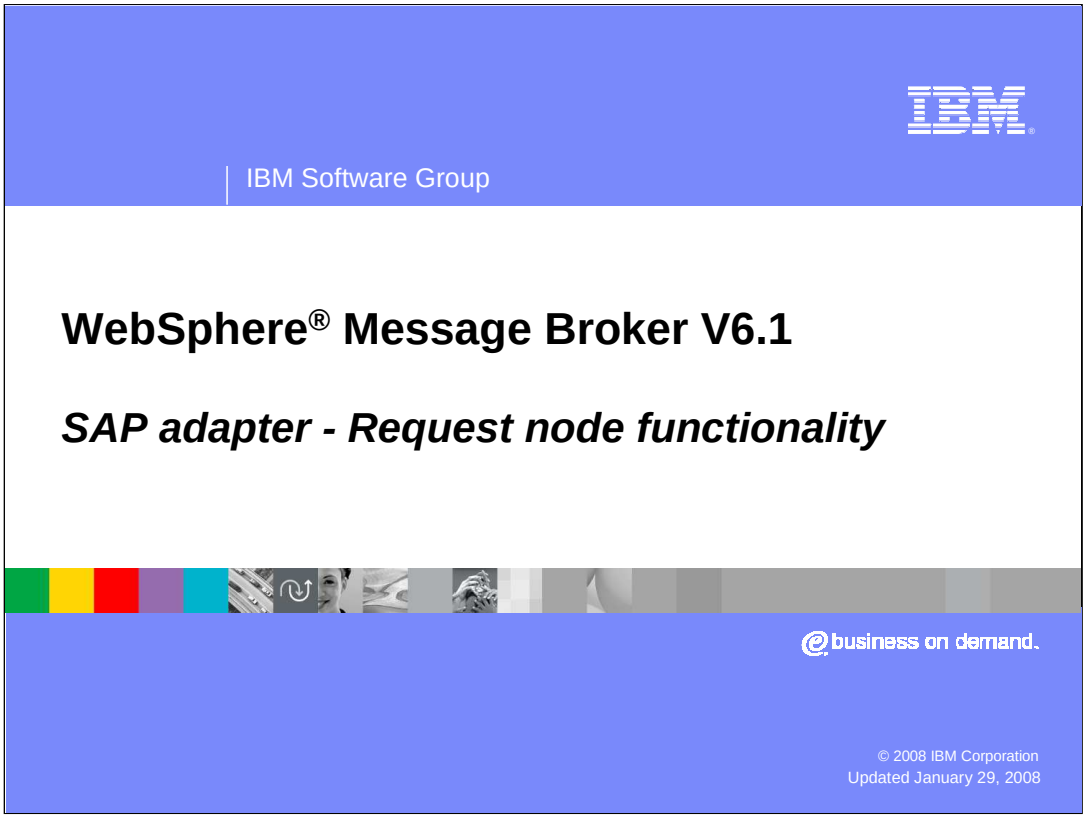


This presentation covers WebSphere Adapter for SAP outbound functionality.

Agenda

- Overview
- Outbound operations
- Summary

The agenda for this presentation is shown here.

