



IBM Software Group

## SAP integration workshop

### *IBM WebSphere Business Integration Adapter for mySAP.com*



@business on demand.

© 2007 IBM Corporation  
Converted to video June 18, 2014

This presentation gives an introduction to the IBM WebSphere® Business Integration Adapter for mySAP.com. After the presentation you will have some familiarity with the environments that the adapter is supported in, the adapter components and their features, and the development tools used with the adapter. The presentation will also list some issues that might be important when using or considering the adapter.

## Agenda

- Introduction to the WebSphere Business Integration Adapter for mySAP.com
  - ▶ Adapter overview
  - ▶ Overview of event detection
  - ▶ Overview of the ABAP extension module
  - ▶ Overview of the BAPI module
  - ▶ Overview of the hierarchical dynamic retrieve module
  - ▶ Overview of the RFC server module
  - ▶ Overview of the ALE module
  - ▶ Overview of the object discovery agent
  - ▶ Summary

The agenda for this presentation begins with the adapter overview, then an overview of the event detection. Next is an overview of the individual modules: the ABAP extension, BAPI, hierarchical dynamic retrieve, RFC Server and the ALE modules. Then, after an overview of the object discovery agent, this presentation will conclude with a summary of the questions, concerns and decisions that accompany the use of this adapter.

## Adapter overview

- Supports SAP R/3 versions 3 and 4, and Web AS versions up to 7.00
- Supported platforms:
  - ▶ Windows® (2000, XP, 2003),
  - ▶ UNIX® (AIX, Solaris, and HP-UX)
  - ▶ Linux® (Red Hat Enterprise and SUSE)
- Written in Java™
- Multiple connector modules offer various ways to integrate with R/3



The mySAP.com adapter includes a connector, message files, configuration tools, and an object discovery agent, or ODA. The connector allows the WebSphere integration broker to exchange business objects with SAP applications.

The adapter supports SAP R/3 versions 3.0 through 4.7, and Web Application Server versions up to 7.

It is written in Java and supports Windows, UNIX, and Linux platforms as listed here.

There are multiple Connector modules which offer various ways to integrate with R/3. These are the ABAP Extension Module, ALE Module, BAPI Module, Hierarchical Dynamic Retrieve Module, and RFC Server Module.

Hierarchical Dynamic Retrieve Module – WebSphere Business Integration delivers a wizard which can be used to generate business objects which run with the Hierarchical Dynamic Retrieve Module. To use this wizard, it needs to be loaded into an SAP system, making it invasive solution. The runtime environment of the Hierarchical Dynamic Retrieve Module, however, is completely noninvasive.

The only difference between the version 3 and version 4 adapters is that the version 4 adapters offer wizards that are not offered with the version 3 adapter. SAP offers more flexibility in version 4 that enable WebSphere Business Integration to deliver more robust tools.

## Adapter features

- Uses SAP's Java API (JCO – Java connector)
- Invasive and non-invasive solutions
- Invasive solution uses /CWLD/ namespace
- Event notification (polling)
- Multi-threaded business object processing



The adapter uses SAP's Java API. SAP's Java connector (also called JCO) is written in C with a Java wrapper

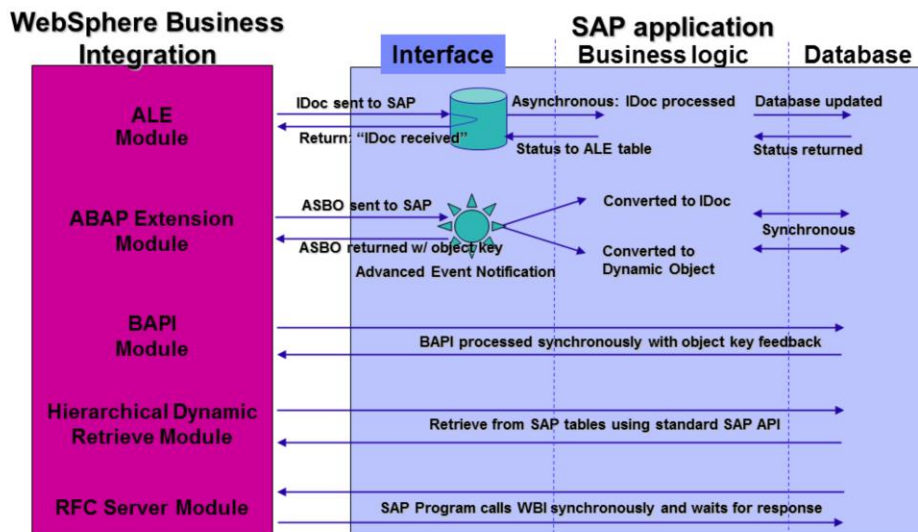
While the connector is written in Java, some modules have components written in ABAP that run in the SAP environment. In other words, there are invasive and non-invasive solutions.

The invasive solution uses the /CWLD/ namespace in SAP based on the acquired company CrossWorlds.

Event notification, provided through polling, is included with the ABAP Extension module.

