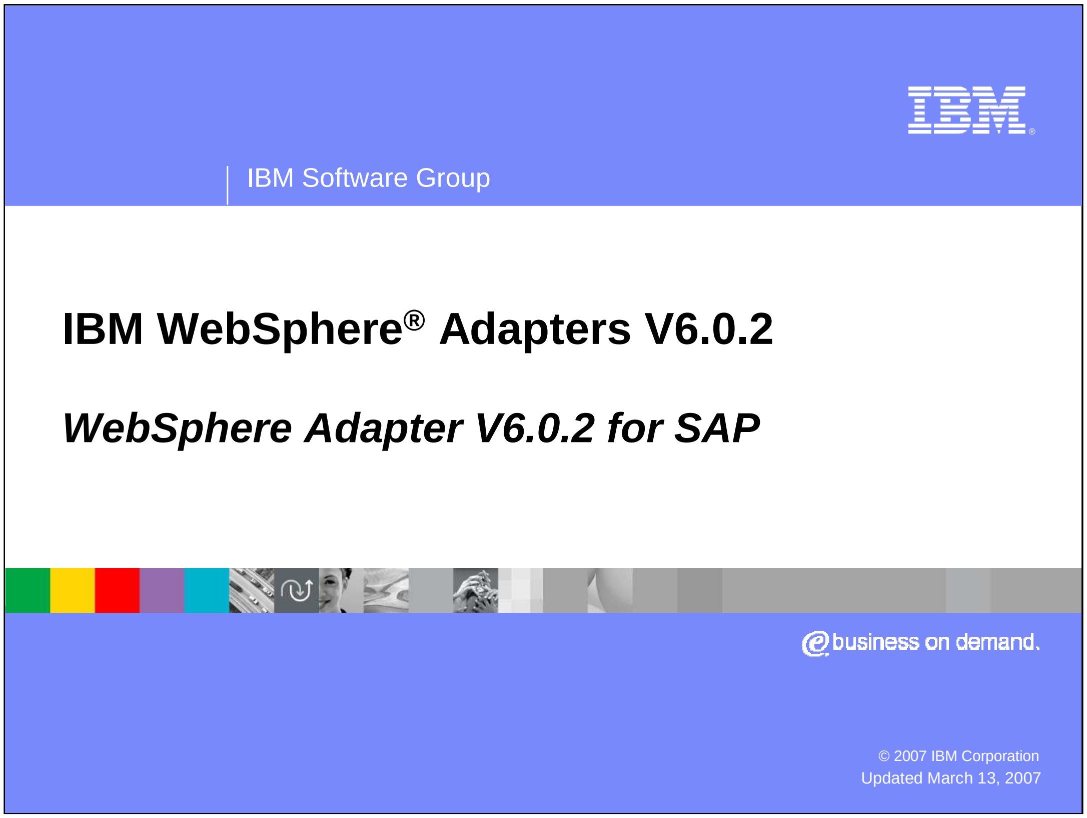


IBM Software Group

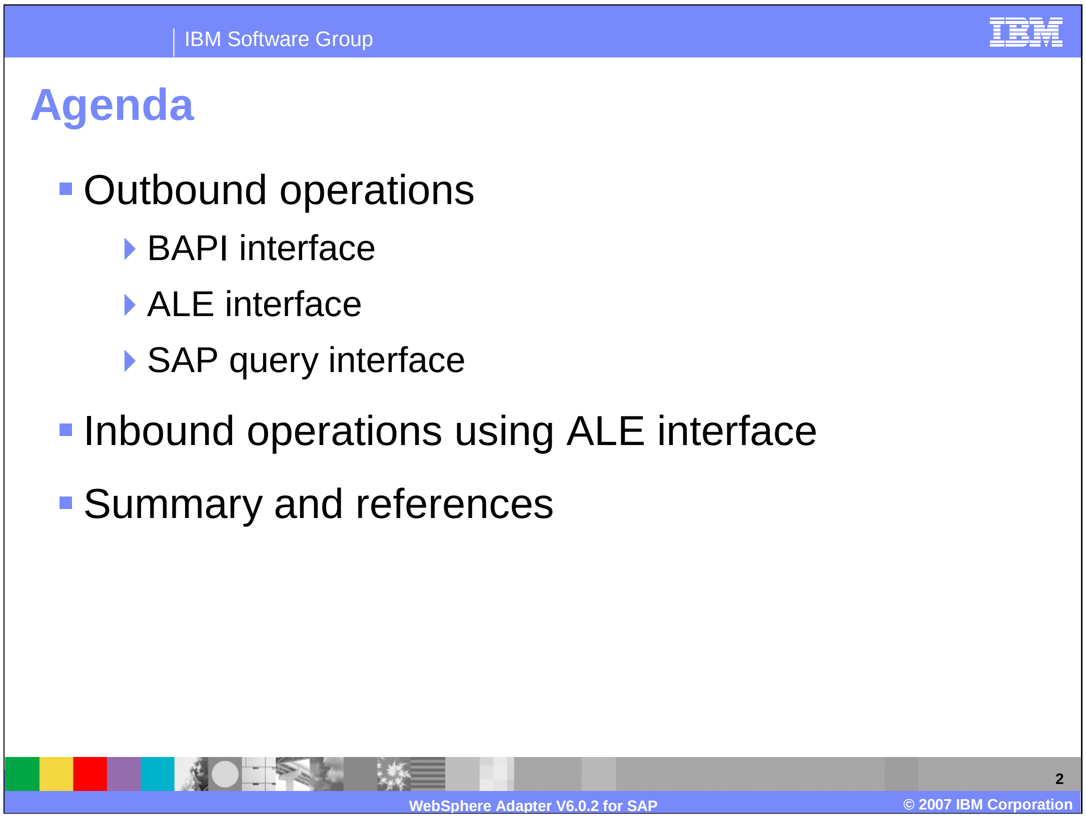
# IBM WebSphere® Adapters V6.0.2

## *WebSphere Adapter V6.0.2 for SAP*

business on demand.

Updated March 13, 2007

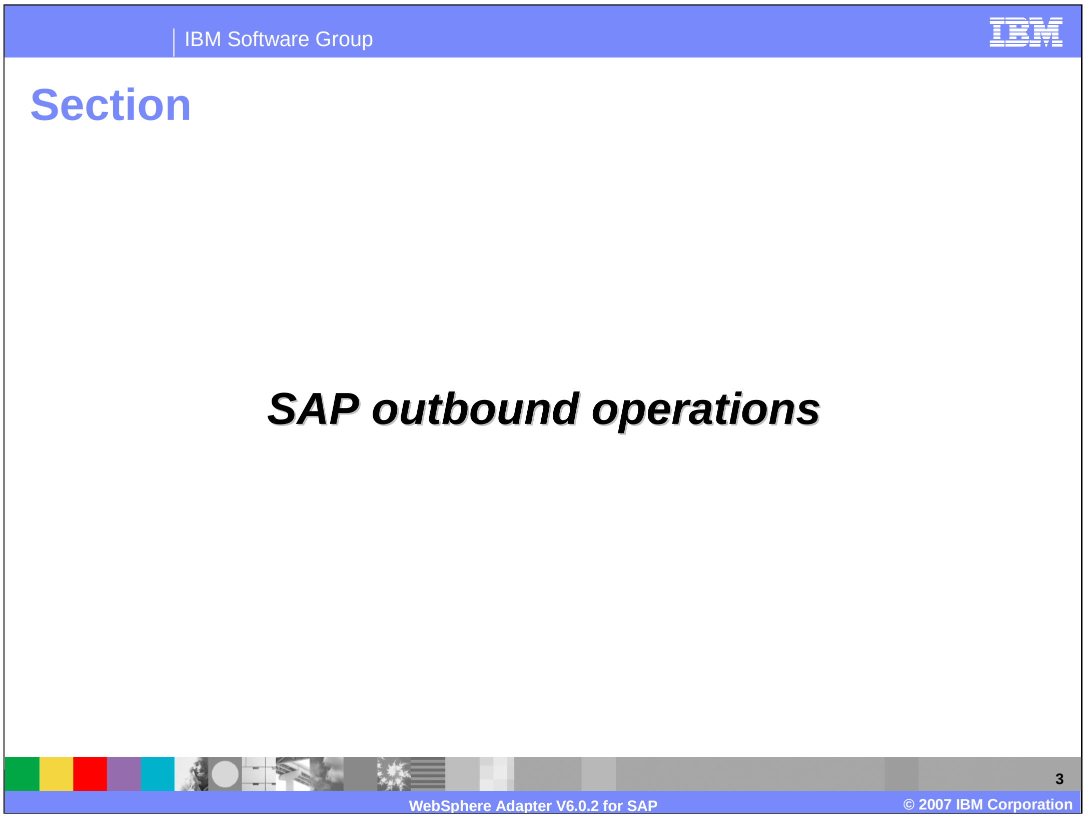
This presentation will focus on the WebSphere Adapter for SAP V6.0.2



## Agenda

- Outbound operations
  - BAPI interface
  - ALE interface
  - SAP query interface
- Inbound operations using ALE interface
- Summary and references

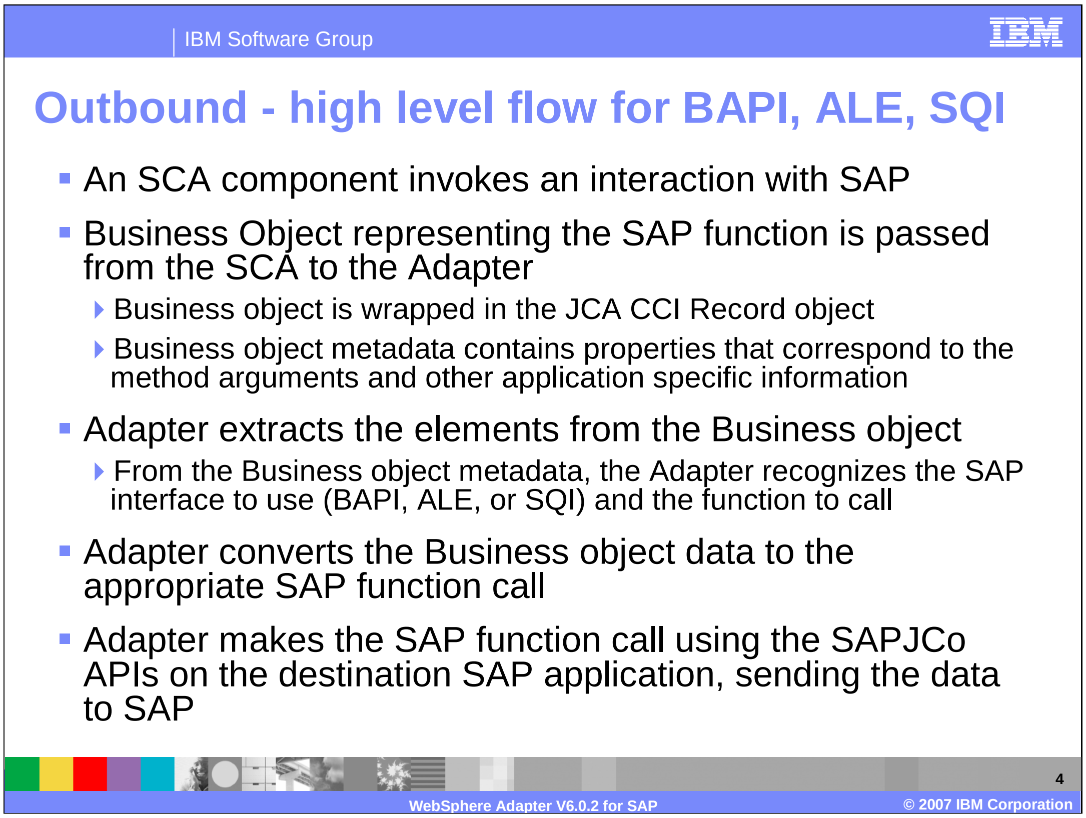
The agenda for this presentation is shown here.



## Section

***SAP outbound operations***

This section covers the outbound operations.