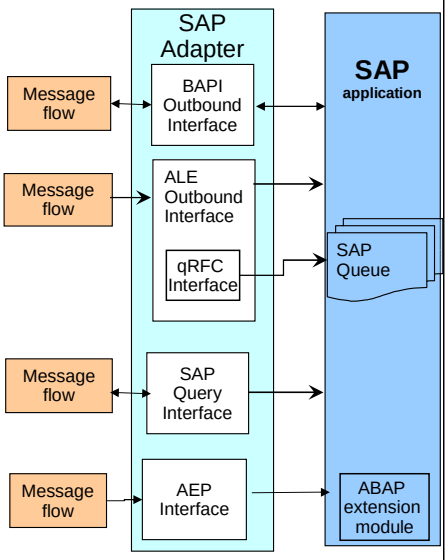
Section

Overview

This section provides an overview of WebSphere Adapter for SAP outbound support

SAP adapter outbound support

- Outbound calls supported using these interfaces:
- BAPI interface
  - Single BAPI call
  - BAPI / RFC
  - Support multiple BAPI calls in a single interaction (BAPI unit of work)
  - BAPI result set
- ALE interface
  - Single IDoc
  - IDoc packets (collection of IDocs)
  - qRFC support
- Advanced Event Processing (AEP) Interface
  - Custom IDocs and ABAP handlers
- Query Interface for SAP Software (QISS) interface
  - Retrieve Application table data



Outbound calls are supported with the BAPI, ALE, and AEP interfaces using Advanced Business Application Programming (ABAP) handlers and Query Interface for SAP Software (QISS). With BAPI, the calls can be simple BAPI calls, BAPI using remote function calls or multiple BAPI calls in a single interaction, referred to as BAPI Unit of work. BAPI outbound calls have request and response interaction style. The ALE interface supports passing single or multiple IDocs. These are one-way calls where the IDocs are passed to the SAP application. With the AEP interface, the adapter makes use of the ABAP handlers, and with the QISS interface you can directly query the SAP application tables

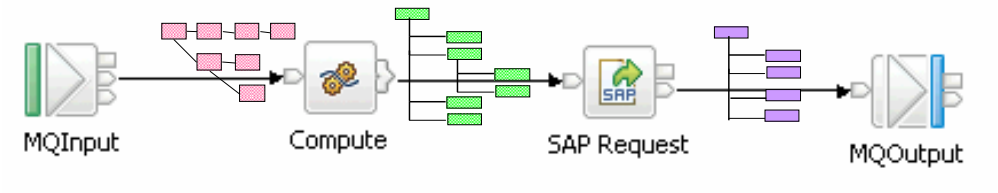
Section

SAP request node
High level functional overview

This section provides an overview of outbound operations supported using different interfaces by the SAP request node.

Outbound – High level flow

- SAP Request node receives a message representing the SAP function call
  - Message definition contains properties that correspond to the method arguments, interface used and other application specific information
- Adapter component extracts the elements from the message
  - Adapter converts the message 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 different supported interfaces is described here. The SAP function call and its parameters are modeled as a message. The message definitions are generated by the Adapter connection wizard within the WebSphere Message Broker Toolkit. The Adapter extracts the message content, metadata and application specific information from the incoming message and then determines the type of call, (BAPI, ALE, QISS or AEP), the SAP function name, and the function attributes. Using the SAPJCo API calls, the adapter makes the SAP function call to the target SAP application. The result (if any) returned by the function call is then sent to the rest of the message flow as a message.