The newer versions of the adapter (version 4.4 and later) are multi-threaded. They feature multiple concurrent interactions from a single adapter JVM. This feature is supported by the Vision framework, which can create separate threads for processing an application event or for concurrent business object requests. That means improved performance. It can be also advantageous to run multiple adapters to use event distribution for data coming out of an SAP application and it is necessary if you want to deliver data to various Interchange Servers. Multi-threading works best for moving data into an SAP application.

## Integration Architecture



5

IBM WebSphere Business Integration Adapter for mySAP.com

© 2007 IBM Corporation

This graphic shows diagram of the adapter modules, and how they interact with SAP. ALE stands for "application link enabling." The ALE module part of this diagram depicts request processing. IDOC data is sent to SAP and success message is returned to the integration broker. The IDOC is processed asynchronously in the SAP

The ABAP Extension Module part of the diagram shows the business object handler request-response sequence and represents the advanced event notification, which is accomplished by polling.

The BAPI module and the Hierarchical Dynamic Retrieve module parts of the diagram show the synchronous request / response sequence from the adapter to SAP

RFC server module shows the synchronous request and response from SAP to the adapter.

## Section

# ***Overview of event detection***

This part of the presentation provides an overview of event detection mechanisms.

## Event detection mechanisms

- **Workflow (business workflow)**
  - ▶ PRO - Real-time integration
  - ▶ PRO - Easy to implement
  - ▶ CON - Limited availability – business object does not always exist for each business process
- **Customer exit (user exit)**
  - ▶ PRO - Real-time integration
  - ▶ CON - Moderately difficult to implement
  - ▶ CON - Limited availability – SAP provided “safe” locations to insert event detection code
- **Code modification**
  - ▶ PRO - Extensive availability – can be inserted precisely at the appropriate location in the code
  - ▶ PRO - Real-time integration
  - ▶ CON - Difficult to implement
  - ▶ CON - Difficult when upgrading R/3 versions
- **Batch programs**
  - ▶ PRO - Extensive availability – can be implemented for most processes
  - ▶ PRO - Good for processing a large number of events
  - ▶ CON - Moderately difficult to implement
  - ▶ CON - Not used for real-time integration

7

IBM WebSphere Business Integration Adapter for mySAP.com

© 2007 IBM Corporation

Typically most adapters use database triggers; however, SAP allows limited access to the underlying database, so event detection mechanisms are implemented in the application transaction layer above the database. WebSphere Business Integration uses four mechanisms. All of them have their advantages and disadvantages.

Business Workflow is easy to implement and provides real-time integration. Disadvantage is limited availability if this mechanism – business object does not always exist for each business process.

User Exit is a design feature provided by SAP that can be used for real-time integration and is moderately difficult to implement. Availability is limited in a sense that SAP provided “safe” locations to insert event detection code.

Code Modification has extensive availability. Code can be inserted precisely at the appropriate location in the existing code and provides real-time integration. It can be difficult to implement and can be difficult to migrate when upgrading R/3 versions.

Batch programs are good for processing a large number of events, have extensive availability in the sense that they can be implemented for most processes. They are moderately difficult to implement and can not be used for real-time integration.

There are more mechanisms in the SAP that could be used or exploited for event detection but these are the four primarily used with WebSphere Business Integration.

## Section

# ***Overview of the ABAP extension module***

This part of the presentation provides an overview of the ABAP extension module.



## Overview – Features

- Uses components written in Java and counterparts in ABAP
- Only module with polling capability
- SAP granted a development license and namespace (/CWLD/)
- Invasive solution enables WebSphere Business Integration to deliver robust functionality
  - ▶ Business object development
  - ▶ Logging
  - ▶ Reprocessing
  - ▶ Event notification (polling)
  - ▶ Event handling
- Delivered with tools (WebSphere Business Integration station)



The ABAP extension module consists of components written in Java and their counterparts, written in ABAP, which run on the SAP system.

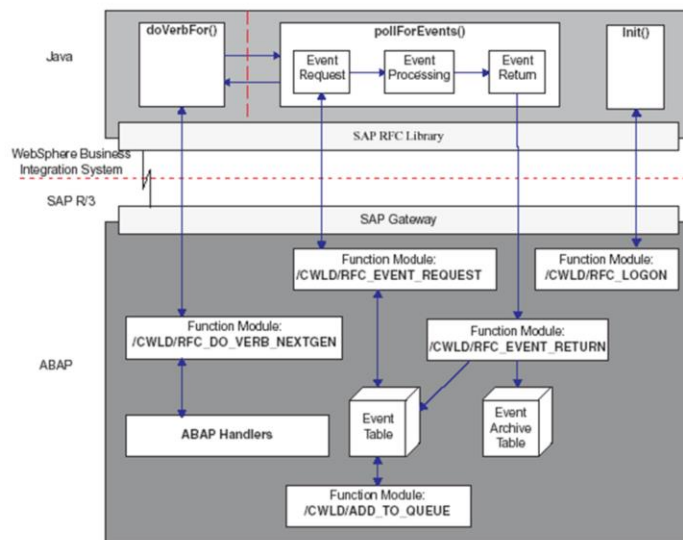
It is the only module with polling capability. This feature can be combined with other modules, for instance: use the ABAP extension module for polling and any other modules for subsequent retrieval.