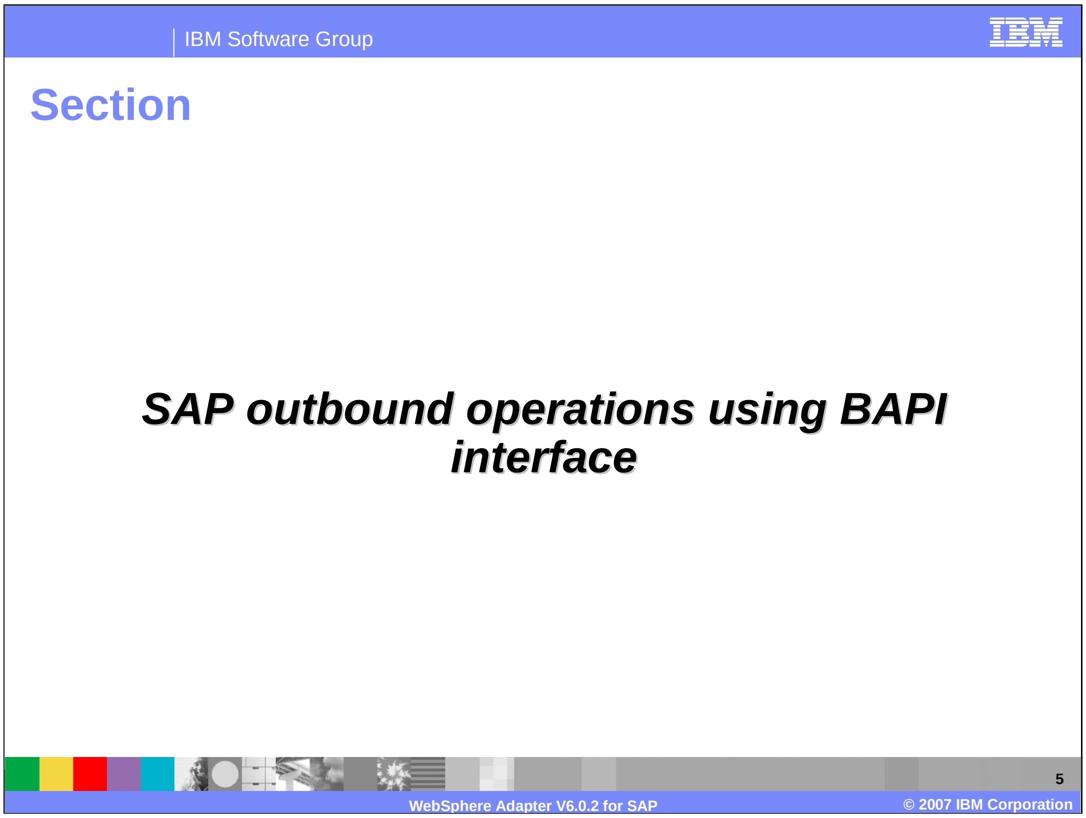


## Outbound - high level flow for BAPI, ALE, SQI

- An SCA component invokes an interaction with SAP
- Business Object representing the SAP function is passed from the SCA to the Adapter
  - Business object is wrapped in the JCA CCI Record object
  - Business object metadata contains properties that correspond to the method arguments and other application specific information
- Adapter extracts the elements from the Business object
  - From the Business object metadata, the Adapter recognizes the SAP interface to use (BAPI, ALE, or SQI) and the function to call
- Adapter converts the Business object 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 the BAPI, ALE or SQI interface is described here.

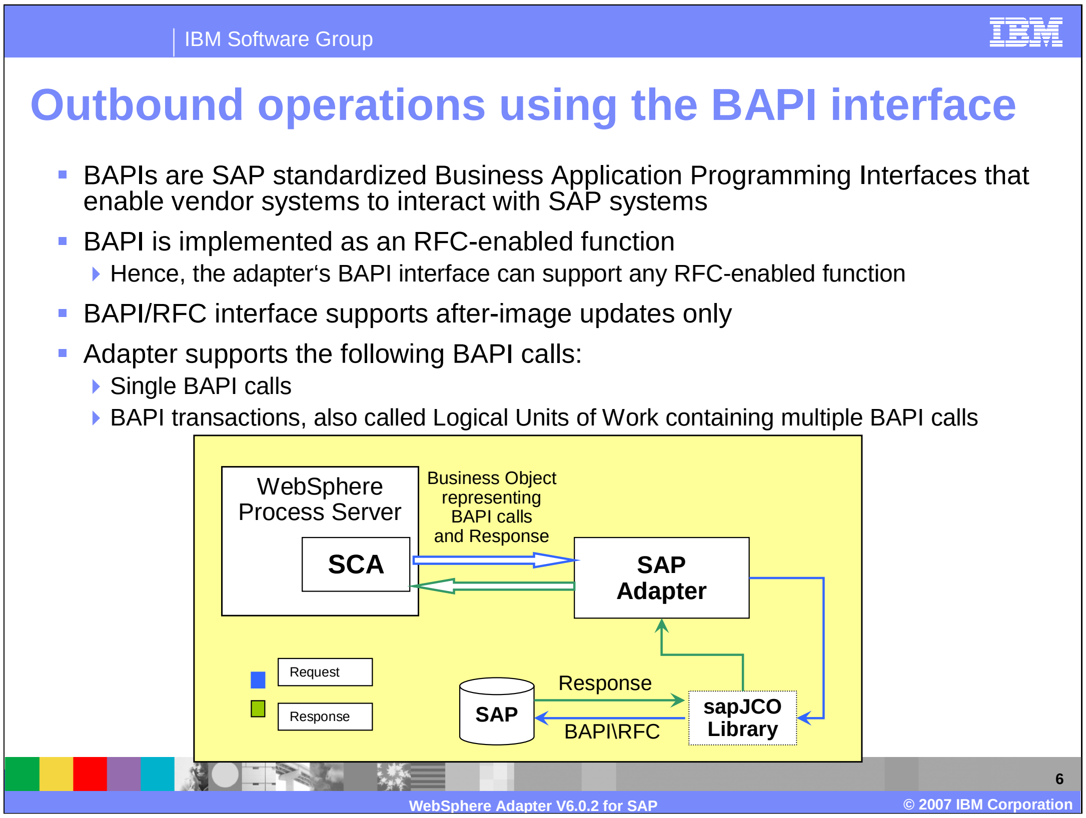
The SAP function call and its parameters and other attributes are modeled using a Business Object. The Business Objects are generated by the Enterprise metadata discovery wizard within WebSphere Integration Developer. The SCA client under the covers wraps the Business object in a JCA CCI Record object. The Adapter extracts the Business object and its metadata from the CCI Record object. It then determines the type of call, BAPI, ALE or SQI, the SAP function name, and the function attributes. Using the SAPJCo APIs, the adapter makes the SAP function call to the target SAP application.



## Section

### *SAP outbound operations using BAPI interface*

This section covers the outbound operations using the BAPI interface.



IBM Software Group

# Outbound operations using the BAPI interface

- BAPIs are SAP standardized Business Application Programming Interfaces that enable vendor 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

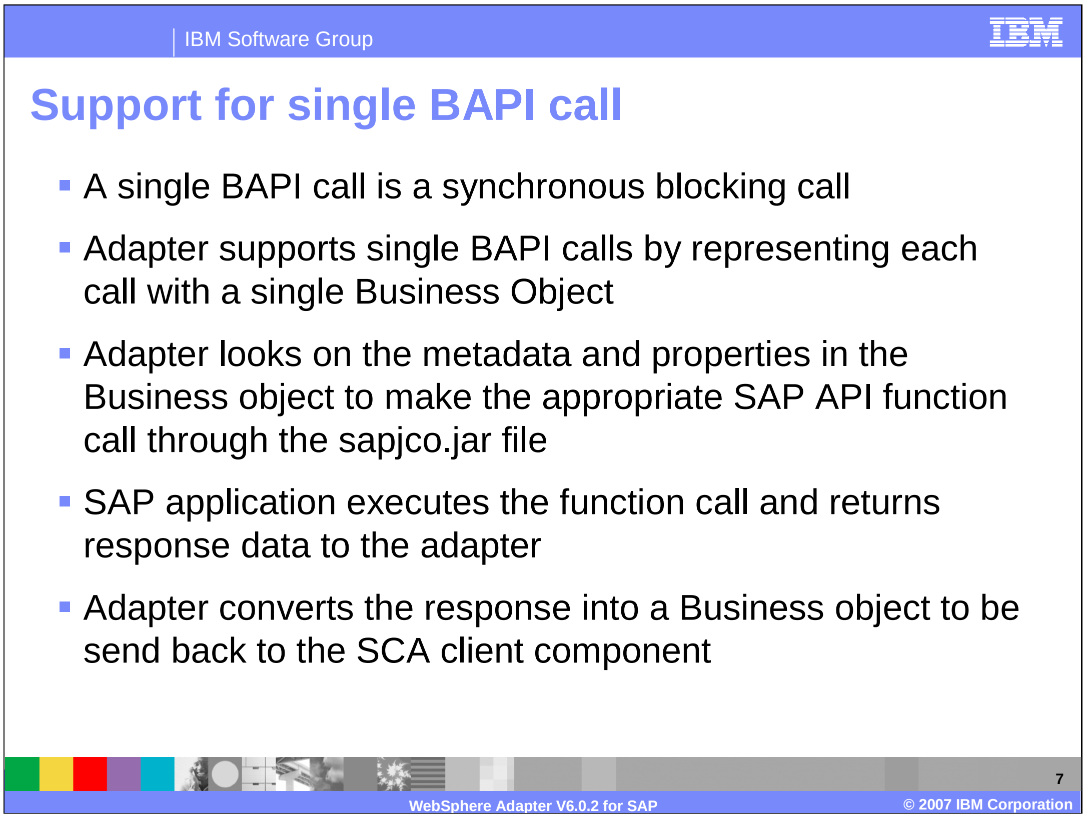
6

WebSphere Adapter V6.0.2 for SAP

© 2007 IBM Corporation

The BAPI interface APIs allow the vendor systems to interact with SAP. the 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.

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



IBM Software Group

## 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
- Adapter looks on the metadata and properties in the Business object 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 Business object to be send back to the SCA client component

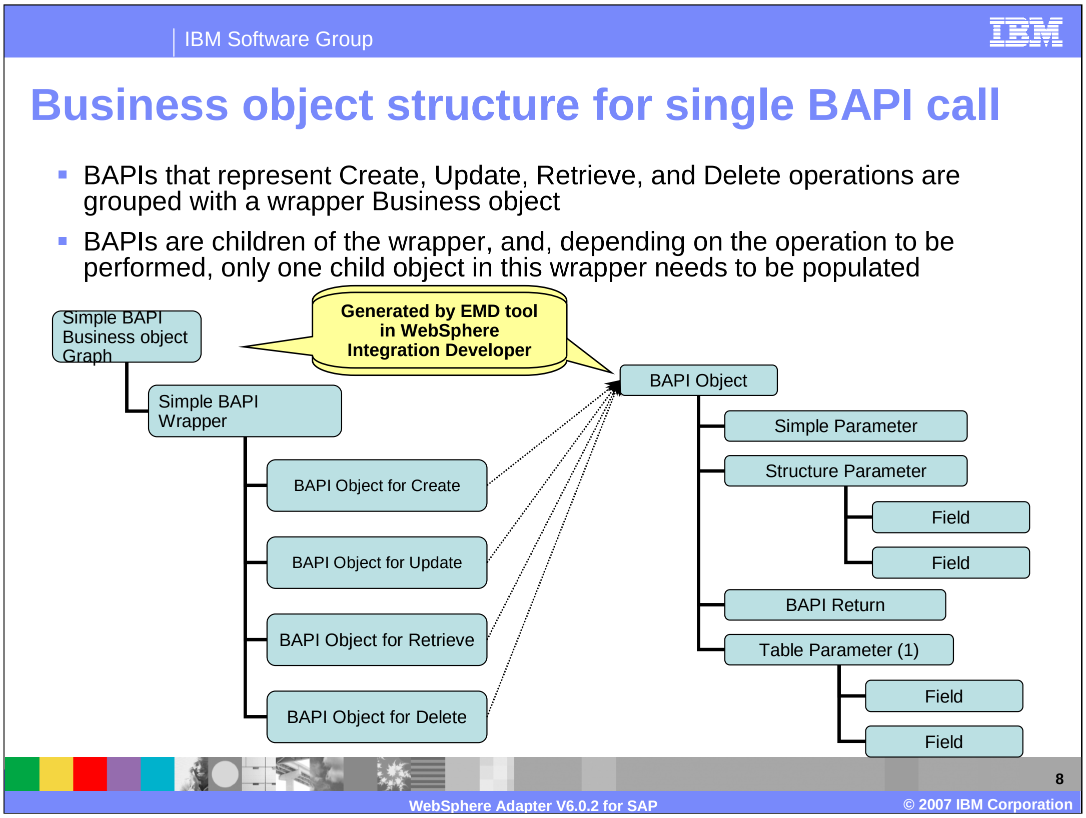
7

WebSphere Adapter V6.0.2 for SAP

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.

The SCA client wraps the Business object within the J2C CCI Record object. The adapter extracts the Business object and looks at the metadata of the Business object to determined the SAP function call.   The SAP application runs  the function call and returns response data to the adapter. The adapter converts the response to a Business object to be sent back to the SCA client.