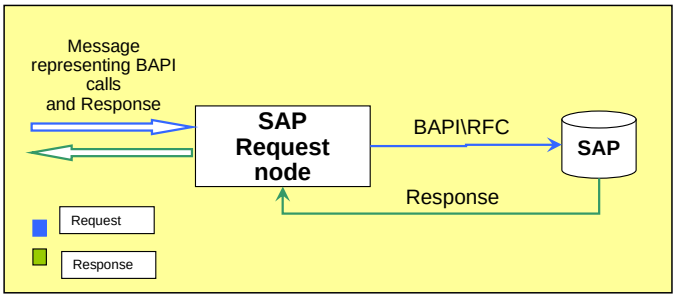
Section

SAP Request node
Operations using BAPI interface

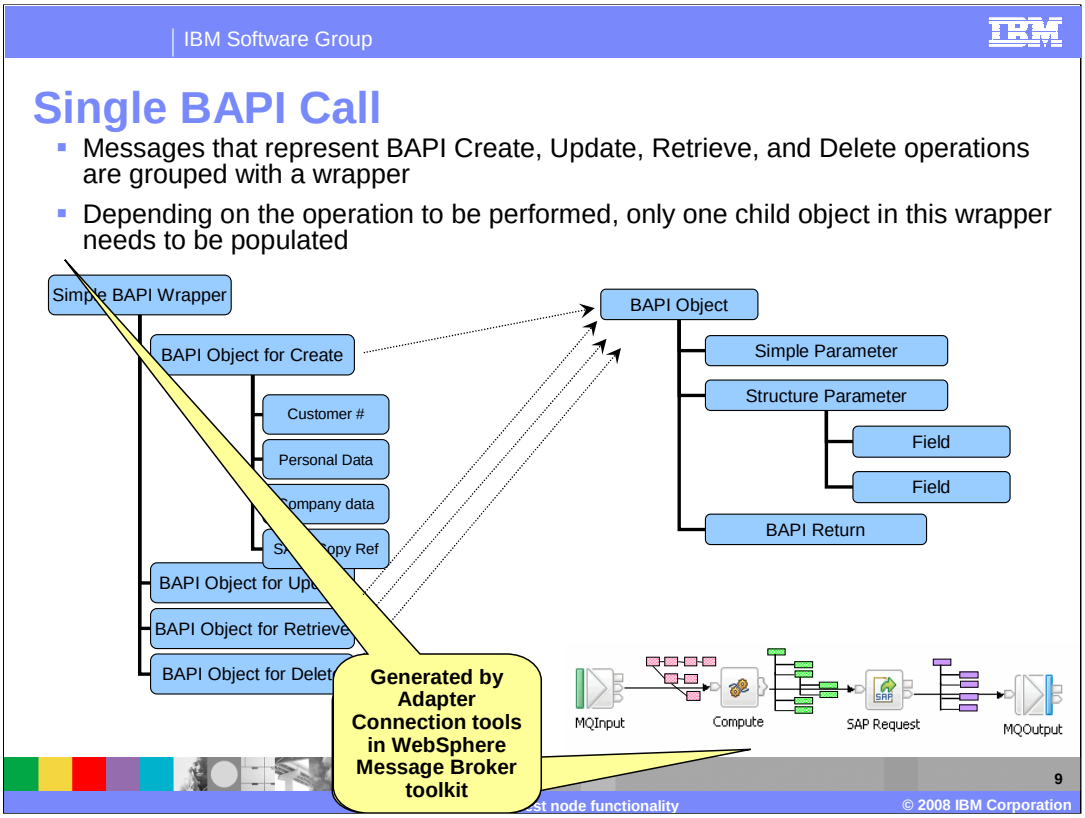
This section covers outbound operations using the BAPI interface.

Outbound calls using BAPI interface

- Adapter supports these BAPI calls:
  - Single BAPI calls
  - BAPI transactions, also called Logical units of work containing multiple BAPI calls
  - BAPI Resultset



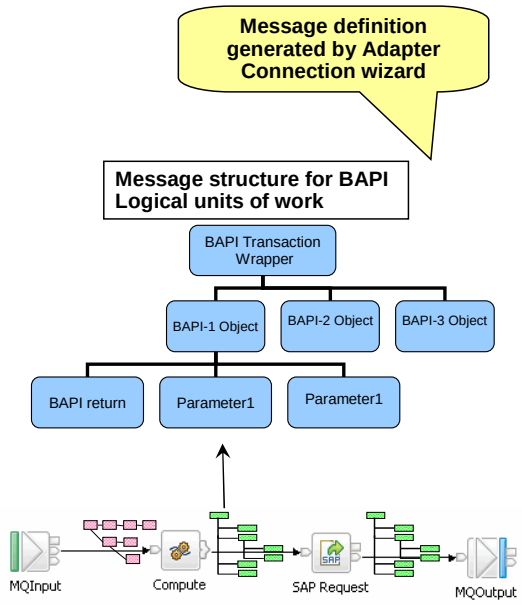
BAPIs are SAP standardized Business Application Programming Interfaces that enable vendor systems to interact with SAP systems. BAPI is implemented as an RFC-enabled function, so the adapter‘s BAPI interface can support any RFC-enabled function. The BAPI interface APIs allow vendor systems to interact with SAP. Adapters provide local transaction support for the BAPI interface using the BAPI calls, BAPI_TRANSACTION_COMMIT and BAPI_TRANSACTION_ROLLBACK. Create, Update, Delete, and Retrieve are the supported operations, except in the case of BAPI Resultset where “retrieveall” is the only supported operation.