SAP granted WebSphere Business Integration a development license and namespace (/CWLD/). WebSphere Business Integration provides a namespace key through Technical Support if it is necessary to modify existing components or develop new components in the WebSphere Business Integration namespace. This may happen only in rare cases.

It is an invasive solution because SAP runtime components must be installed for this module to function. However, it enables WebSphere Business Integration to deliver robust functionality such as Business object development, Logging, Reprocessing, Event notification using polling, and Event handling.

The module is delivered with tools running on SAP. These tools are grouped as WebSphere Business Integration station, and can be invoked from the transaction /CWLD/HOME

## Overview – Components



10

IBM WebSphere Business Integration Adapter for mySAP.com

© 2007 IBM Corporation

The connector for SAP is written in Java and consists of two parts: the vision connector framework and connector modules. The vision connector framework calls any implementation of the `init()`, `pollForEvents()`, and `terminate()` methods. It routes business objects to specific business object handlers based on the verb using the `doVerbFor()` method.

The connector's ABAP function module `/CWLD/RFC_DO_VERB_NEXTGEN` and an ABAP handler in the ABAP component of the connector module handle the requests for business object processing. The supplied ABAP handlers are `/CWLD/DYNAMIC_RETRIEVE`, `/CWLD/DYNAMIC_TRANSACTION`, and `CWLD/IDOC_HANDLER`. Additional business-object-specific handlers may need to be developed.

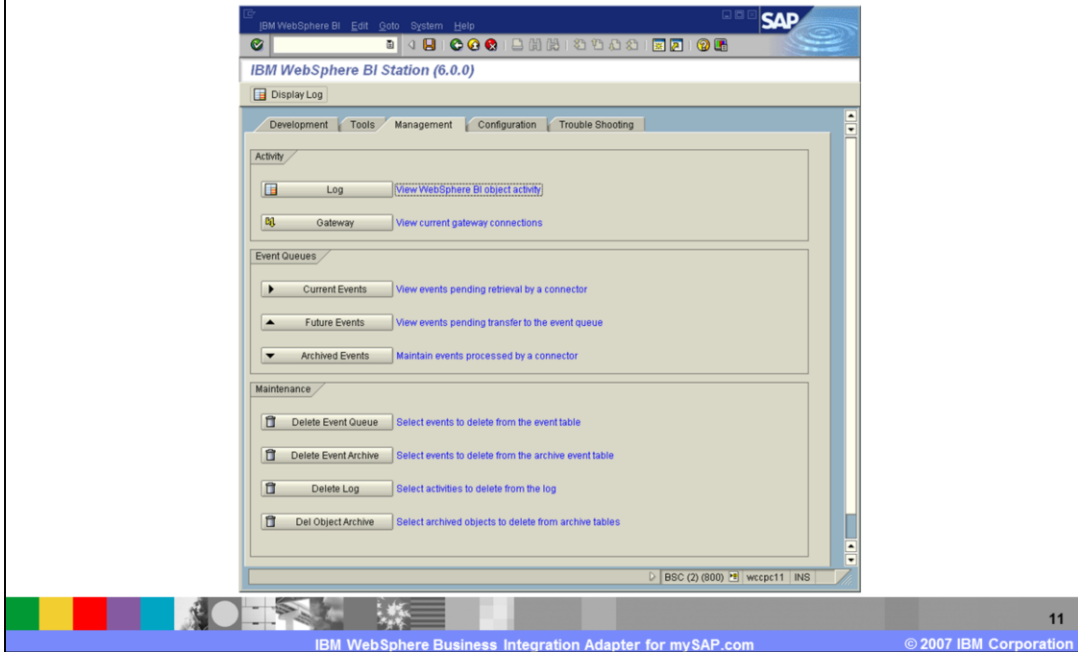
Event notification for the connector consist of two functions. Event polling and event triggering.

Event polling consists of three functions that are carried out by the `pollForEvents()` method: Event request, event processing, and event return. Event request Java component has a counterpart function module in the SAP application - `/CWLD/RFC_EVENT_REQUEST` which selects the events to return from the event table. The result is passed to the event processing function, which handles them one at a time. After each event is processed, it is returned to the SAP application using function module `/CWLD/RFC_EVENT_RETURN`. This module adds a copy of the processed event to the event archive table (`/CWLD/EVT_ARC`), and deletes the original entry from the event table.

After the event is detected in the SAP system, it is triggered using one of the adapter-delivered event triggers. Function module `/CWLD/ADD_TO_QUEUE` adds event to the current event table for immediate processing. The event detection was discussed in the

earlier section of this presentation.

# IBM WebSphere BI Station



The ABAP extension module is delivered with a tool – IBM WebSphere Business Integration station. It runs on SAP, and is invoked by transaction /CWLD/HOME. The functionality is grouped in several tabs – development, tools, management, configuration and troubleshooting. The slide shows the management tab which is useful for examination and maintenance of activity, logs, event queues, and archives.