IBM Software Group

## Business object structure for single BAPI call

- BAPIs that represent Create, Update, Retrieve, and Delete operations are grouped with a wrapper Business object
- 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

**Generated by EMD tool in WebSphere Integration Developer**

Simple BAPI Business object Graph

Simple BAPI Wrapper

BAPI Object for Create

BAPI Object for Update

BAPI Object for Retrieve

BAPI Object for Delete

BAPI Object

Simple Parameter

Structure Parameter

Field

Field

BAPI Return

Table Parameter (1)

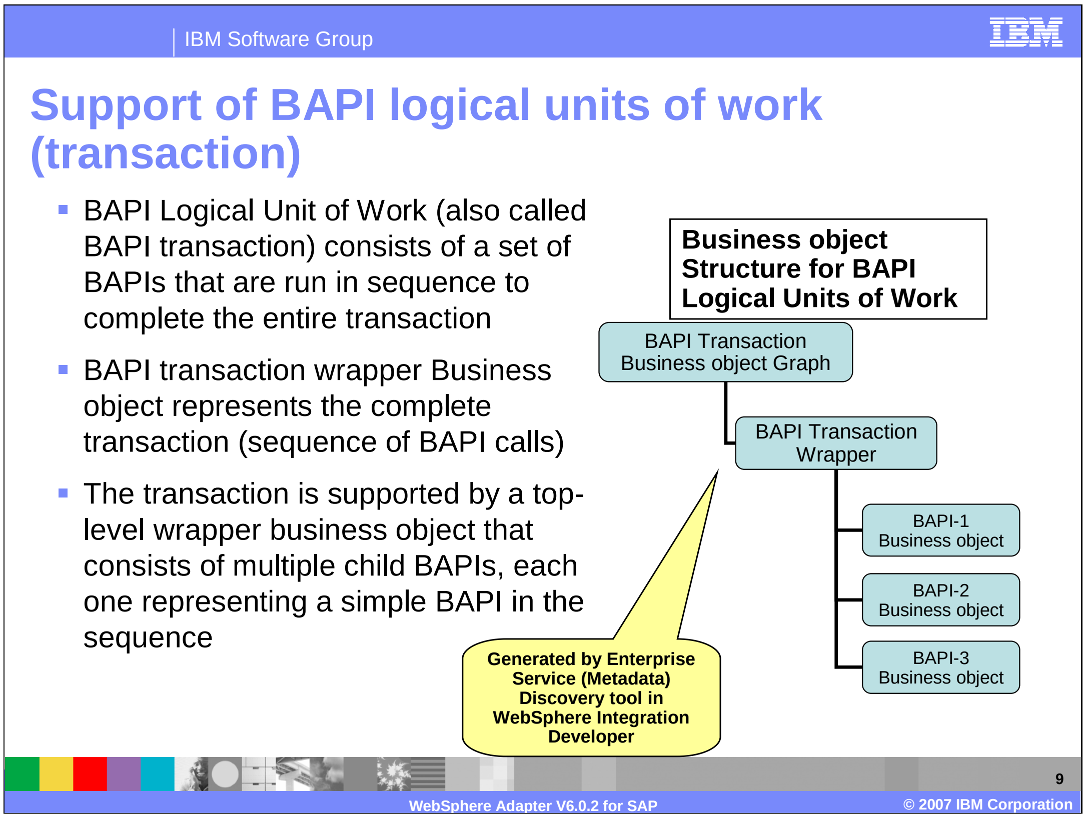
Field

Field

8

WebSphere Adapter V6.0.2 for SAP

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



IBM Software Group

# 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 run in sequence to complete the entire transaction
- BAPI transaction wrapper Business object 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

**Business object Structure for BAPI Logical Units of Work**

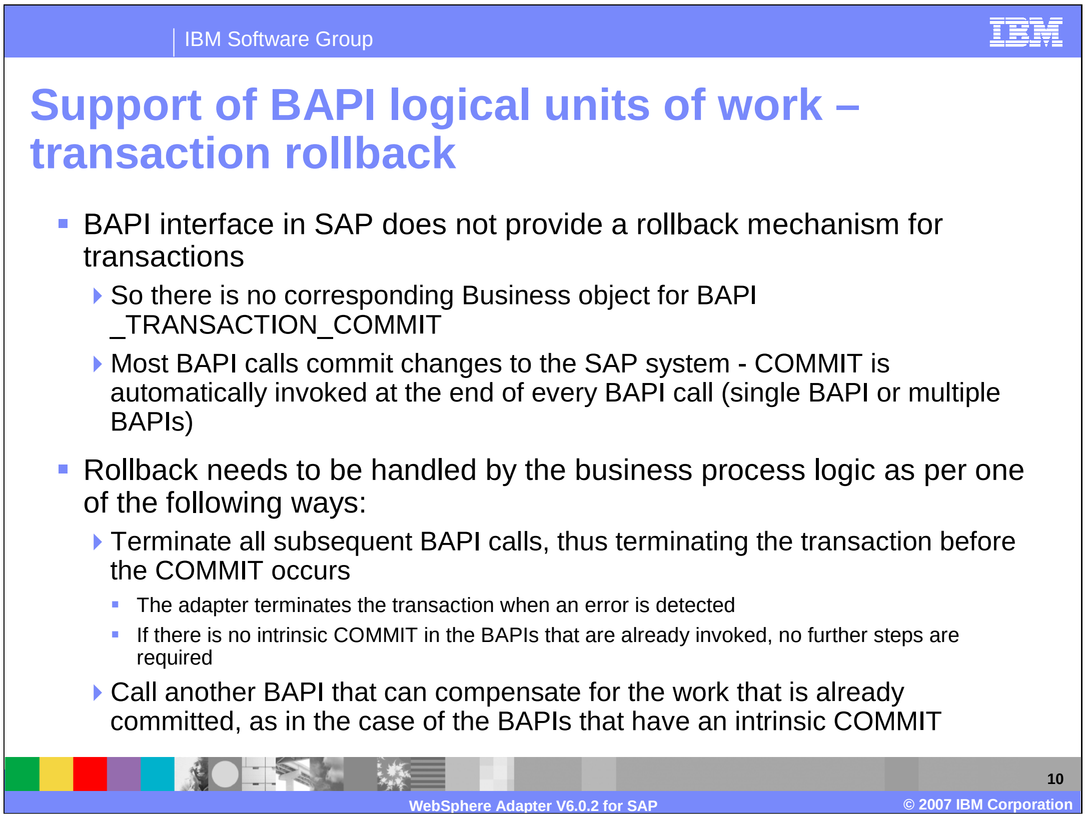
**Generated by Enterprise Service (Metadata) Discovery tool in WebSphere Integration Developer**

9

WebSphere Adapter V6.0.2 for SAP

This slide 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 Business object, as shown in the diagram as BAPI-1 Business object, BAPI-2 Business object 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), and application specific information to determine the sequence of the BAPI calls.



IBM Software Group

## 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 Business object 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

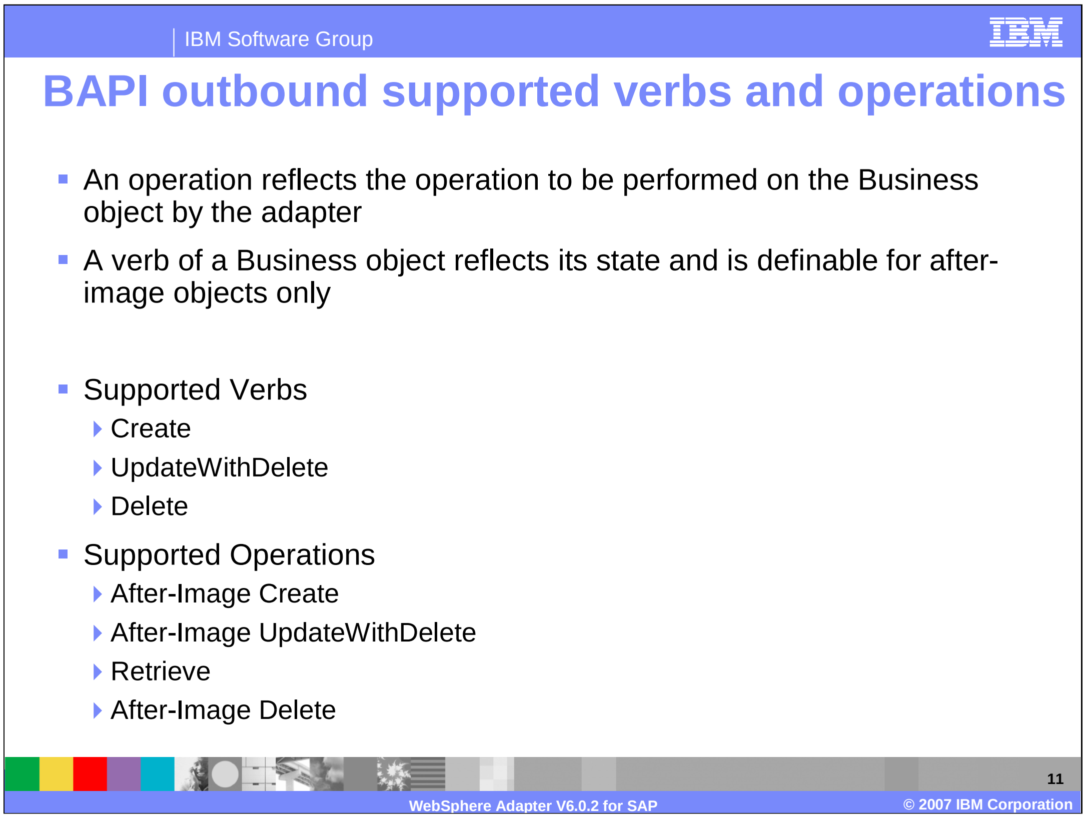
10

WebSphere Adapter V6.0.2 for SAP

© 2007 IBM Corporation

This slide outlines support for BAPI logical units of work.  BAPI interface in SAP does not provide a rollback mechanism for transaction.  Thus, there is no corresponding Business object for BAPI _TRANSACTION_COMMIT.  Since most of 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. Terminate all subsequent BAPI calls, thus terminating the transaction before the COMMIT occurs.  And 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.

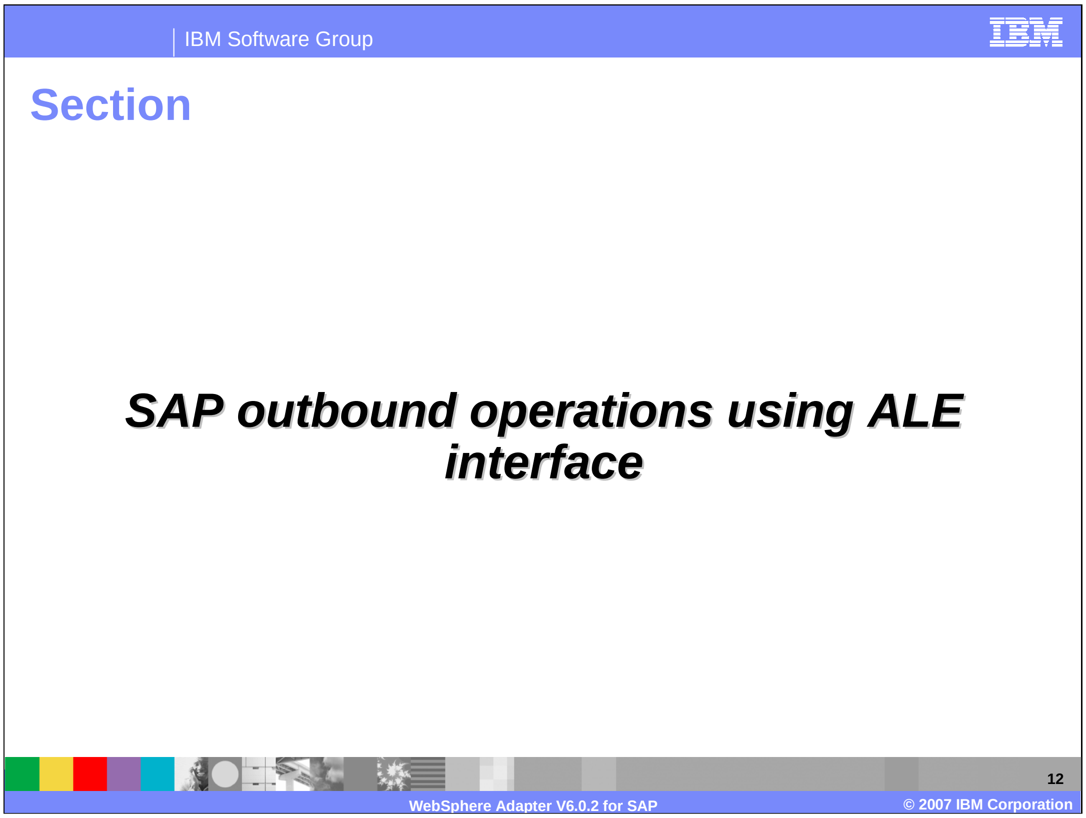


## BAPI outbound supported verbs and operations

- An operation reflects the operation to be performed on the Business object by the adapter
- A verb of a Business object 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

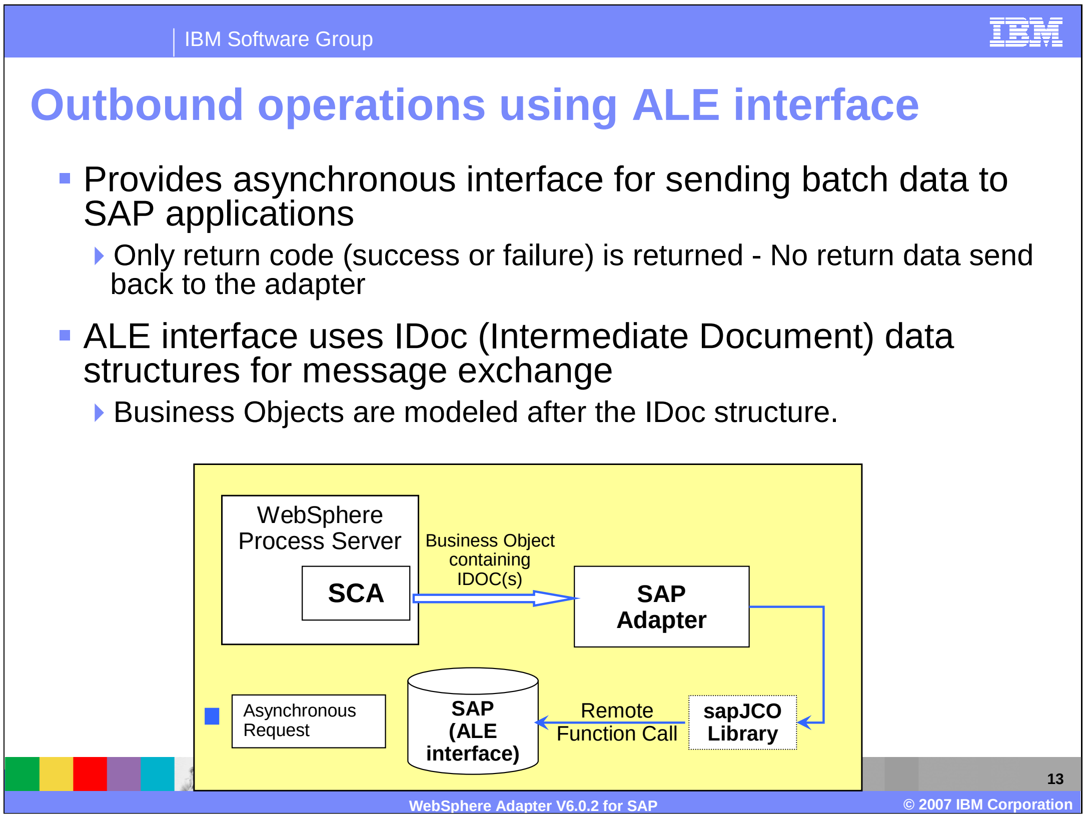
The supported verbs and operations for BAPI outbound calls are listed on this slide.