A single BAPI call is a synchronous blocking call. SAP Request node supports single BAPI calls by representing each call with a single message. The adapter looks at the metadata and properties in the incoming message to make the appropriate SAP API function call through the sapjco.jar file. The SAP application executes the function call and returns response data to the adapter, which converts the response into a message, which is passed on to the rest of message flow. This diagram shows the message structure for a single BAPI call. These definitions are generated by the adapter connection wizard with the WebSphere Message Broker toolkit. The BAPI calls that represent the operations, namely, create, update, retrieve and delete are grouped and wrapped within a wrapper. The leaf nodes in the tree representing the operation have a structure containing the attributes of the function call, as shown on the right side of the page.

BAPI logical units of work

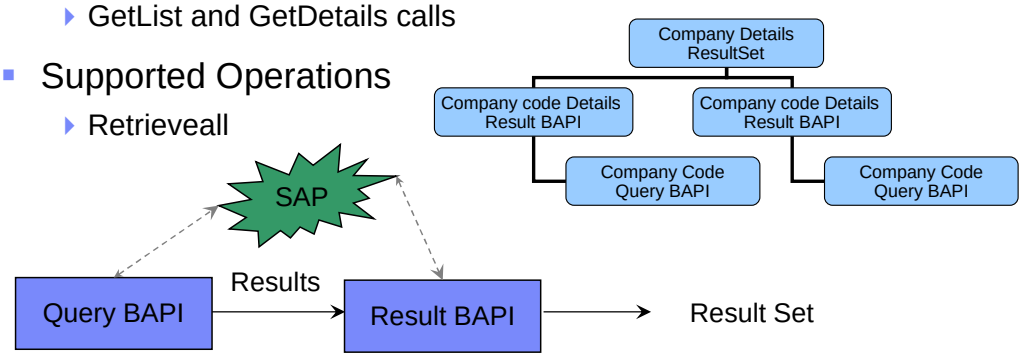
- Set of BAPIs that are processed in sequence to complete the entire transaction
- BAPI transaction wrapper represents the complete transaction (sequence of BAPI calls)
- The transaction is supported by a top-level object that consists of multiple child BAPIs, each one representing a simple BAPI in the sequence



Shown here are the BAPI logical units of work or transactions, where there can be one or more BAPI calls. Each BAPI call is represented by its child, as shown in the diagram as BAPI-1, BAPI-2, and BAPI-3. The BAPI transaction wrapper represents the complete transaction, with each second-level child representing a structure parameter or table parameter of the method. Simple attributes correspond to simple parameters of the method and the adapter uses the operation application specific information to determine the sequence of the BAPI calls.

BAPI result set interface

- Provides functionality to combine two BAPI calls
  - For example queryMethod BAPI can get a list of customer IDs which is passed on to resultMethod BAPI to retrieve detailed information
  - GetList and GetDetails calls
- Supported Operations
  - Retrieveall



This feature provides a new approach to building result set data by combining two BAPI calls. Out of two methods , one method acts as a query, which fetches all the keys that are based on data passed to this method, while the second method gets details based on the key. The query method and result method are combined to form a result set. The Result set query “queryMethod” is represented as a child of the “resultMethod” message. The “resultMethod” property level application specification information provides a foreign-key relationship with the “queryMethod” properties. This relationship is used to map key data from the query method to the result method using the foreign key. One by one all the key data are set into the resultMethod message.