## Section

# ***Overview of the BAPI module***

This part of the presentation provides an overview of the BAPI module.

## Overview – Features of the BAPI module

- Written in Java
- Non-invasive solution
- Supports any RFC-enabled function module
- SAP object discovery agent

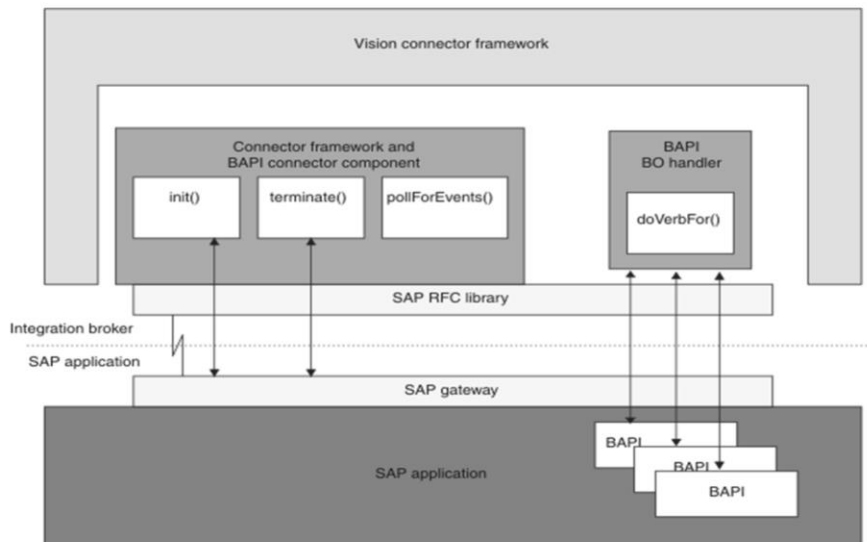


The BAPI Module is a connector module, written in Java, that supports native BAPI calls directly to an SAP application. It uses the SAP RFC libraries written in Java and C, which enable external programs to communicate with an SAP application.

It is a non-invasive solution that supports any RFC-enabled function module.

SAP ODA is used to generate business object definitions for each RFC-enabled function. It uses SAP's standard APIs to generate a business object representation of a BAPI. Also creates a BAPI-specific business object handler in Java to support the generated business object and BAPI.

## Overview – Components



14

IBM WebSphere Business Integration Adapter for mySAP.com

© 2007 IBM Corporation

The BAPI Module implements the `init()`, `terminate()`, `pollForEvents()`, and `doVerbFor()` methods. However, the `pollForEvents()` method is not used because the BAPI Module supports request operations only. The `init()` method opens an RFC connection with the SAP application through the SAP Gateway. If the connector fails to initialize, it terminates using the `terminate()`.

A single implementation of the `doVerbFor()` method in the vision connector framework's business object handler initiates all business object requests. The vision business object handler processes all of the business objects passed between the BAPI Module and the integration broker. In the BAPI Module, a single BAPI business object handler supports all BAPI calls.

## Section

# ***Overview of the hierarchical dynamic retrieve module***

This part of the presentation provides an overview of the hierarchical dynamic retrieve module.



## Overview – Features

- Written in Java
- Runtime completely non-invasive
- Advanced outbound wizard generates business object definitions
- ODA can be used to develop objects for individual tables
- Ideal when retrieving data from custom tables

16

IBM WebSphere Business Integration Adapter for mySAP.com

© 2007 IBM Corporation

The hierarchical dynamic retrieve module is a connector written in Java that processes hierarchical business objects recursively. It performs the same steps for each child business object until it has processed all individual business objects in the hierarchy.

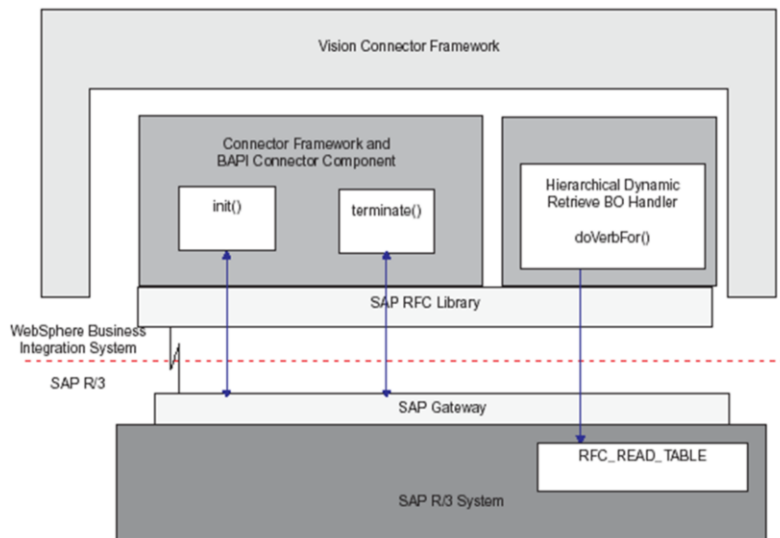
The runtime is completely non-invasive.