## Section

# *SAP outbound operations using ALE interface*

This section covers the outbound operations using the ALE interface.



IBM Software Group

## 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
  - Business Objects are modeled after the IDoc structure.

WebSphere Adapter V6.0.2 for SAP

13

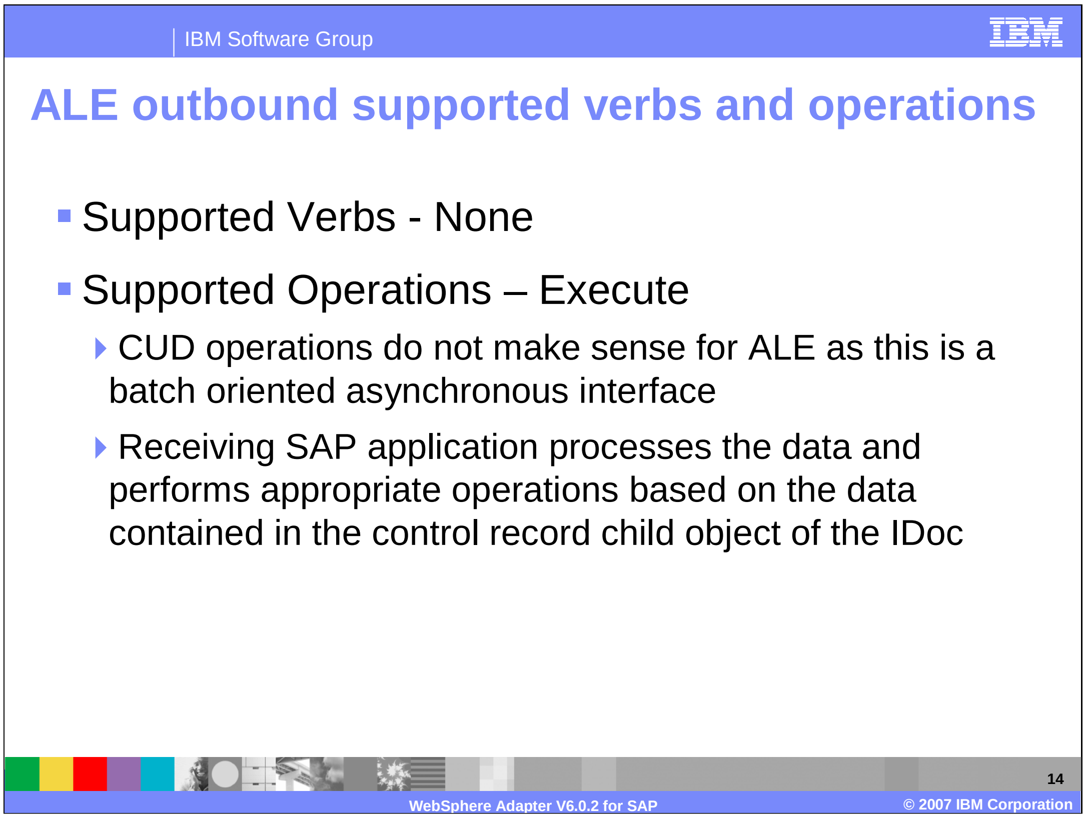
© 2007 IBM Corporation

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

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

The adapters extracts the IDoc from the Business object and then follows these four steps:

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.



IBM Software Group

## ALE outbound supported verbs and operations

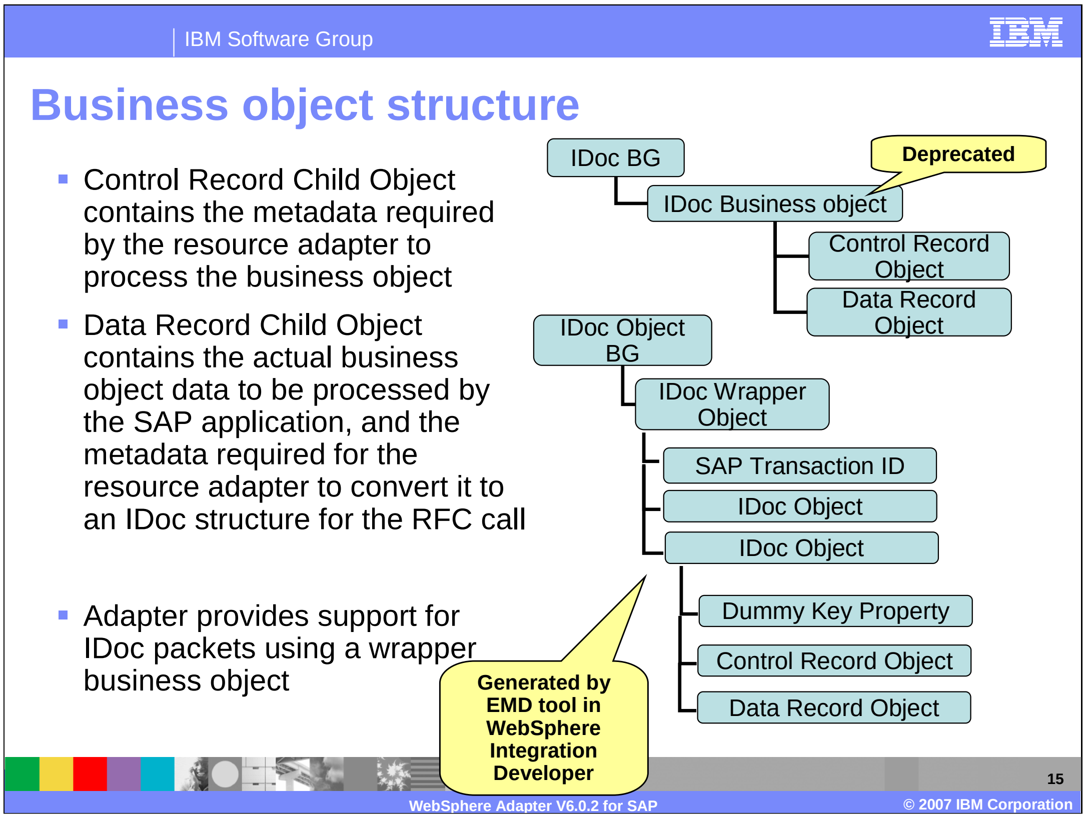
- Supported Verbs - None
- Supported Operations – Execute
  - CUD operations do not make sense for ALE as this is a batch oriented asynchronous interface
  - Receiving SAP application processes the data and performs appropriate operations based on the data contained in the control record child object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

14

WebSphere Adapter V6.0.2 for SAP

© 2007 IBM Corporation

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



IBM Software Group

## Business object 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

IDoc BG

IDoc Business object

Deprecated

Control Record Object

Data Record Object

IDoc Object BG

IDoc Wrapper Object

SAP Transaction ID

IDoc Object

IDoc Object

Dummy Key Property

Control Record Object

Data Record Object

**Generated by EMD tool in WebSphere Integration Developer**

15

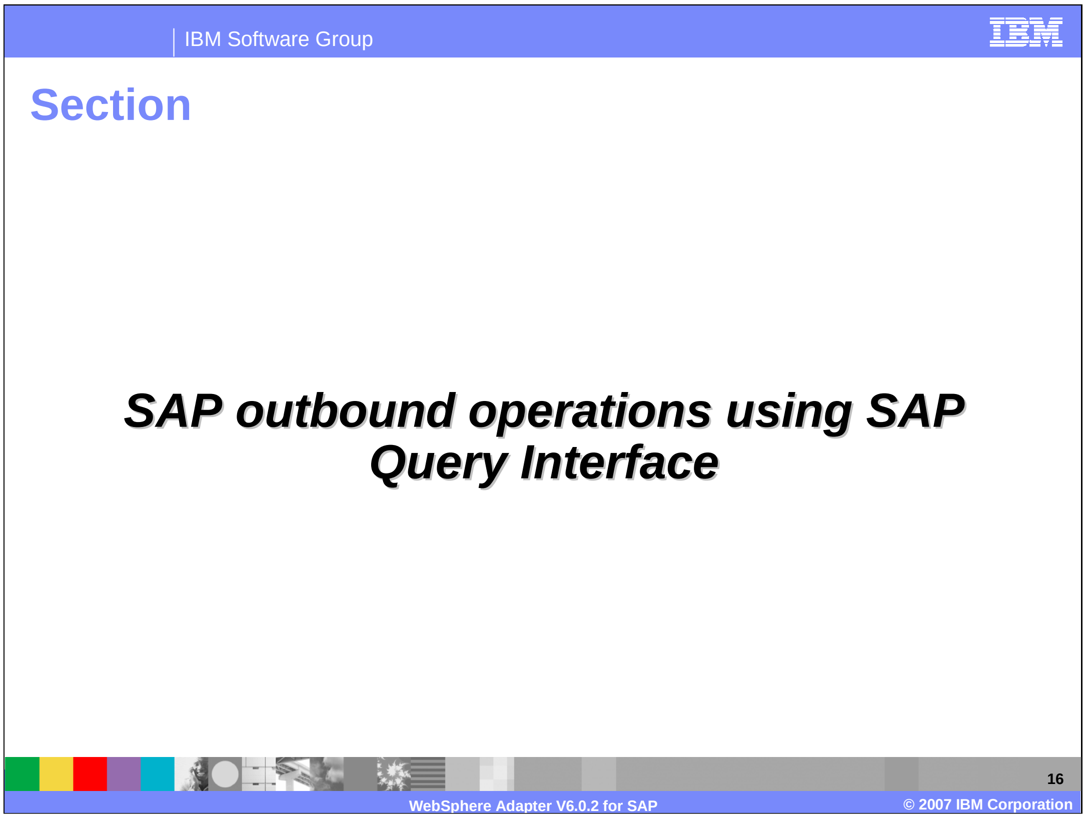
WebSphere Adapter V6.0.2 for SAP

© 2007 IBM Corporation

The diagram on the top of this slide shows the simple IDoc Business object which has been deprecated.

The diagram object on the bottom of the slide shows the support for IDoc packets, where each IDoc is represented by the IDoc Business object.   The IDoc business graph has a wrapper business object that will contain instances of an IDoc business object.  For individual IDocs, the wrapper business object will contain only one instance of an IDoc business object.  For IDoc packets, the wrapper business object will contain many instances of an IDoc business object.  The IDoc Business object 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.

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



IBM Software Group

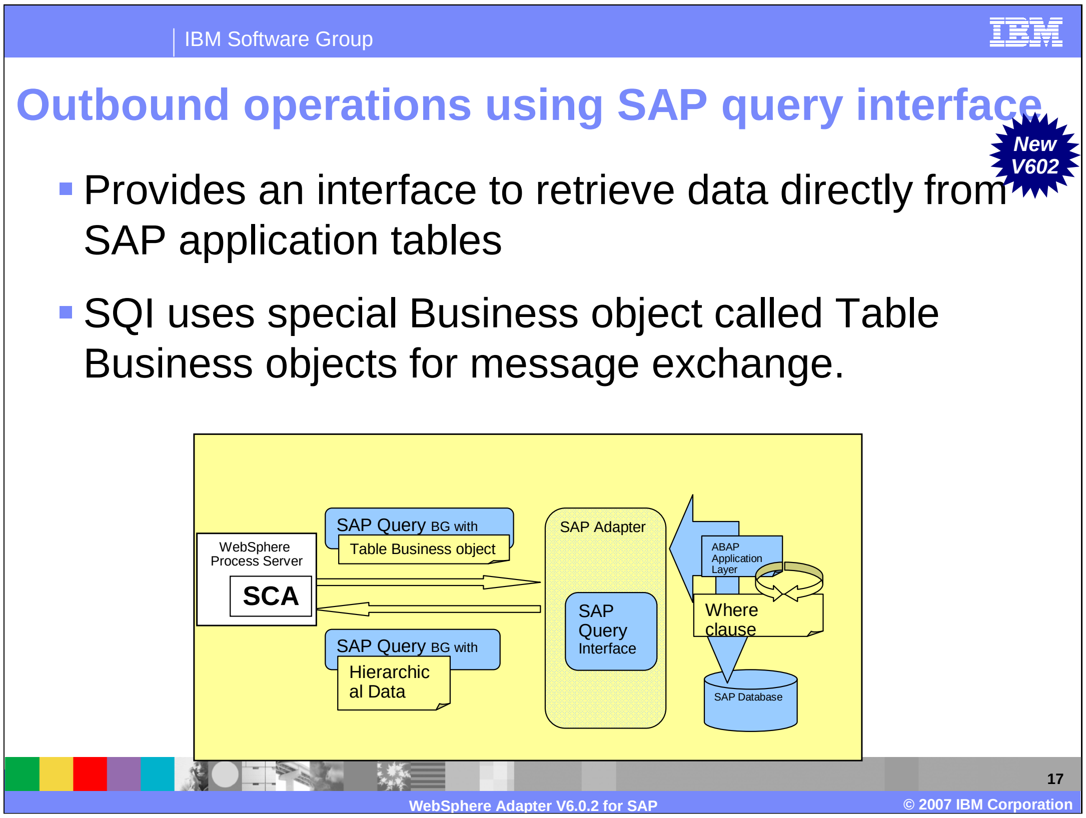
## Section

**_SAP outbound operations using SAP Query Interface_**

16

WebSphere Adapter V6.0.2 for SAP

This section covers the outbound operations supported by SAP Query Interface.



