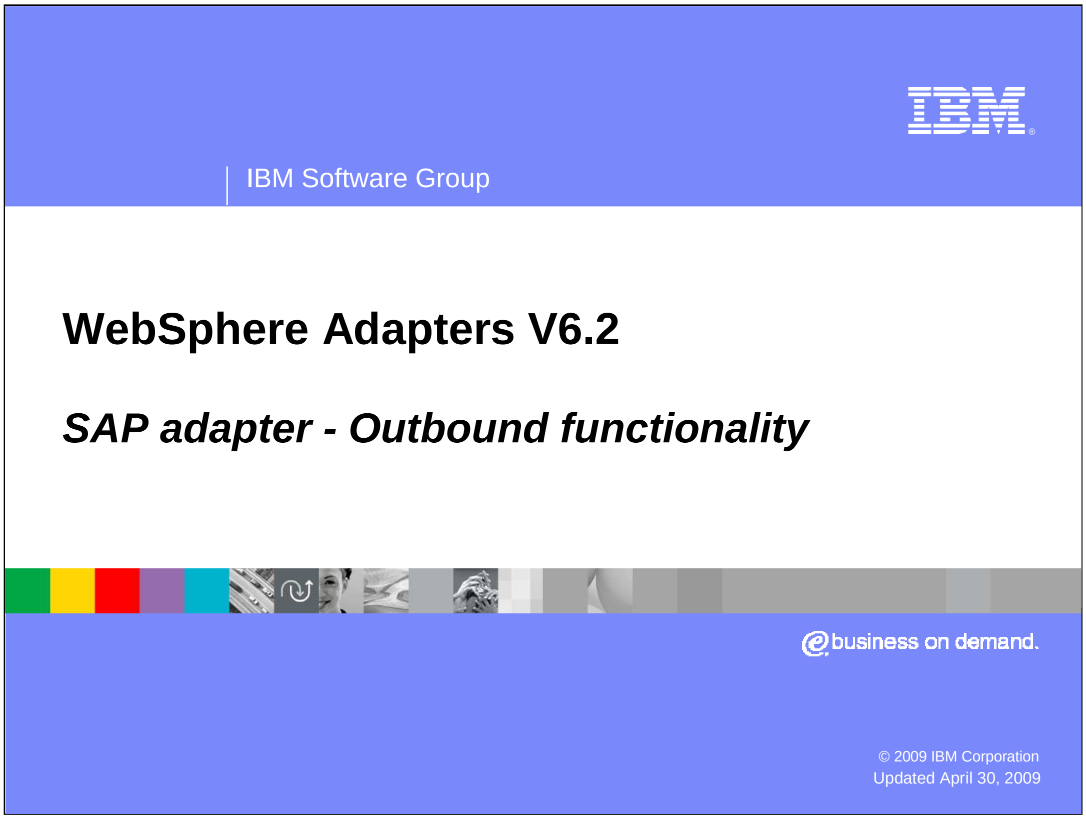


IBM Software Group

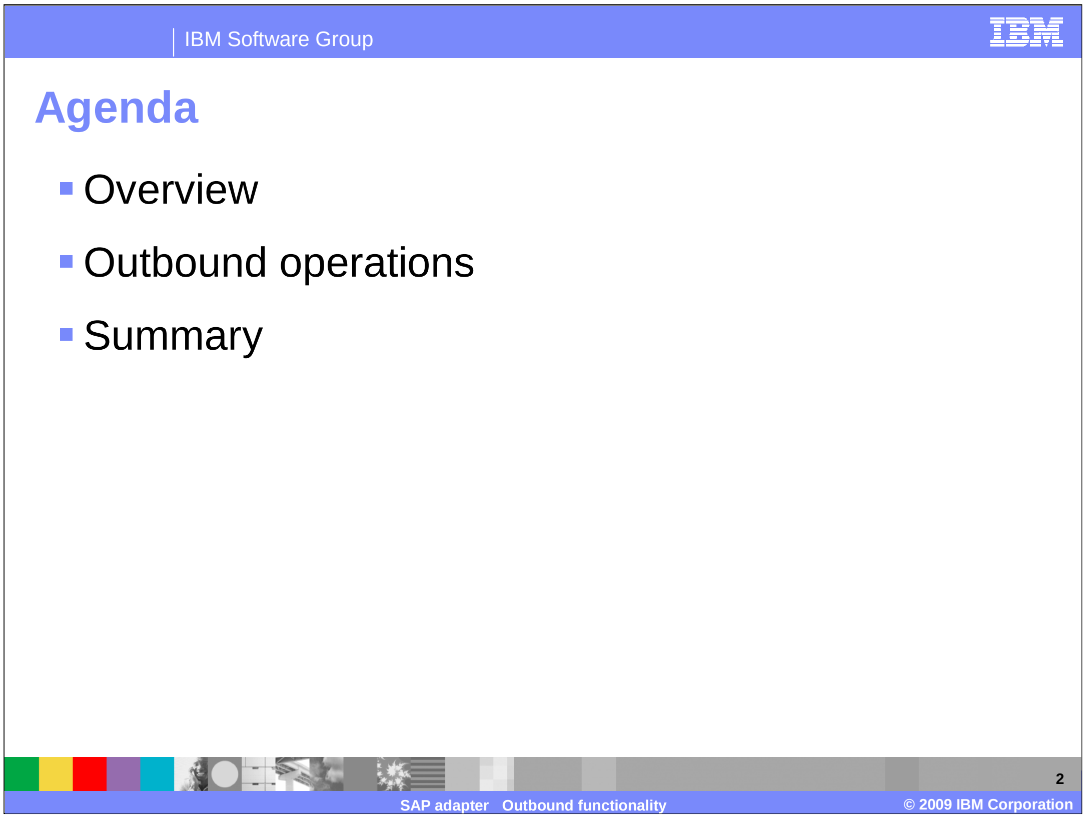
WebSphere Adapters V6.2

SAP adapter - Outbound functionality

business on demand.

Updated April 30, 2009