There is an Advanced Outbound Wizard deployable in R/3 which generates business object definitions, but it only needs to be installed on the development system and not on the client's production system.

On WebSphere development side, ODA can be used to develop objects for individual tables. Business object designer can also be used to link them into hierarchies.

The hierarchical dynamic retrieve module is especially suited to retrieving data from custom tables.

## Overview – Components



The hierarchical dynamic retrieve module extends the vision connector framework. The module does not have its own application-specific component, it uses the application-specific component for BAPI. Therefore, the module consists of the connector framework, the application-specific component for BAPI, the Dynamic Retrieve business object handler, and the SAP RFC libraries.

## Application specific information

The screenshot shows the Business Object Designer interface for the object SAP\_CustomerAddress:ICL101. The 'Attributes' tab is active, displaying a table of attributes with their properties and application-specific information.

Pos	Name	Type	Key	Foreign	Required	Card	Maximum	App Spec Info
1	CustomerId	String	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255	TN=KNA1:CN=KUNNR
2	Name	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255	TN=KNA1:CN=NAME1
3	Street	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255	TN=KNA1:CN=STRAS
4	AddressId	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255	TN=KNA1:CN=ADRNR
5	<input type="checkbox"/> SAP_CustomerEmail	SAP_CustomerEmail	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1		
5.1	5.1 AddressId	String	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255	TN=ADR6:CN=ADDRNUMBER:FK=..AddressId
5.2	5.2 EmailAddress	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255	TN=ADR6:CN=SMTP_ADDR
5.3	5.3 ObjectEventId	String						
6	6 ObjectEventId	String						
7	7		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		255	

This slide shows how the business object designer can be used to link the O.D.A. developed objects for individual tables into hierarchies.

The meaning of the information in the application specific information column is listed in the notes for this slide.

Note that the period (.) in the application specific information column points to the parent business object as seen in the example at line 5.1. (FK=..ForeignKeyAttributeOfParent).

TN=TABLE:

CN=FIELD:

FK=ForeignKeyAttribute

FK=..ForeignKeyAttributeOfParent

OP=Operand (EQ,LE,GE,LT,GT,LIKE, Default EQ)

## Section

# ***Overview of the RFC server module***

This part of the presentation provides an overview of the RFC server module.

## Overview – Features

- Written in Java
- Non-invasive solution
- Acts as a server to an SAP application (SAP becomes the client)
- Supports all RFC-enabled functions
- Uses multiple listener threads to pick up events from the SAP Gateway
- SAP object discovery agent generates WebSphere Business Integration business object definition files and business object handlers

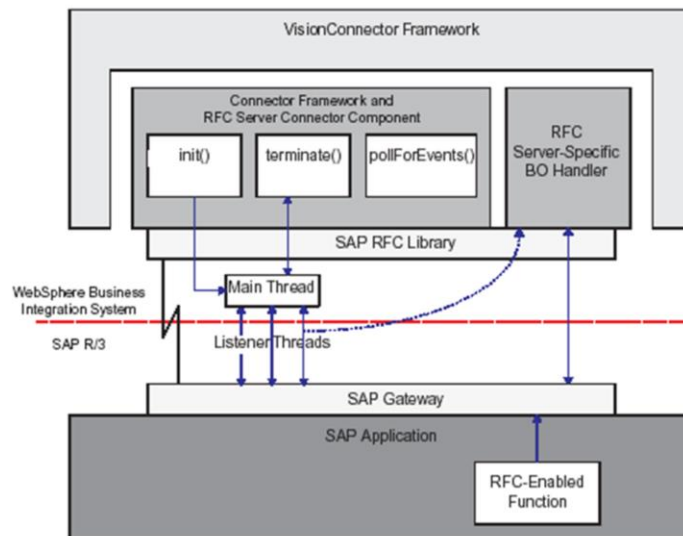
The RFC server module is a connector module written in Java that supports RFC calls directly from an SAP application.

It is a non-invasive solution – the connector has no components that need to run on the R/3 system. It acts as a server to an SAP application and SAP becomes the client, and supports all RFC-enabled functions.

The RFC server module uses multiple listener threads to pick up events from the SAP Gateway.

The SAP object discovery agent generates WebSphere Business Integration business object definition files and handlers.

## Overview – Components



21

IBM WebSphere Business Integration Adapter for mySAP.com

© 2007 IBM Corporation

The connector extends the vision connector framework by implementing the VisionConnectorAgent class. It spawns listener threads that open handles to the SAP application using the SAP RFC library and the SAP Gateway. Each listener thread opens a single handle to the SAP application.

Then the connector processes requests from RFC-enabled functions in the SAP application and finally terminates connections to the SAP application.

The RFC Server-specific business object handler retrieves the RFC event data and populates the associated WebSphere business object for SAP. Then it passes the business object to the integration broker and receives a business object in return.