IBM Software Group

## Outbound operations using SAP query interface

New V602

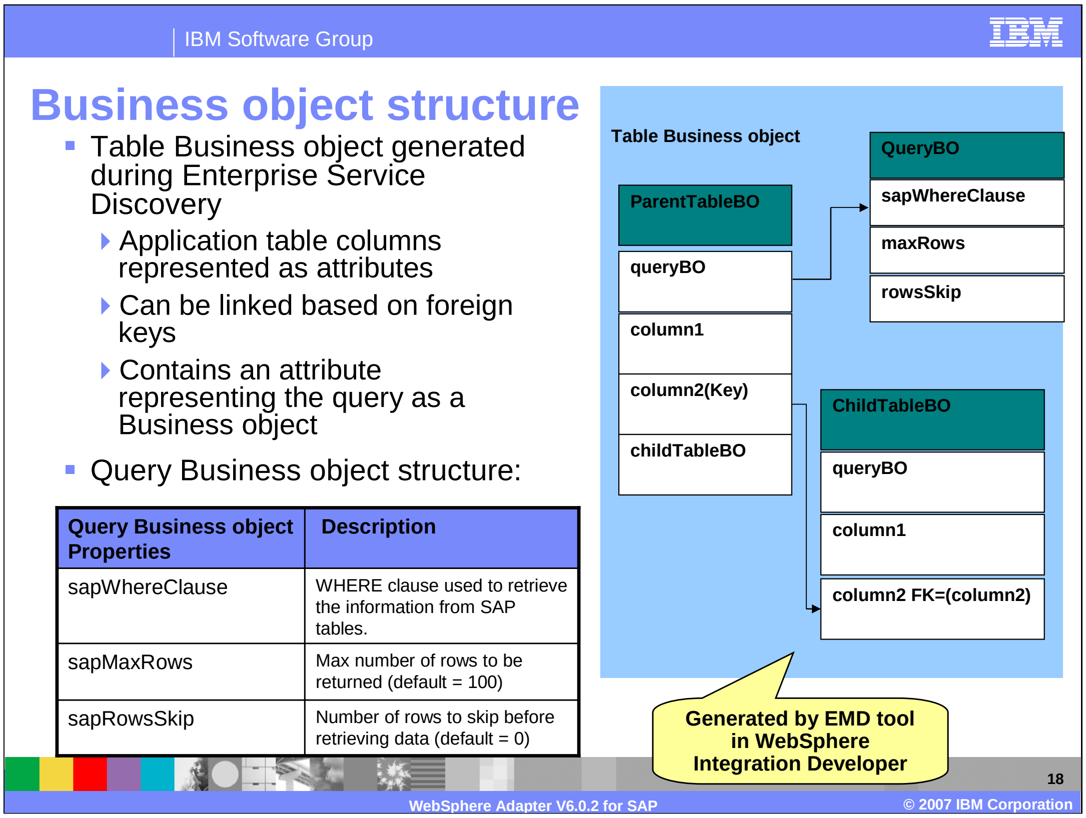
- Provides an interface to retrieve data directly from SAP application tables
- SQI uses special Business object called Table Business objects for message exchange.

17

WebSphere Adapter V6.0.2 for SAP

To access data in hierarchical form, SAP provides RFC and BAPI. However, in some certain scenarios, you may want to retrieve data in some specific hierarchical form or from SAP application tables for which SAP does not provide any interface. SAP Query Interface, which is new to this release of the SAP Adapter, supports the retrieval of data directly from the SAP Application tables that can be represented in custom hierarchical form.

The SCA client sends the Table Business Object by wrapping the Business object in a J2C CCI Record object. The adapter extracts the Table Business object from the record and then collects details about the Table to be queried, the columns that need to be retrieved and the where clause which is defined from the Application Specific Information. The query is run on the SAP table and the resulting rows are converted to a Table Business object and then inserted into a container.



IBM Software Group

# Business object structure

- Table Business object generated during Enterprise Service Discovery
  - Application table columns represented as attributes
  - Can be linked based on foreign keys
  - Contains an attribute representing the query as a Business object
- Query Business object structure:

| Query Business object Properties | Description |
|---|---|
| sapWhereClause | WHERE clause used to retrieve the information from SAP tables. |
| sapMaxRows | Max number of rows to be returned (default = 100) |
| sapRowsSkip | Number of rows to skip before retrieving data (default = 0) |

Generated by EMD tool in WebSphere Integration Developer

18

WebSphere Adapter V6.0.2 for SAP

© 2007 IBM Corporation

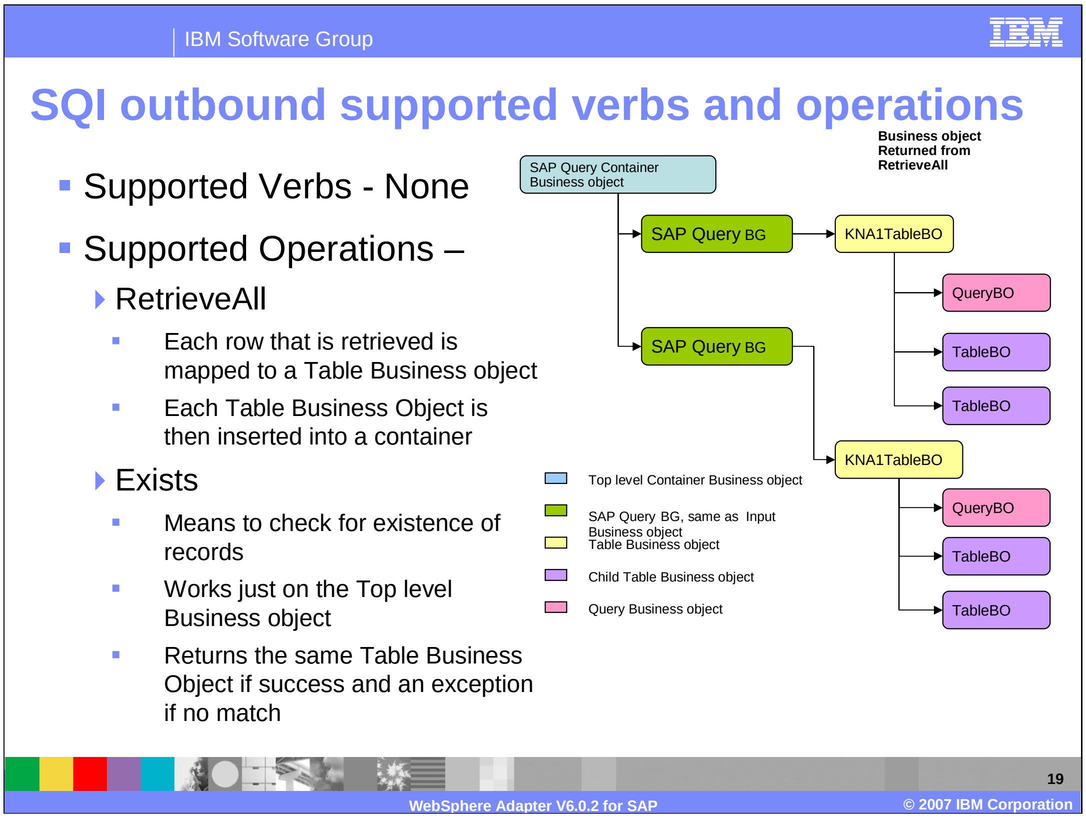
The diagram on the right side shows the structure of the Table Business object. The columns in the application table on which the query is to be run map as attributes of the Table Business object. Another important attribute of the table Business object is the Query Business object. Using the Enterprise Service Discovery, you can generate a query Business object for a SAP application table that you want to retrieve data from. Table Business Objects can be linked using foreign keys defined inside the Table Business Object to define a hierarchy of parent-child relationships. The child Table Business Object will have a foreign key that references a property in the parent query Business object.

Query Business object also has these three attributes:

1) The where clause entered using the Enterprise Service Discovery is the table's default WHERE clause. You can override the default where clause in the sapWhereClause property at runtime. The syntax of the WHERE CLAUSE is not validated before it is run. If the WHERE fails, the appropriate error is thrown and returned by the underlying ABAP API

2) The maxRows attribute defines the maximum number of rows that need to be retrieved

3) The rowsSkip attribute defines the number of rows to skip before retrieving the rows



IBM Software Group

# SQI outbound supported verbs and operations

- Supported Verbs - None
- Supported Operations –
  - RetrieveAll
    - Each row that is retrieved is mapped to a Table Business object
    - Each Table Business Object is then inserted into a container
  - Exists
    - Means to check for existence of records
    - Works just on the Top level Business object
    - Returns the same Table Business Object if success and an exception if no match

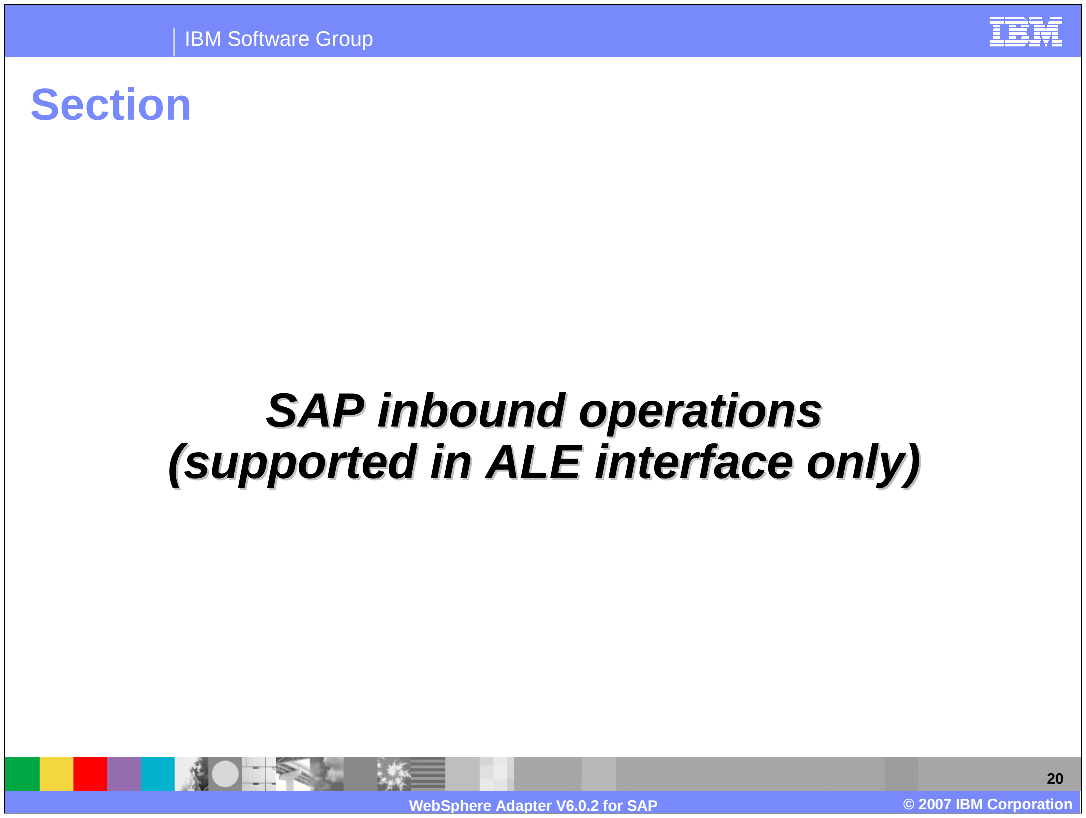
| | |
|---|---|
| 🟦 | Top level Container Business object |
| 🟩 | SAP Query BG, same as Input Business object |
| 🟨 | Table Business object |
| 🟪 | Child Table Business object |
| 🟥 | Query Business object |

19

WebSphere Adapter V6.0.2 for SAP

© 2007 IBM Corporation

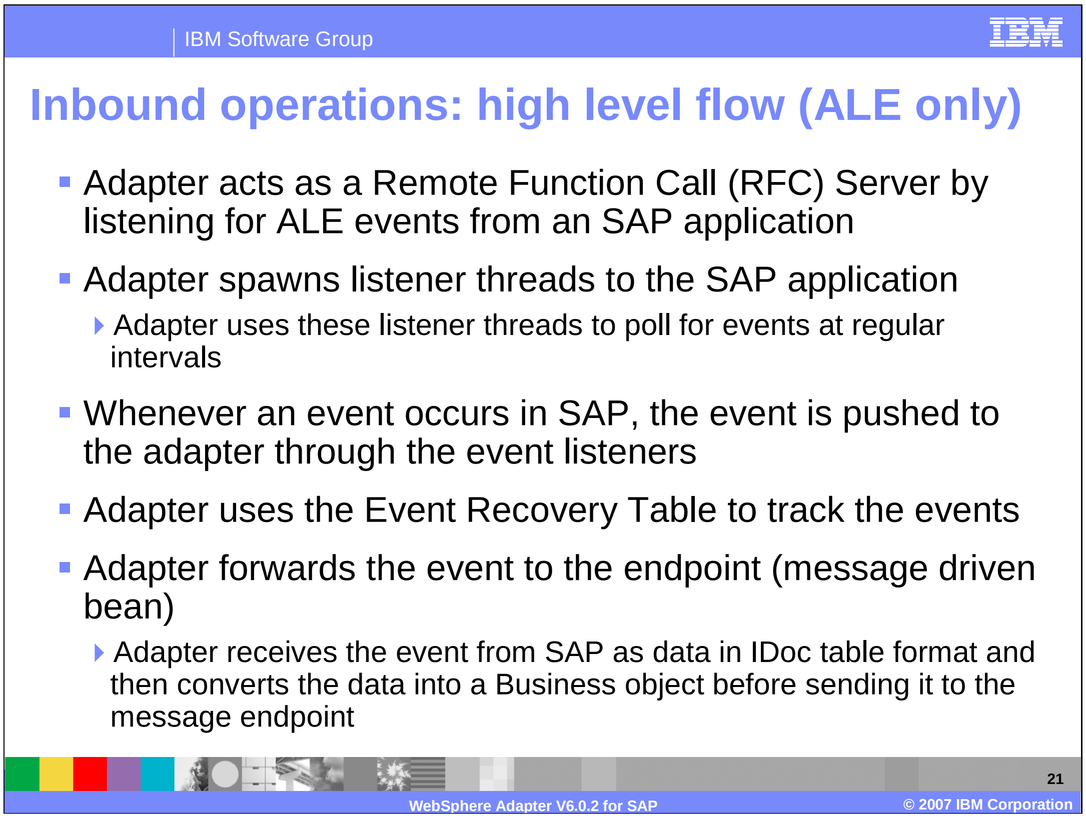
For the SAP Query Interface outbound call, there are no verbs and the supported operations for SQI are RetrieveAll and Exists. For ‘RetrieveAll’ processing, each row that is retrieved is mapped to a Table Business object and each table business object is then inserted into a container. ‘Exists’ processing checks for existence of records and works just on the top level Business object. It then returns the same table Business object if successful, and an exception if there is no match.