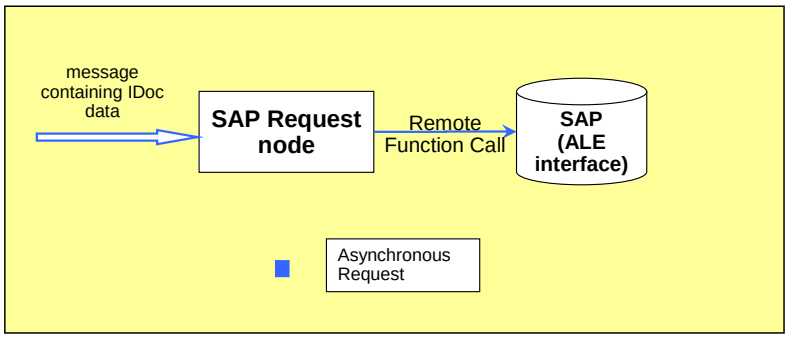
Section

SAP Request node
Outbound operations using ALE interface

This section covers outbound operations using the ALE interface.

Outbound operations using ALE interface

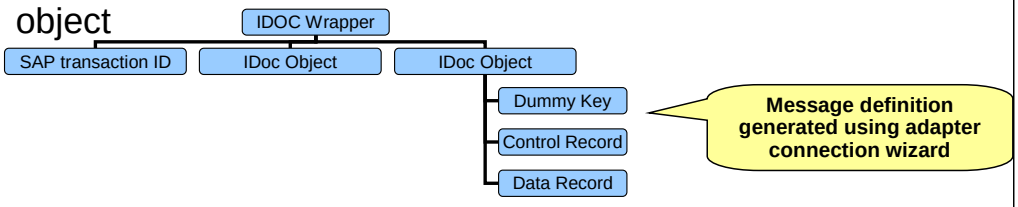
- Provides asynchronous interface for sending batch data to SAP applications
- ALE interface uses IDoc (Intermediate Document) data structures for message exchange
- Supported operations – Execute
  - Create, Update, Delete operations do not make sense for ALE
  - Operations called on SAP application based on the data contained in the control record child object of the IDoc



The ALE interface provides an asynchronous interface for sending batch data represented by IDocs to SAP applications. The adapter extracts the IDoc information from the message and then uses SAP JCo function calls to convert the message representing IDoc object to a table format that is compatible with the IDoc format. The adapter then uses Remote Function Calls (RFC) in the SAP RFC library to establish an RFC connection to the ALE interface and pass the IDoc data to the SAP system. After passing the data to SAP, the adapter releases the connection to SAP. 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. Only return code (success or failure) is returned and no return data is sent back to the adapter. For the ALE outbound call, all the information is in the IDoc and the only supported operation is “execute” on the IDoc. Create, Update, and Delete operations do not make sense for ALE as this is a batch oriented asynchronous interface. The receiving SAP application processes the data and performs appropriate operations based on the data contained in the control record child object of the IDoc

Outbound IDoc support

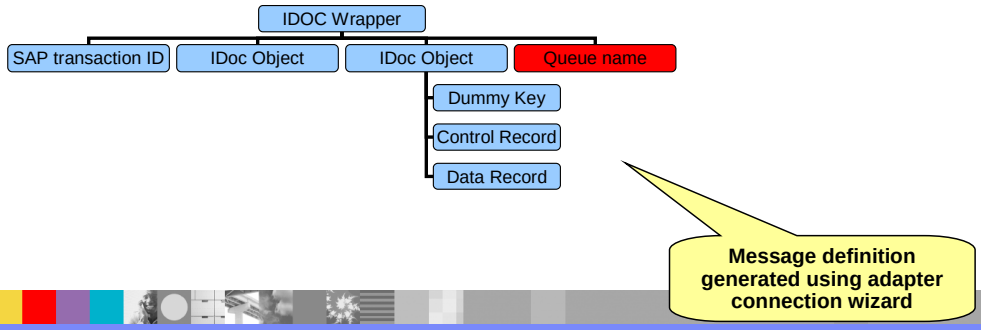
- Control record child object contains the metadata required by the resource adapter to process the message
- Data record child object contains
  - data to be processed by the SAP application and
  - the metadata required for the adapter to convert data to an IDoc structure for the Remote Function Call (RFC)
- Adapter provides support for IDoc packets using a wrapper object