## Section

# ***Overview of the ALE module***

This section of the presentation provides an overview of the ALE module.

## Overview – Features of the ALE module

- Non-invasive Java solution
- Asynchronous communication
- Uses SAP's ALE (application link enabling) technology
- Supports IDocs and SAP's tRFC protocol
- SAP ODA tool generates WebSphere Business Integration business object definitions based on IDocs
- Ideal way to leverage existing ALE interfaces

23

IBM WebSphere Business Integration Adapter for mySAP.com

© 2007 IBM Corporation

The ALE module, written in Java, is a non-invasive solution.

ALE module uses asynchronous communication.

It is based on SAP's application link enabling, or ALE, technology.

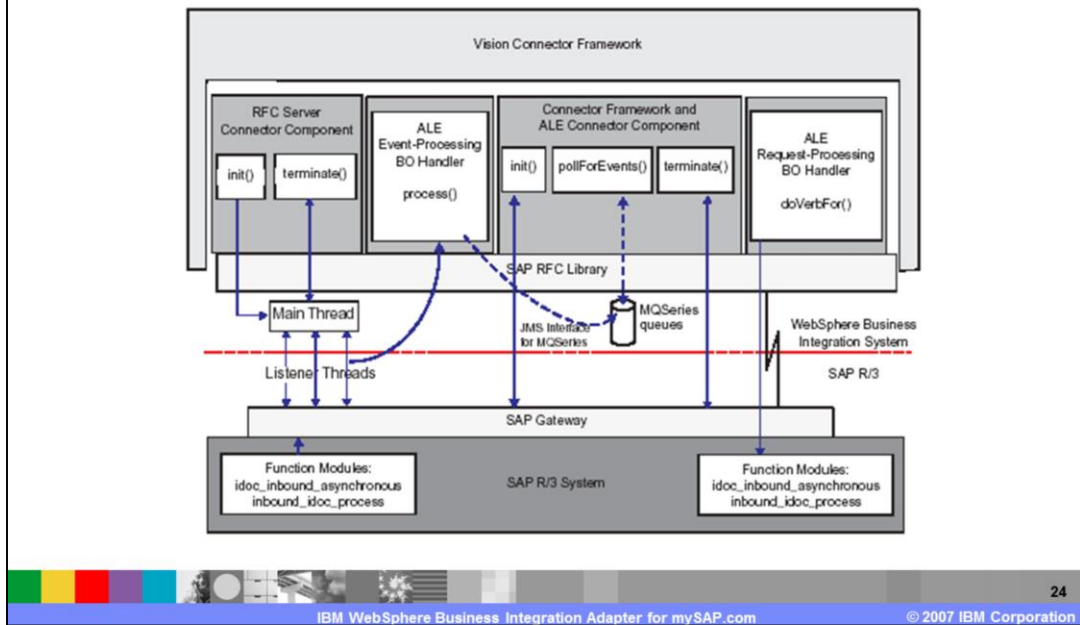
It supports IDocs and SAP's tRFC protocol

SAP ODA tool generates WebSphere Business Integration business object definitions based on IDocs.

It is an ideal way to leverage existing ALE interfaces.



## Overview – Components



The ALE module extends the vision connector framework. The module components include the RFC server adapter agent, which is used for event processing only. The ALE module uses the RFC server connector component because the similarities for event processing both support RFC calls directly from the SAP application.

Listener Threads are spawned by the RFC Server adapter agent. They register on the SAP Gateway and process events sent from the SAP Gateway.

The ALE event-processing business object handler is used during event processing; it retrieves RFC event data from R/3, passes it to the integration broker, and then returns a response to the ALE-specific function through the SAP Gateway.

The ALE adapter agent is used during request processing; it opens an RFC connection with an R/3 application.

Finally, the ALE request-processing business object handler is used during request processing; it converts WebSphere Business Integration business object data to and from an IDoc data format.

## Business Object Processing

- Event processing
  - ▶ All events initiated from the R/3 application
  - ▶ ALE data handler used to convert data
  - ▶ Transaction ID is stored persistently
- Request processing
  - ▶ All events sent to the R/3 applications
  - ▶ WebSphere Business Integration business object converted to IDoc data format
  - ▶ Transaction ID is persisted

25

IBM WebSphere Business Integration Adapter for mySAP.com

© 2007 IBM Corporation

There are two kinds of business object processing; event processing and request processing.

In event processing, the RFC Server connector component is started and spawns listener threads. Each listener thread registers a single handle to the SAP application and starts processing events from the SAP application.

All event processing in the ALE module is initiated from the SAP application by two RFC-enabled functions: IDOC\_INBOUND\_ASYNCHRONOUS and INBOUND\_IDOC\_PROCESS.

Each event from SAP is considered a transaction. The connector uses a two-step process with a transaction ID - also called TID - to handle each event, guaranteeing once-only delivery of data from SAP to the connector. MQ Series queues persistently store a JMS-MQ message for each event. Each JMS-MQ message stores the TID identifying the event, the status of the TID, the IDoc data associated with the event, and the processing status of the IDoc.