## Section

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

This section covers the inbound operations. The inbound operations are supported only using the ALE interface.



IBM Software Group

## 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 through 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 Business object before sending it to the message endpoint

21

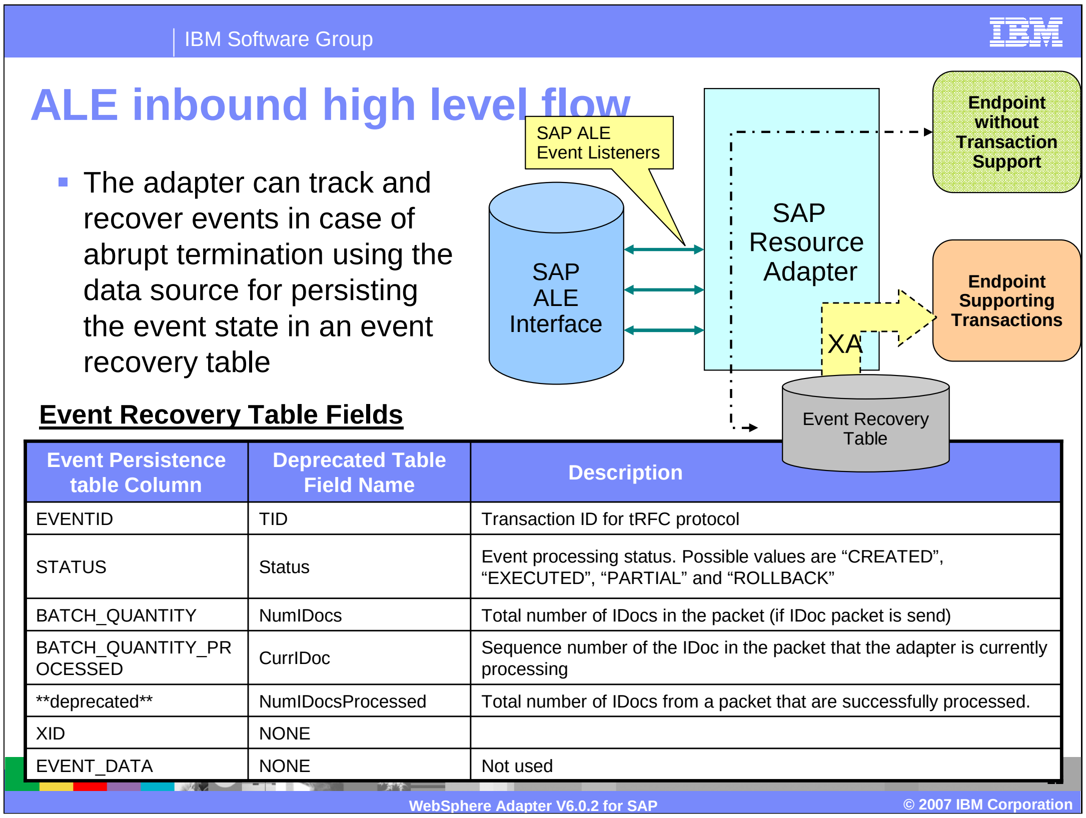
WebSphere Adapter V6.0.2 for SAP

© 2007 IBM Corporation

The adapter spawns a listener thread to the SAP system. The listener thread regularly polls 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 through the event listeners. There is no event store within the SAP application. The adapter uses a persistent event recovery table to track the events.

The adapter receives the event from SAP as data in IDoc table format and then converts the data into a Business object before sending it to the endpoint.



IBM Software Group

## 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**

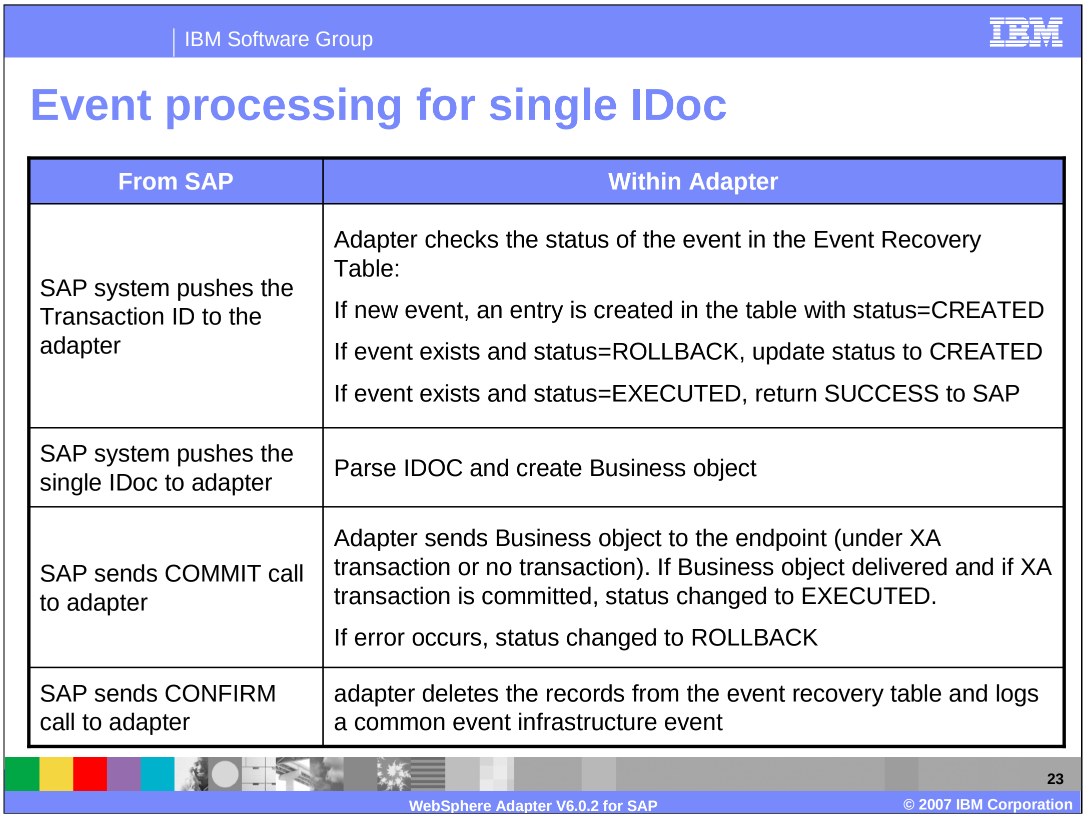
| Event Persistence table Column | Deprecated Table Field Name | Description |
|---|---|---|
| EVENTID | TID | Transaction ID for tRFC protocol |
| STATUS | Status | Event processing status. Possible values are “CREATED”, “EXECUTED”, “PARTIAL” and “ROLLBACK” |
| BATCH_QUANTITY | NumIDocs | Total number of IDocs in the packet (if IDoc packet is send) |
| BATCH_QUANTITY_PROCESSED | CurrIDoc | Sequence number of the IDoc in the packet that the adapter is currently processing |
| **deprecated** | NumIDocsProcessed | Total number of IDocs from a packet that are successfully processed. |
| XID | NONE | |
| EVENT_DATA | NONE | Not used |

WebSphere Adapter V6.0.2 for SAP

© 2007 IBM Corporation

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.

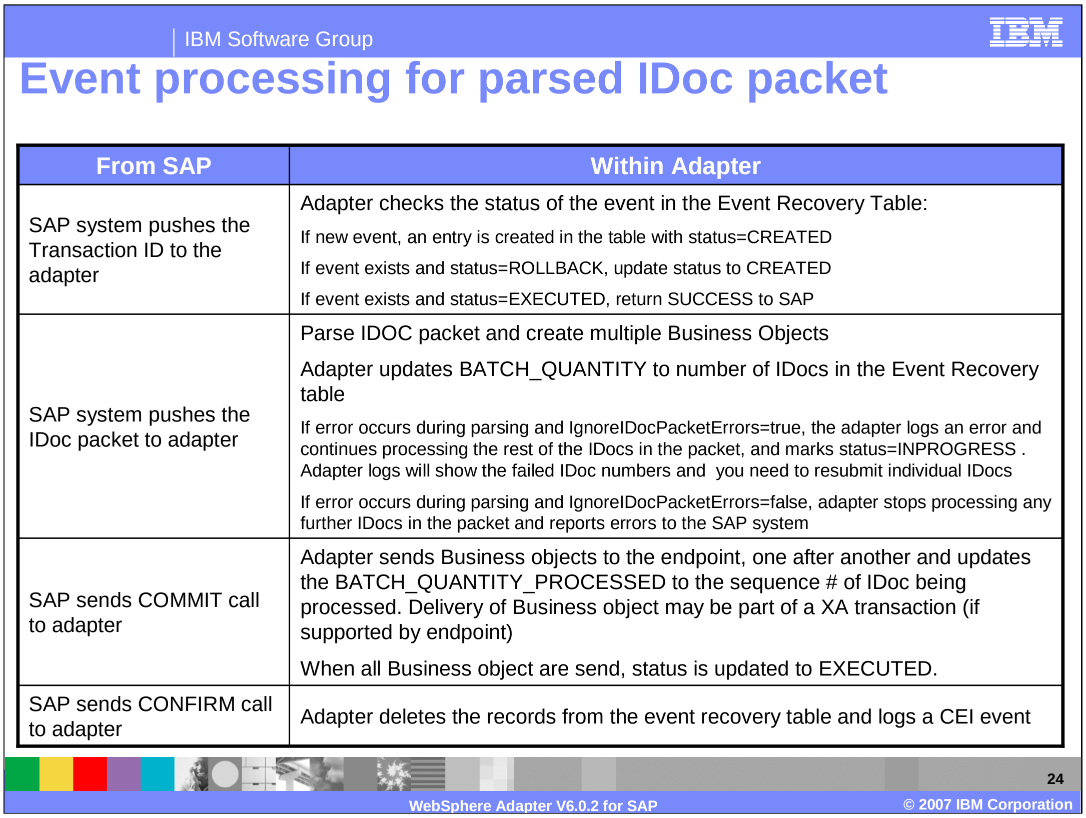
The schema of the event recovery table is shown. The EVENTID field is the unique transaction ID that the SAP application uses for events. SAP will send the same Transaction ID for subsequent processing of the same event. An event may represent an IDoc packet. The table has fields for the total number 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 to the adapter | Adapter checks the status of the event in the Event Recovery Table:<br>If new event, an entry is created in the table with status=CREATED<br>If event exists and status=ROLLBACK, update status to CREATED<br>If event exists and status=EXECUTED, return SUCCESS to SAP |
| SAP system pushes the single IDoc to adapter | Parse IDOC and create Business object |
| SAP sends COMMIT call to adapter | Adapter sends Business object to the endpoint (under XA transaction or no transaction). If Business object delivered and if XA transaction is committed, status changed to EXECUTED.<br>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 event |