This diagram shows the support for IDoc packets, where each IDoc is represented by the IDoc object. The message tree has a wrapper object that contains instances of IDoc objects. For individual IDocs, the wrapper contains only one instance of an IDoc object, whereas for IDoc packets, the wrapper contains many instances of an IDoc object. The IDoc object contains the Control Record and Data Record objects and the Control Record object contains the metadata required by the SAP adapter to process the IDoc object.

Outbound IDocs – qRFC support

- Option available in adapter connection wizard when using ALE interface for outbound interaction
- Enables batch transfer of IDocs to SAP logical queues
- Queue name can be set at design time or also during adapter connection creation in qRFCQueue name



A qRFC logical unit of work is represented by a logical message tree. The Adapter processes the outbound qRFC message with queue Remote Function Call. For processing qRFC, the call queue-name and queue-counter are required. This information is available in the RFC wrapper level property “qRFCQueueName” . Each qRFC call is processed in sequence by the adapter. Once the logical units of work reach the SAP Queue, the SAP application processes the logical units of work in the order they were delivered. The qRFC wrapper level property “qRFCQueueName” contains a default queue name, which is gathered at design time by the adapter connection wizard. If you do not provide a queue name at runtime in the “qRFCQueueName” property, the default value is used as the queue name. You can override the default value by setting the queue name in the message definition. The adapter finds the list of queues available at runtime and SAP uses them to determine if the queue name is valid.

Section

SAP request node
Outbound operations using AEP interface

This section covers the outbound operations using the Advanced event processing interface.

AEP interface - Outbound

- Advanced Event Processing (AEP)
  - Process data using custom ABAP handlers.
  - You need to develop the operation specific ABAP handler for each object that needs to be supported
  - Message data is converted into ABAP handler function and is invoked on the SAP system.
  - Data returned by ABAP handler function mapped to business object and is returned as response.
  - Supported operations Create, Update, Delete, Retrieve

Business objects for the AEP Interface are defined in SAP as an IDoc and either standard IDocs delivered with SAP or custom-built IDocs can be used. The adapter connection wizard is used to generate the message definition based on an IDoc. Business object development for the ABAP Extension Module consists of creating an application-specific business object definition and an associated ABAP handler for each operation that you want to support. ABAP handlers reside in the SAP application as ABAP function modules and communicate with the connector and are needed to get business object data into or out of the SAP application database. ABAP handlers are responsible for adding business object data into the SAP application database (Create, Update, Delete operations) or for retrieving data from the SAP application database (Retrieve operation). You must develop operation-specific ABAP handlers for each supported hierarchical business object. The application-specific business object and the ABAP handler rely on each other's consistency to pass data into and out of the SAP application. Therefore, if you change the business object definition, you must also change the ABAP handler. In the outbound processing, the adapter receives the AEP record as a message. The message contains the business data along with the metadata. Depending on the operation set, the adapter looks up the ABAP handler to be invoked from the operation Application Specific Information. The adapter converts the business data to handler data format and passes it along with the ABAP Handler information to the SAP adapter infrastructure component in the SAP application. After the object-specific ABAP handler finishes processing the business object data, it returns the business object data in IDoc format to the ABAP Component, which converts the business object data back to its original format and returns it to the adapter. The adapter component in the SAP Application invokes the particular ABAP handler and returns the results back to the adapter, which converts the results to a message and returns it to the caller.

AEP Interface – Developing ABAP handlers

- IBM WebSphere Business Integration Station tool
  - Call Transaction Recorder Wizard
    - Can be accessed by running transaction /CWLD/HOME_AEP
    - Generates code stubs for each screen modified during recording phase
    - Need to adopt this code in ABAP handler
    - Does not generate the IDoc business object.
- Adapter Connection Wizard provides ability to generate the required message definitions and other artifacts for IDocs (either custom or standard).
- Adapter provides samples that can be referenced for understanding the AEP interface implementation.