The ALE data handler creates WebSphere business objects from the stored event message, and sends the business objects to the integration broker.

For request processing, the ALE module opens an RFC connection to the SAP R/3 application. Then the ALE request-processing business object handler processes requests from the integration broker, converting them from business object format to IDoc data based on the SAP IDoc format.

For every request sent to the application, the ALE module persistently stores Transaction IDs in a TID queue as a JMS-MQ message. The TID guarantees that the request is delivered once and only once. However, if the integration broker sends an object that has the same value in the transaction ID attribute, this object will be processed again. Once an object has been successfully sent, the expectation is that the integration broker will not send the object again.

Finally, the ALE module releases the connection to the SAP R/3 application.

## Section

# *Overview of the tools*

This section of the presentation provides an overview of the tools.

## SAP object discovery agent

- IDocs:
  - ▶ Generates application-specific business object definitions from IDoc definition files or by connecting to R/3
- BAPI's and RFC's:
  - ▶ Maps business object attributes to BAPI parameters
  - ▶ Generates business object definition file
  - ▶ Generates BAPI/RFC-specific business object handler .class and .java files
  - ▶ Generated .java file can be changed and recompiled
  - ▶ Allows for synchronous or asynchronous calls
- Hierarchical dynamic retrieve
  - ▶ Generates business object definition file from SAP table definition

27

IBM WebSphere Business Integration Adapter for mySAP.com

© 2007 IBM Corporation

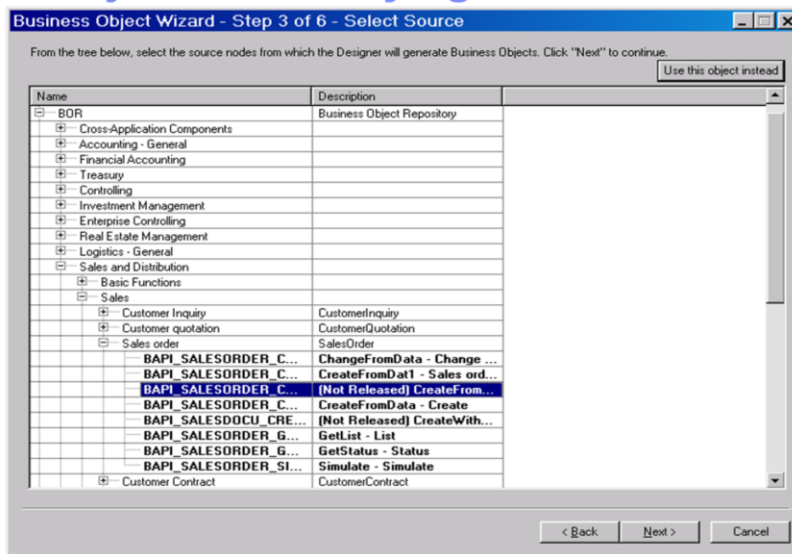
SAP ODA - object discovery agent - generates business object definitions for the WebSphere Business Integration adapter for mySAP.com. The connector works with objects that are based on IDoc types, BAPIs, RFC-enabled function modules defined in an SAP system, and SAP tables representing a business process. Because of this, SAP ODA uses these objects to discover business object requirements specific to its SAP data source.

For IDocs it generates application-specific business object definitions from IDoc definition files or by connecting to R/3.

For BAPIs and RFC it maps business object attributes to BAPI parameters, then generates business object definition file. Then it generates BAPI/RFC-specific business object handler .class and .java files. The generated .java file can be changed and recompiled. The ODS allows for synchronous or asynchronous calls.

For hierarchical dynamic retrieve, the ODA generates business object definition files from SAP table definitions.

## WebSphere Business Integration Adapter for SAP – Object discovery agent



28

IBM WebSphere Business Integration Adapter for mySAP.com

© 2007 IBM Corporation

SAP Object Discovery Agent is used in the business object designer to generate business object definitions. The business object designer has a wizard that configures the SAP ODA properties and launches the tool.

The business object designer displays a tree with five initial nodes: IDoc types, BOR, RFC, DTR, and HDR.

Under IDoc types you can browse for extracted IDoc definition files and select IDocs in the SAP system - both Basic IDoc Types and Extension Types) .

BOR is the Business object repository, where you can select objects that represent BAPIs from the SAP application.

Under RFC you can select objects that represent RFC-enabled functions from the SAP application.

DTR is Dynamic Transaction and Retrieve, where you can select the definitions that represent objects from the dynamic transaction and dynamic retrieve metadata tables.

And finally, HDR or Hierarchical Dynamic Retrieve. Here you can select the tables required to represent an entity for SAP transactions processed by the Hierarchical Dynamic Retrieve module.

The nodes whose names are preceded by a plus sign (+) are expandable. Click on them to display more nodes or leaves. SAP ODA generates business object definitions only from leaves.