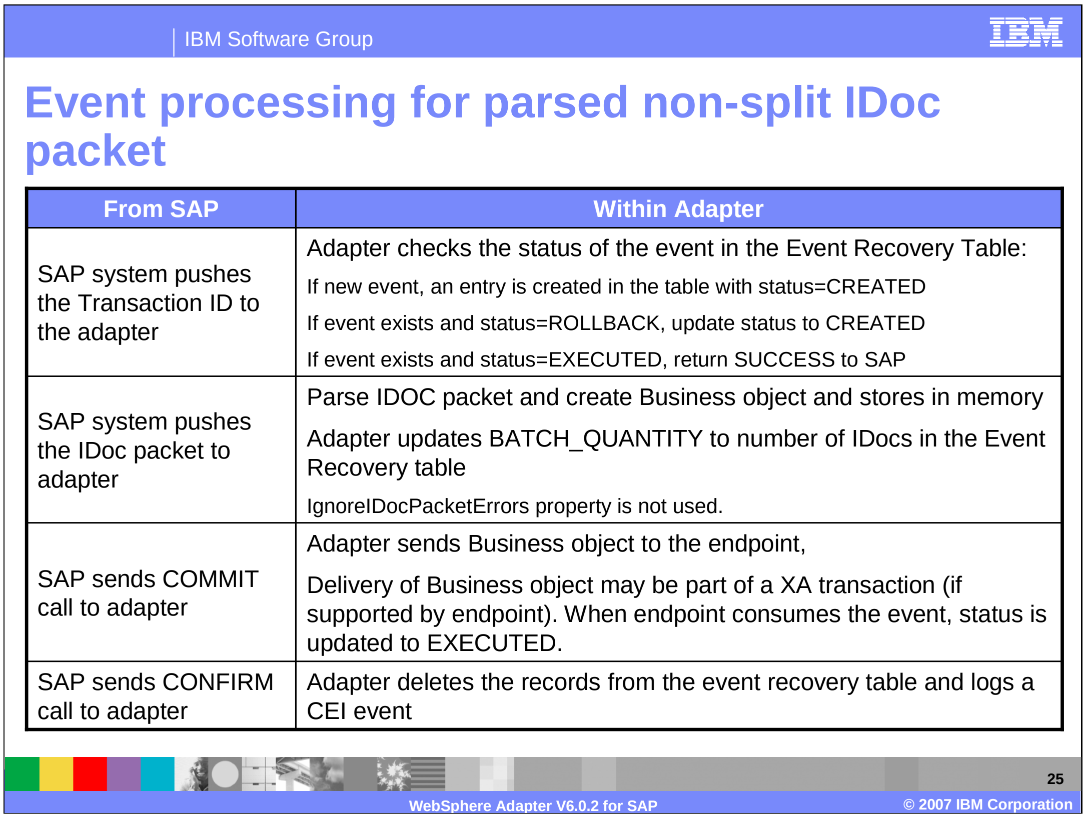
This table shows the actions that occur on the SAP side and within the adapter when processing a Single IDoc.



# Event processing for parsed IDoc packet

| From SAP | Within Adapter |
|---|---|
| SAP system pushes the Transaction ID to the adapter | Adapter checks the status of the event in the Event Recovery Table:<br>If new event, an entry is created in the table with status=CREATED<br>If event exists and status=ROLLBACK, update status to CREATED<br>If event exists and status=EXECUTED, return SUCCESS to SAP |
| SAP system pushes the IDoc packet to adapter | Parse IDOC packet and create multiple Business Objects<br>Adapter updates BATCH_QUANTITY to number of IDocs in the Event Recovery table<br>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=INPROGRESS . Adapter logs will show the failed IDoc numbers and you need to resubmit individual IDocs<br>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 Business objects to the endpoint, one after another and updates the BATCH_QUANTITY_PROCESSED to the sequence # of IDoc being processed. Delivery of Business object may be part of a XA transaction (if supported by endpoint)<br>When all Business object are send, status is updated to EXECUTED. |
| 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. 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 you terminate the adapter manually while processing an IDoc.

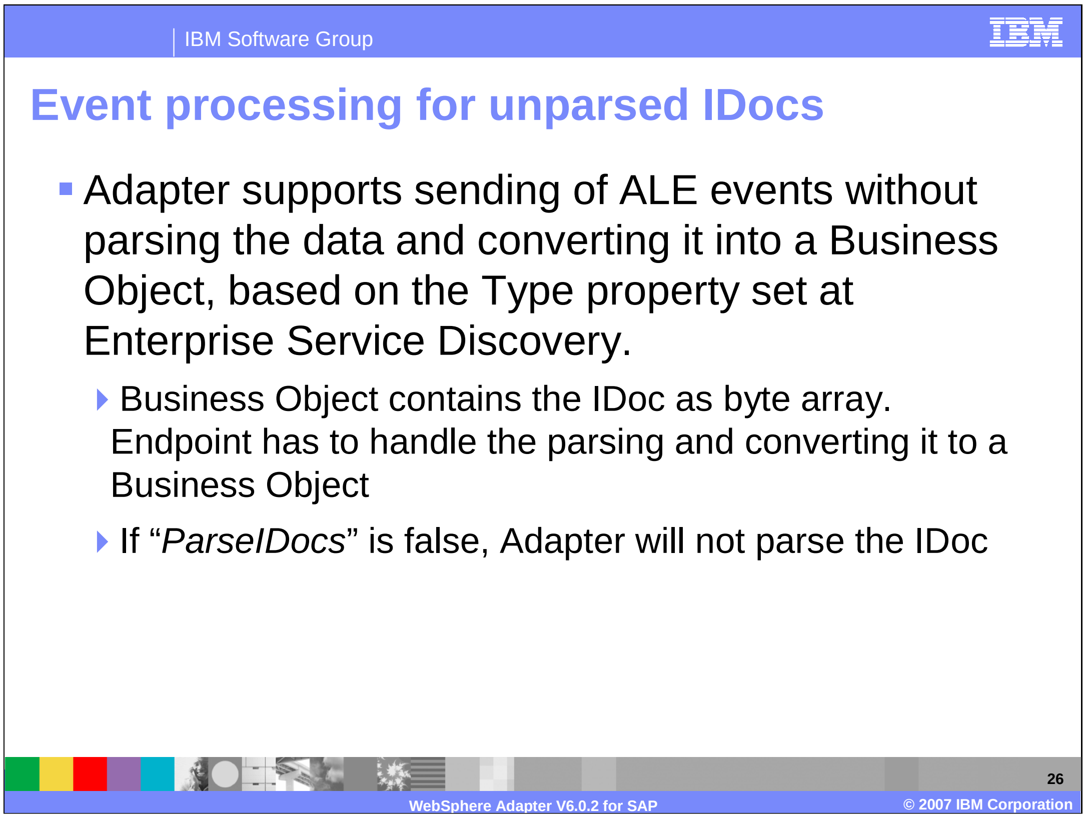


# Event processing for parsed non-split IDoc packet

| From SAP | Within Adapter |
|---|---|
| SAP system pushes the Transaction ID to the adapter | Adapter checks the status of the event in the Event Recovery Table:<br>If new event, an entry is created in the table with status=CREATED<br>If event exists and status=ROLLBACK, update status to CREATED<br>If event exists and status=EXECUTED, return SUCCESS to SAP |
| SAP system pushes the IDoc packet to adapter | Parse IDOC packet and create Business object and stores in memory<br>Adapter updates BATCH_QUANTITY to number of IDocs in the Event Recovery table<br>IgnoreIDocPacketErrors property is not used. |
| SAP sends COMMIT call to adapter | Adapter sends Business object to the endpoint,<br>Delivery of Business object may be part of a XA transaction (if supported by endpoint). When endpoint consumes the event, status is updated to EXECUTED. |
| 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 non-split IDoc packet. One case scenario for this feature is for maintaining the transaction of data if inventory levels for a set of materials need to be updated at one time. This feature will most likely increase memory usage since the whole IDoc packet will be stored in memory. The adapter can handle processing an IDoc packet as one unit of work if all the IDocs in the packet are of the same IDoc type

The event processing is similar to a single IDoc with respect to the event recovery mechanism. The IDoc packet is considered a single unit of work. So, a failure registered by the endpoint for any one IDoc fails the whole packet. A failure requires all the IDocs in the IDoc packet to be sent to the endpoint again.



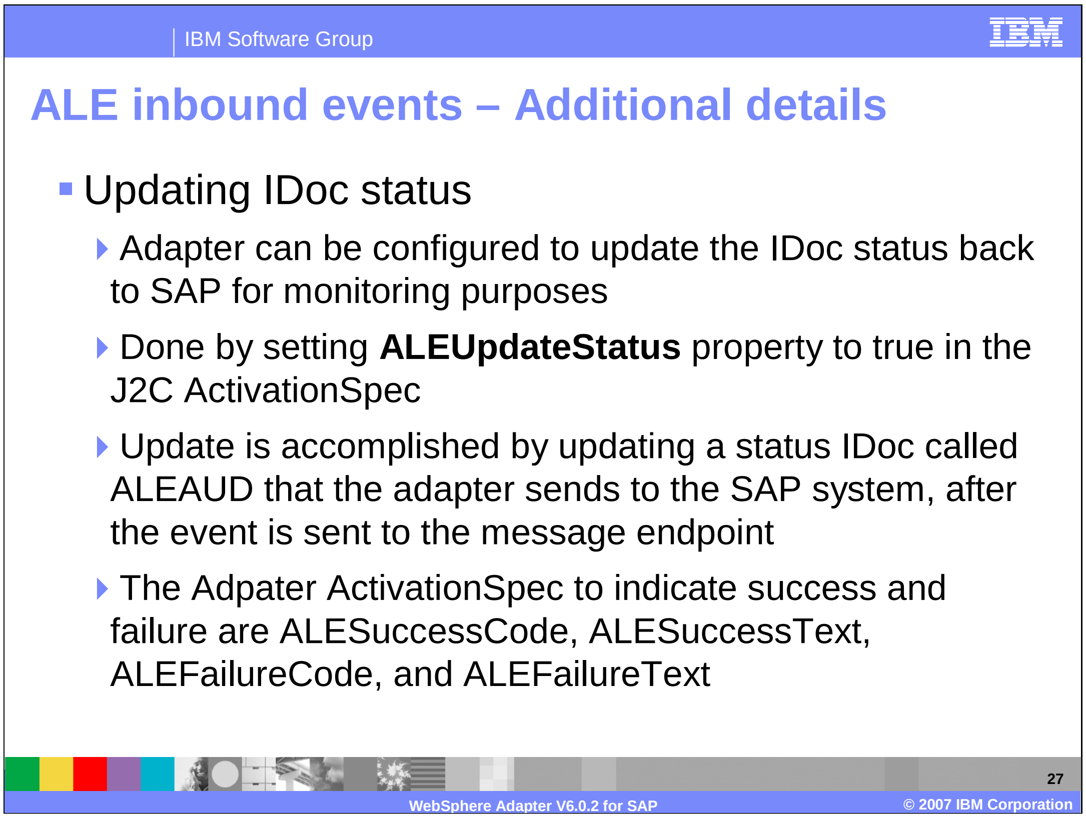
## Event processing for unparsed IDocs

- Adapter supports sending of ALE events without parsing the data and converting it into a Business Object, based on the Type property set at Enterprise Service Discovery.
  - Business Object contains the IDoc as byte array. Endpoint has to handle the parsing and converting it to a Business Object
  - If “*ParseIDocs*” is false, Adapter will not parse the IDoc

The adapter supports processing unparsed ALE events for Applications or components that have parsing capability.

A value of “UNPARSEDIDOC” is set for the metadata property “Type” at the business object level. The value for this property is set by the Enterprise Service Discovery when it determines the kind of Business Object structure to generate ( IDOC or UNPARSEDIDOC) .

If the end point requires the incoming IDoc not to be parsed, you can set the type of IDoc at the Enterprise Service Discovery process to UNPARSED and the Business Object contains the IDoc as byte array.



IBM Software Group

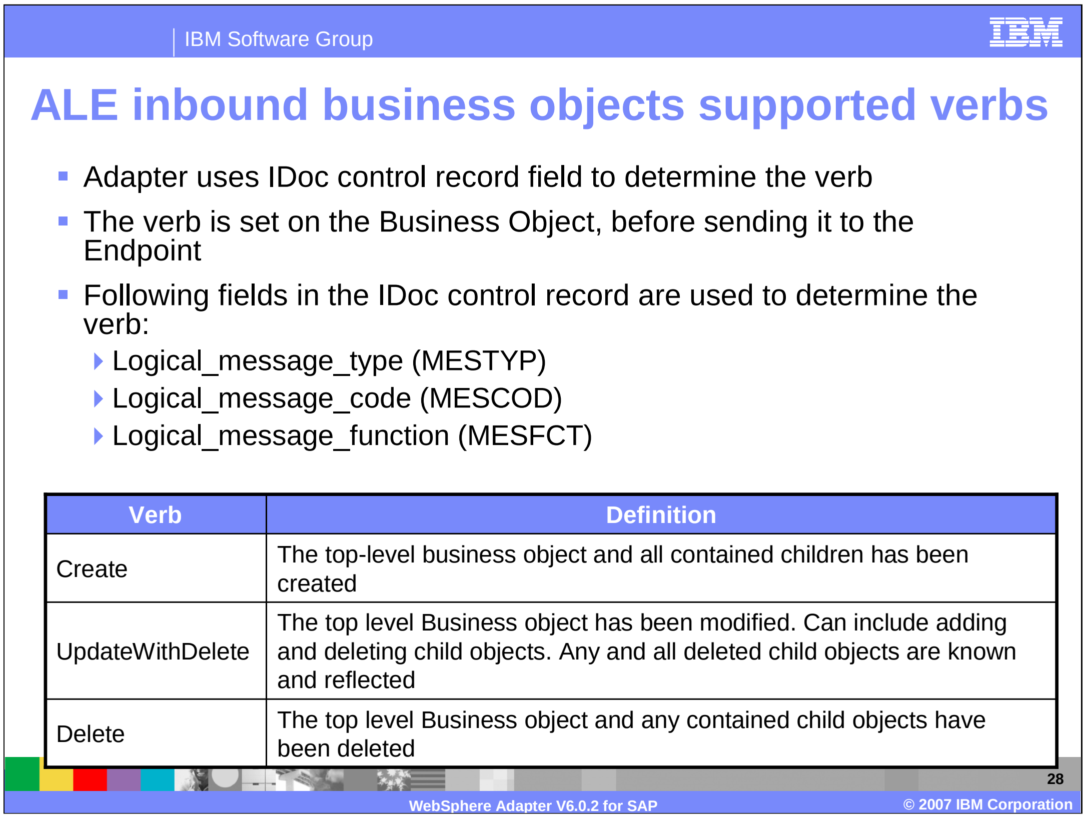
## ALE inbound events – Additional details

- 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 Adpater ActivationSpec to indicate success and failure are ALESuccessCode, ALESuccessText, ALEFailureCode, and ALEFailureText

27

WebSphere Adapter V6.0.2 for SAP

Here are some additional details for ALE Inbound events. 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 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.

