakpoints only after the debugger opens. Important: This property should always be set to false in a production environment. The default value is false.
RFCTraceOn	This property specifies whether or not to generate a text file detailing the RFC activity for each listener thread. You can specify a value of true or false. A value of true activates tracing, which generates a text file. It is recommended that you use these text files in a development environment only, because the files can grow rapidly. The default is false.

37

This table shows some of the ActivationSpec configuration properties for inbound calls. The details in the table are self explanatory.

Activation specification properties (for inbound) – connection properties

Property	Description
GatewayHost	SAP Gateway Host where the gateway service is running.
GatewayService	Gateway server identifier; often sapgw00 . The 00 is the system number of the server running the SAP Gateway (usually an application server) and may not be 00 if you have more than one. The default is sapgw00.
RfcProgramID	Program identifier that the RFC Server program will register under
Client	SAP Client number under which the resource adapter logs in, often 100
Username	Name of the resource adapter's user account on the SAP application.
Password	Password for the resource adapter's user account on the SAP application.
Language	Language in which the resource adapter logs in. The default is E , for English.
ApplicationServer Host	When configuring the resource adapter to run without load balancing, specifies the IP address or the name of the SAP application server

This table shows more ActivationSpec configuration properties for inbound calls. The details in the table are self explanatory.

Activation specification properties - connection properties (cont.)

Property	Description
SystemNumber	System number of the application server. The value is a two-digit number, often 00. The default is 00.
Group	When configuring the resource adapter for load balancing, specifies the name of the logon group that represents a group of application servers.
MessageServerHost	When configuring the resource adapter for load balancing, specifies the name of the message server.
SAPSystemID	When configuring the resource adapter for load balancing, specifies the logical name of the SAP system, which is also known as R3name.
RFCTraceOn	This property specifies whether or not to generate a text file detailing the RFC activity for each listener thread. You can specify a value of true or false. A value of true activates tracing, which generates a text file. It is recommended that you use these text files in a development environment only, because the files can grow rapidly. The default is false.

This table shows more ActivationSpec configuration properties for inbound calls. The details in the table are self explanatory.

Activation specification properties for error recovery table

Property	Description
EDTDriverName EDTDatabaseName EDTTableName EDTUserName EDTUserPassword EDTSchemaName EDTURL EDTServerName EDTPortName	Event Recovery Table information for inbound events. SAP does not have its own event recovery mechanism. Hence, persistent event recovery table is required for SAP adapter.
AutoCreateEDT	Flag that indicates whether the adapter should create the Error Recovery Table automatically if it doesn't already exists – Default: TRUE

The table shows ActivationSpec configuration properties for inbound calls related to the Event recovery table.

The details in the table are self explanatory.

Activation specification properties for IDoc processing

Property	Description
ParselDocs	Used to decide whether to convert the IDocs into business objects or not
SplitIDocPackets	Used to decide whether the adapter needs to send the entire IDoc packet or split it into multiple IDocs. This property is only relevant for sending IDocs without parsing.
IgnoreIDocPacketErrors	If adapter encounters an error while processing the IDoc packet, it can behave in two different ways depending upon this configuration property. When this property is set to false adapter will stop processing further IDocs in that packet and report error to the SAP application. When this property is set to true adapter logs an error and continues processing the rest of the IDocs in that packet.

This table shows ActivationSpec configuration properties for inbound calls related to how the IDoc is to be processed by the adapter.

The details in the table are self explanatory.

Activation specification properties for ALE properties

Property	Description
AleUpdateStatus	Specifies whether an audit trail is required for all message types. This property must be set to true to cause the connector to update a standard SAP status code after the ALE Module has retrieved an IDoc object for event processing.
AleSelectiveUpdate	Specifies which IDoc Type and MessageType combinations are to be updated when the connector is configured to update a standard SAP status code. Valid only if AleUpdateStatus=True The syntax for this property is: <i>IDocType:MessageType[,IDocType:MessageType [...]]</i> where a colon (:) delimiter separates each IDoc Type and MessageType, and a comma (,) delimiter separates entries in a set. Example below illustrates two sets. In the example, MATMAS03 and DEBMAS03 are the IDocs, and MATMAS and DEBMAS are the message types: MATMAS03:MATMAS,DEBMAS03:DEBMAS
AleStatusMsgCode	If required, specifies the message code to use when the connector posts the ALEAUD Message IDoc (ALEAUD01). Configure this message code in the receiving Partner Profile. Valid only if AleUpdateStatus=True

This table shows ActivationSpec configuration properties for inbound calls related to ALE update status that is sent back to the SAP application for monitoring purposes.

The details in the table are self explanatory.

Activation specification properties for ALE properties (cont.)

Property	Description
AleSuccessCode	Specifies the success status code for Application Document Posted. You must specify a value for this property (52 or 53) to cause the connector to update the SAP success status code after the ALE Module has retrieved an IDoc object for event processing. SAP converts this value to status 41 (Application Document Created in Receiving System).
AleFailureCode	Specifies the status code for dispatch failure. You must specify a value for this property (68 or 58) to cause the connector 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.
AleSuccessText	Specifies the descriptive text for successful Application Document Posted. Specifying a value for this property is optional, even when you set AleUpdateStatus to true.
AleFailureText	Specifies the descriptive text for dispatch failure. Specifying a value for this property is optional, even when you set AleUpdateStatus to true.

The table shows ActivationSpec configuration properties for inbound calls related to ALE update status that is sent back to the SAP application for monitoring purposes.

The details in the table are self explanatory.

Problem determination



This section is a review of Problem Determination. Problem Determination methodology, log and trace files are similar in all the WebSphere Adapters, and are covered in the common details presentation.

Problem determination

- Covered in the Common Adapter details presentation –
Recap:
 - ▶ WebSphere Process Server log files (SystemOut.log and SystemErr.log)
 - ▶ Adapter Log and Trace files configured in WebSphere Integration Developer EMD and using the RAR custom properties in the Administrative console of the Process Server
 - ▶ Different logging level and tracing levels can be set
- Enabling trace for SAP Adapter in WPS :
 - ▶ Set the tracing level string as "com.ibm.j2ca.sap.*=finest"
- Before interacting with the Adapter, test the components in SAP, where possible



Listed here are the log files for WebSphere Process Server. Note the trace string used to turn on tracing of the SAP adapter, along with the levels.

Best practice is, where possible, to try to test the SAP components like the custom event component, generation of event triggers within the SAP backend before trying the end to end scenario with the adapter.

Summary and references

This section provides the summary and references.

Summary and references

- Summary

- ▶ Discussed SAP Adapter in details covering the architecture, operations, EMD, Problem Determination

- References

- ▶ Information Center
- ▶ User Guide
- ▶ IBM Education Assistant



In summary, this presentation covered the details of WebSphere Adapter for SAP.

More information can be found in the user guide and the Information Center for the Adapter.

Trademarks, Copyrights, and Disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM	CICS	IMS	MQSeries	Tivoli
IBM (logo)	Cloudscape	Informix	OS/390	WebSphere
eIogo business	DB2	Series	OS/400	xSeries
AIX	DB2 Universal Database	Lotus	pSeries	zSeries

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds.

Other company, product and service names may be trademarks or service marks of others.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements and/or changes in the product(s) and/or program(s) described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (e.g., IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. 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 in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

© Copyright International Business Machines Corporation 2006. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.