## WebSphere Integration Developer

- Using Enterprise Service Discovery wizard
  - ▶ Select the WebSphere Business Integration adapter artifact importer
  - ▶ Specify the connector configuration file and business object schema directory [artifacts created using System Manager]
  - ▶ Discover and configure the available enterprise services.
  - ▶ Generate artifacts.

At this point the exports are available in the assembly editor

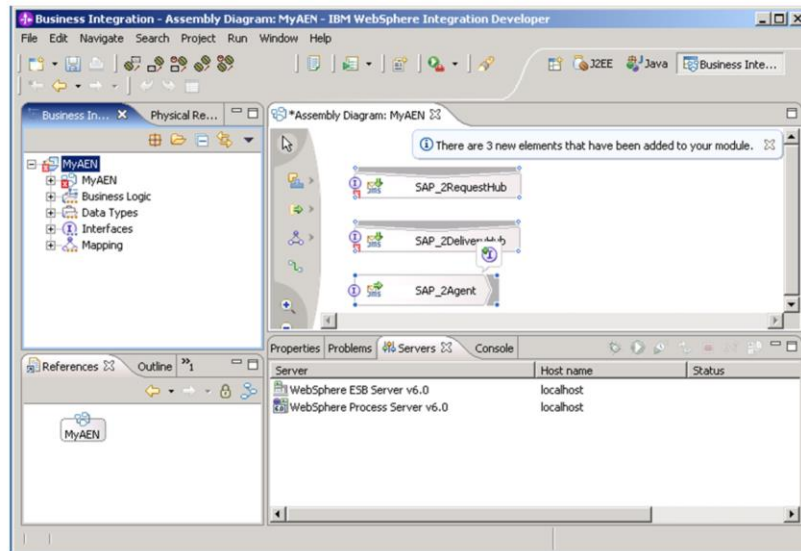


The WebSphere Business Integration Adapter for mySAP.com can be used with the WebSphere Process Server and the WebSphere ESB brokers. The development tool in these environments is the WebSphere Integration Developer.

Use the business object designer to configure ODA and create the business objects. Use the WebSphere Integration Developer enterprise service discovery wizard to select the WebSphere Business Integration adapter artifact importer. Then specify the connector configuration file and business object schema directory. Discover and configure the available enterprise services, and finally generate the artifacts.

At this point the exports are available in the assembly editor

# WebSphere Integration Developer



30

This slide shows the assembly diagram for the module MyAEN after the artifacts were generated.

## Section

# *Summary*

This section of the presentation provides a summary.



## Deciding which module to use

- Key questions to ask:
  - ▶ Is inbound or outbound support required ?
  - ▶ Is event triggering possible ?
  - ▶ Are there existing customer interfaces ?
  - ▶ Do you need to populate or read custom tables ?
  
- Possible concerns
  - ▶ Leverage existing interfaces (ALE)
  - ▶ No custom ABAP (BAPI, HDR)
  - ▶ No event triggering (RFC, ALE)

This presentation has described the various modules of the adapter, each providing different capability. In choosing the module, the good news is that all of them are present in the adapter. However it is necessary to consider if there are components that need to be deployed in SAP. So there are some key questions to ask listed here.

Some common concerns or requirements that need to be addressed include leveraging existing interfaces, the lack of custom ABAP, and the lack of event triggering with RFC and ALE.

## Decisions

- Outbound
  - ▶ BAPI available? BAPI
  - ▶ Custom logic required on SAP side? ABAP
  - ▶ ALE interfaces exist? ALE
  - ▶ SAP initiated call required? RFC
- Inbound
  - ▶ BAPI available? BAPI
  - ▶ Simple retrieve from related tables? HDR
  - ▶ Flat object? ABAP-Dynamic Transaction
  - ▶ Custom logic required? ABAP
  - ▶ Standard IDoc available? ALE

The direction considered in this slide is from SAP. From the WebSphere side – reverse inbound and outbound.

For outbound processing

If BAPI is available, use BAPI.

If custom logic is required on SAP side, then use ABAP code.

If ALE interfaces exist, use ALE.

If SAP initiated calls are required, you need the RFC server module.

If event processing is required, you need the ABAP extension module

For inbound processing

Use BAPI if it is available.

If you need simple retrieve from related tables, use HDR.

If you need flat objects, use ABAP dynamic transaction.

When custom logic is required, you need to write some ABAP code.

And if standard IDoc is available, use ALE.

## References

### ■ Web

- ▶ Support statement –

<http://www-1.ibm.com/support/docview.wss?uid=swg21178358>

- ▶ WebSphere Adapter for mySAP.com information center –

▶ [http://publib.boulder.ibm.com/infocenter/dmndhelp/v6rxmx/topic/com.ibm.ws.adapters602.jca\\_sap.doc/doc/stsap\\_welcome.html](http://publib.boulder.ibm.com/infocenter/dmndhelp/v6rxmx/topic/com.ibm.ws.adapters602.jca_sap.doc/doc/stsap_welcome.html)



For more information, see these references.

## 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:

AX IBM WebSphere

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

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

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

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 or changes in the products or programs 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 NON-INFRINGEMENT. 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 (for example, 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 2007. 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.