This presentation covers the outbound functionality of the WebSphere® Adapter for SAP.



Agenda

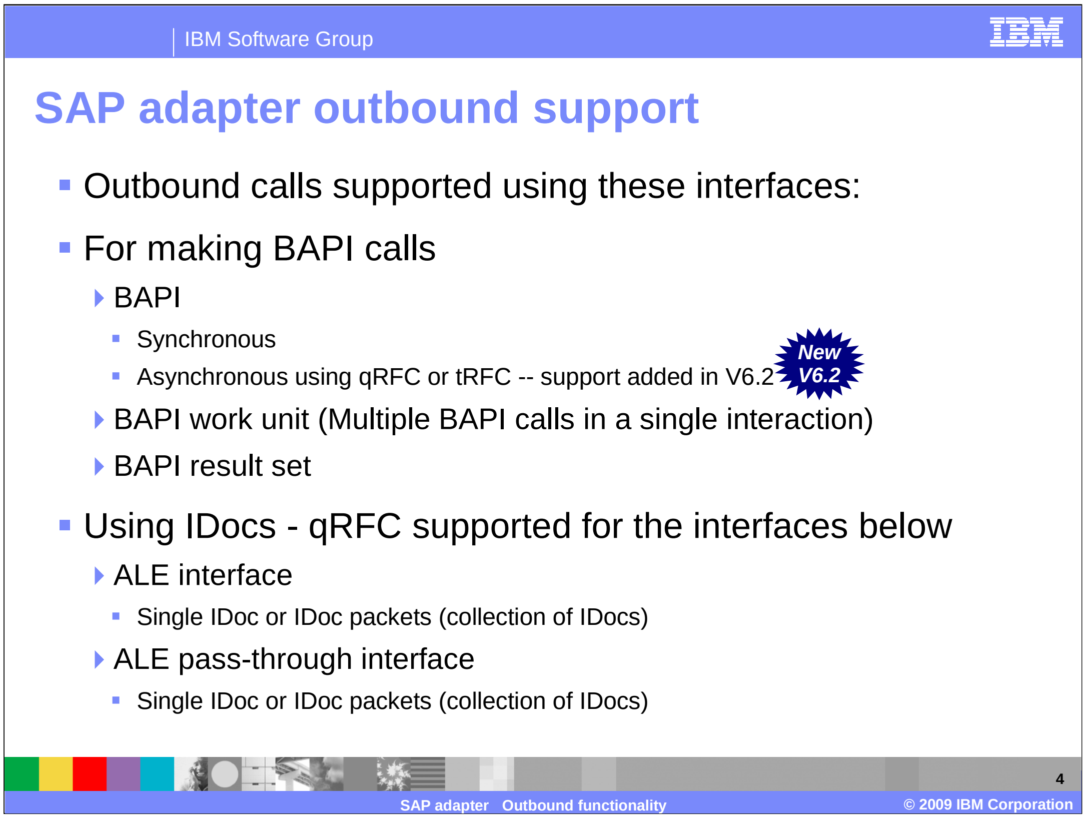
- Overview
- Outbound operations
- Summary

The agenda for this presentation is shown here.