WebSphere Message Broker toolkit provides a set of transports that can be used to set up the IBM WebSphere Business Integration Station tool, which provides a wizard called the Call Transaction Recorder to capture the changes you make and code stubs that can be used to generate ABAP handlers. The wizard does not create the handler, just code stubs that can be used to develop handlers. The tool does not create the IDocs for you, so you must also create the IDocs on the SAP backend. The WebSphere Message Broker toolkit provides the adapter connection wizard, which can generate the message definitions for the custom IDocs you created on the SAP backend.

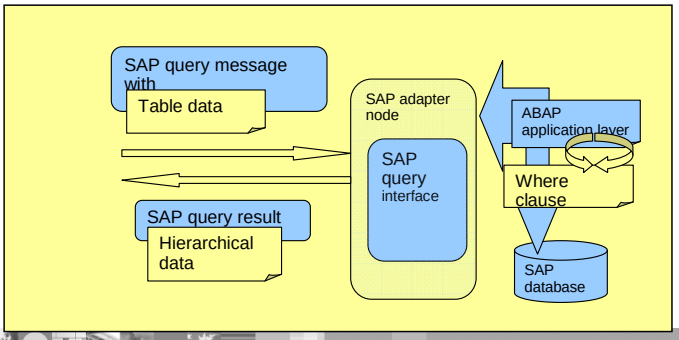
Section

SAP outbound operations using QISS

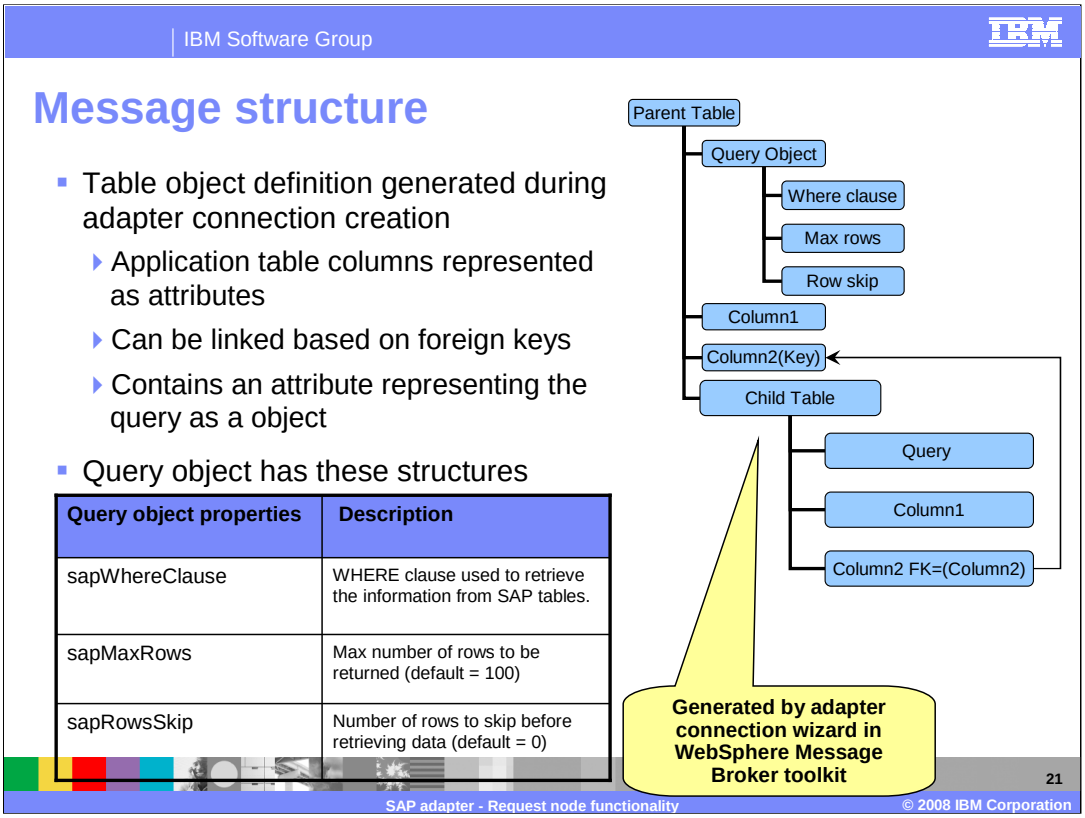
This section covers outbound operations supported by Query Interface for SAP software (QISS)