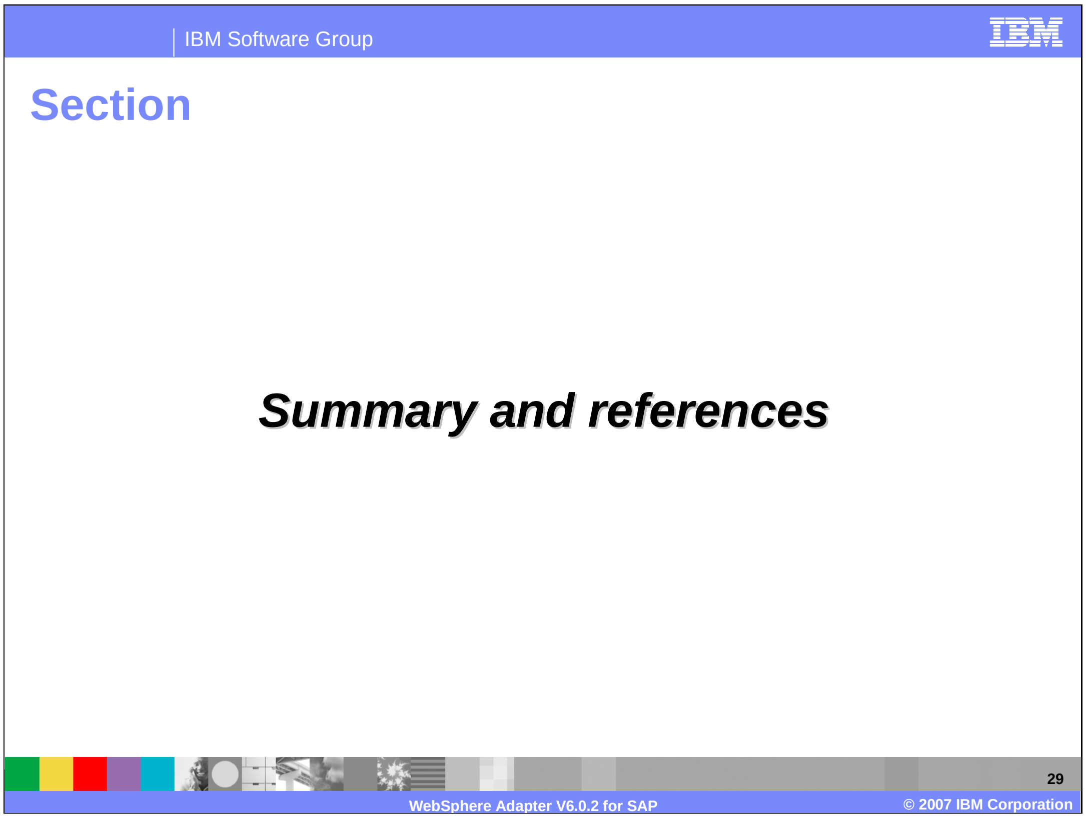


## ALE inbound business objects supported verbs

- 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 Business object has been modified. Can include adding and deleting child objects. Any and all deleted child objects are known and reflected |
| Delete | The top level Business object and any contained child objects have been deleted |

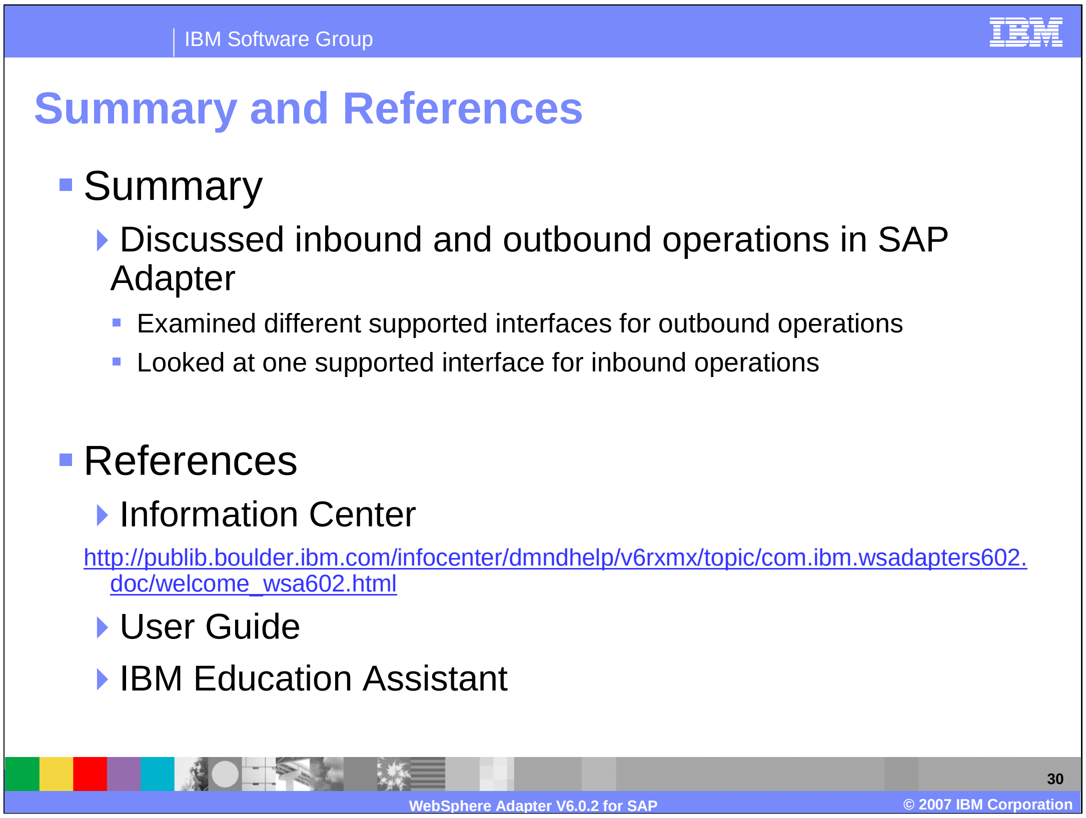
The supported verbs for ALE inbound operations are create, UpdateWithdelete and Delete. The adapter uses an IDoc control record field to determine the verb. The verb is then set on the Business Object, before sending it to the Endpoint



## Section

# ***Summary and references***

This section covers the summary and references.



IBM Software Group

## Summary and References

- Summary
  - Discussed inbound and outbound operations in SAP Adapter
    - Examined different supported interfaces for outbound operations
    - Looked at one supported interface for inbound operations
- References
  - Information Center

    http://publib.boulder.ibm.com/infocenter/dmndhelp/v6rxmx/topic/com.ibm.wsadapters602.doc/welcome_wsa602.html
  - User Guide
  - IBM Education Assistant

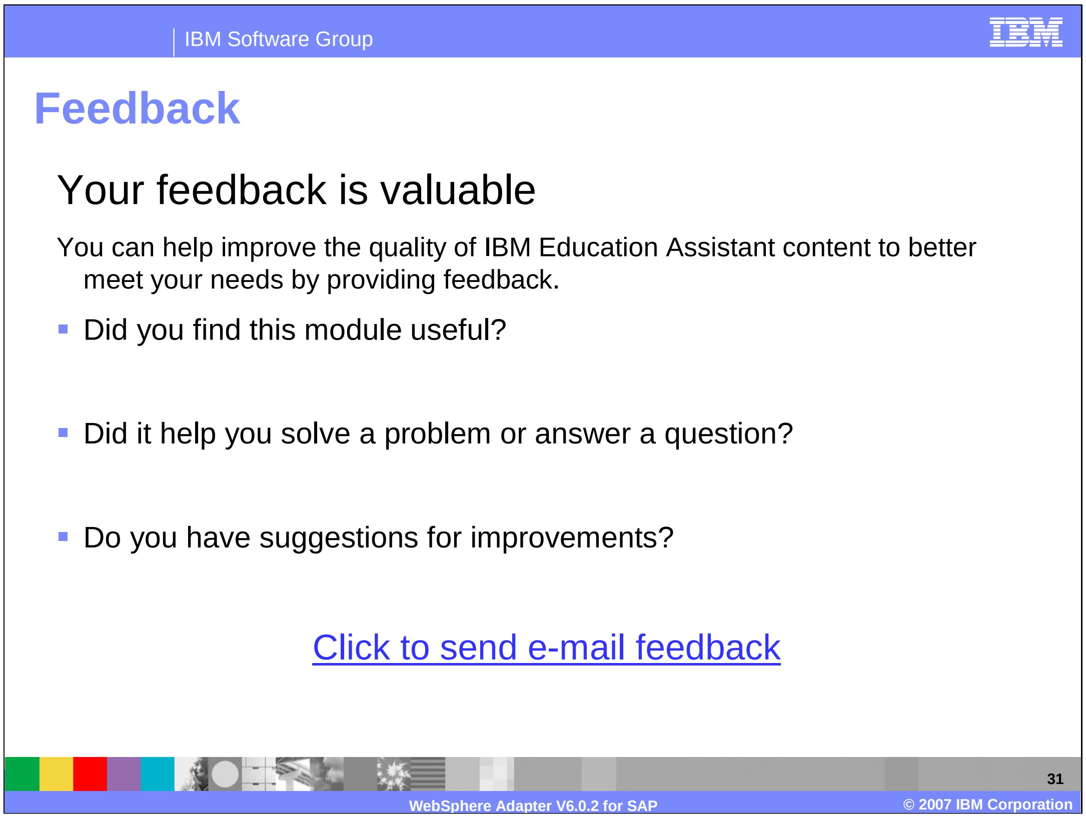
30

WebSphere Adapter V6.0.2 for SAP

© 2007 IBM Corporation

In summary, this presentation covered the details of inbound and outbound operations for the WebSphere Adapter for SAP.

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



# Feedback

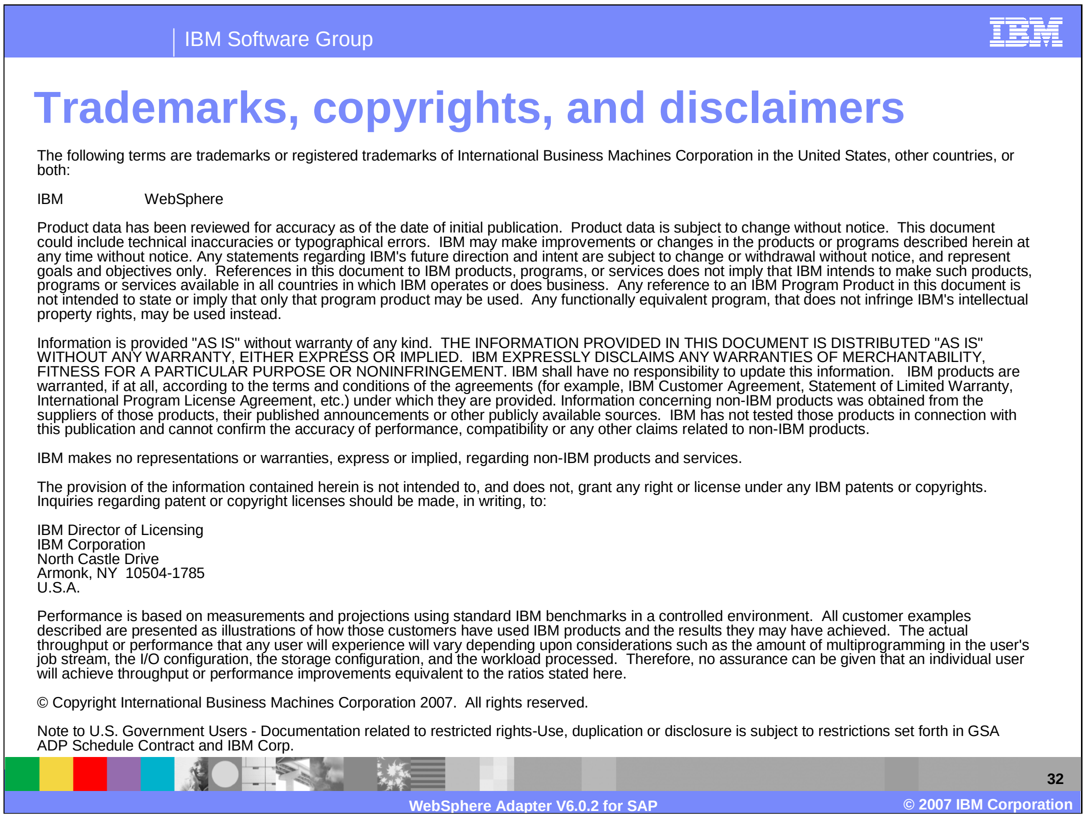
## Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send e-mail feedback

You can help improve the quality of IBM Education Assistant content by providing feedback.



# 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 WebSphere

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 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 (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.

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.