Outbound operations supported by QISS

- Provides an interface to retrieve data directly from SAP application tables
- QISS uses special object called table object for message exchange.



To access data in hierarchical form, SAP provides Remote Function Calls (RFC) and BAPI. However, in some scenarios, you might want to retrieve data in some specific hierarchical form or from SAP application tables for which SAP does not provide an interface. Query interface for SAP Software supports the retrieval of data directly from the SAP Application tables that can be represented in custom hierarchical form. The SAP Request node receives the message with table information, and the adapter extracts the 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 result (rows) are converted to a table and inserted into a container.

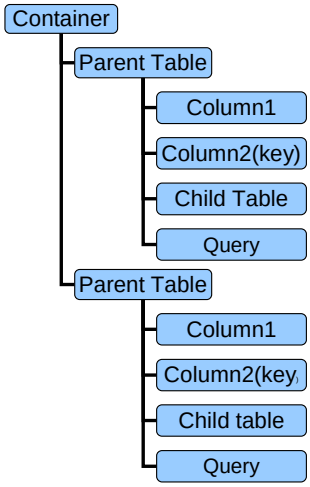


The diagram on the top right shows the structure of the table object. The columns in the application table on which the query is to be run map as attributes of the table object. Another important attribute of the table object is the query object. Using the adapter connection wizard, you can generate a query object for an SAP application table that you want to retrieve data from. Table objects can be linked using foreign keys defined inside the table object's property to define a hierarchy (parent-child relationships). The child table object has a foreign key that references a property in the parent query object. Query object has a WHERE clause, maxrows and rowskip as attributes. The WHERE clause entered using the adapter connection wizard becomes the table's default WHERE clause, but you can override the default WHERE clause at runtime. The syntax of the WHERE clause is not validated before the query is run. If running the WHERE fails, the appropriate exception is created. The maxRows attribute defines the maximum number of rows to retrieve, while the rowsSkip attribute defines the number of rows to skip before retrieving the rows.

QISS interface - Supported operations

- Supported operations –
  - RetrieveAll
    - Each row that is retrieved is mapped to a table object
    - Each table object is then inserted into a container
  - Exists
    - Means to check for existence of records
    - Returns the same table object if success and an exception if no match

results returned from RetrieveAll



For the SAP Query Interface outbound call, supported operations are RetrieveAll and Exists. For RetrieveAll processing, each row that is retrieved is mapped to a Table Object and each table Object is then inserted into a container. For Exists processing, it checks for the existence of records and works just on the top level object. It then returns a message with a Boolean value.

Section

Summary and references

This section provides a summary and references.

Summary and references

- Summary
  - Discussed SAP request node's functionality and various interfaces supported in detail
- References
  - Information center
  - User guide
  - IBM Education Assistant

This presentation covered the details of SAP Request nodes functionality and the various interfaces supported by the adapter for outbound interaction. More information can be found in the 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:

mailto:iea@us.ibm.com?subject=Feedback_about_WMBV61_SAPAdapter_RequestNode.ppt

This module is also available in PDF format at: ../WMBV61_SAPAdapter_RequestNode.pdf

You can help improve the quality of IBM Education Assistant content by providing feedback on this module using the link on this slide.

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.