Section

Overview

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



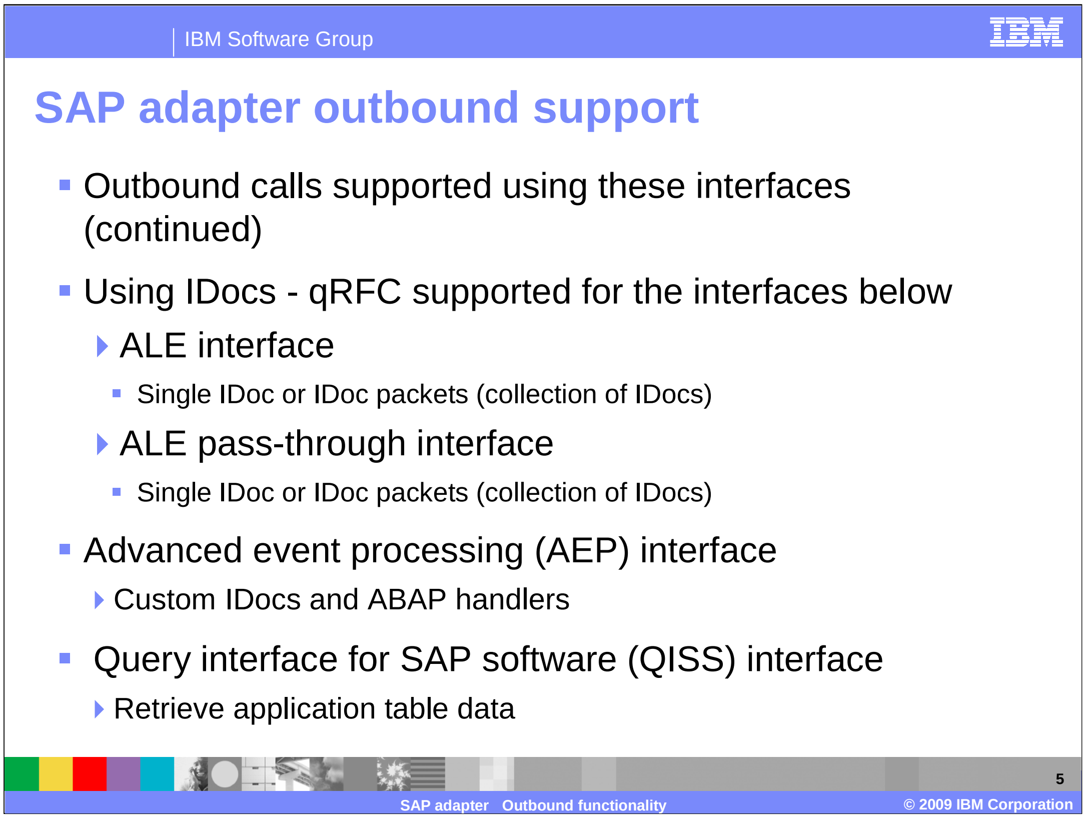
SAP adapter outbound support

- Outbound calls supported using these interfaces:
- For making BAPI calls
  - BAPI
    - Synchronous
    - Asynchronous using qRFC or tRFC -- support added in V6.2 New V6.2
  - BAPI work unit (Multiple BAPI calls in a single interaction)
  - BAPI result set
- Using IDocs - qRFC supported for the interfaces below
  - ALE interface
    - Single IDoc or IDoc packets (collection of IDocs)
  - ALE pass-through interface
    - Single IDoc or IDoc packets (collection of IDocs)

Outbound calls are supported with the BAPI interface, BAPI work unit, BAPI result set, ALE interface, or Advanced Event Processing (AEP) interface using ABAP handlers and Query Interface for SAP Software (QISS).

With BAPI, outbound can be simple BAPI calls, or BAPI using remote function calls, or multiple BAPI calls in a single interaction, referred to as a BAPI work unit. BAPI outbound calls have request and response interaction style. In V6.1 result set functionality has been added. Using the BAPI result set interface, you can configure the adapter to return combination of output from two different BAPI calls. In V6.2 the adapter has added support for transactional and queued RFC support for BAPI calls With the Advanced Event Processing interface, the adapter makes use of the Advanced Business Application Programming (ABAP) handlers.

With the QISS interface, you can directly query the SAP application tables.



IBM Software Group

SAP adapter outbound support

- Outbound calls supported using these interfaces (continued)
- Using IDocs - qRFC supported for the interfaces below
  - ALE interface
    - Single IDoc or IDoc packets (collection of IDocs)
  - ALE pass-through interface
    - Single IDoc or IDoc packets (collection of IDocs)
- Advanced event processing (AEP) interface
  - Custom IDocs and ABAP handlers
- Query interface for SAP software (QISS) interface
  - Retrieve application table data

5

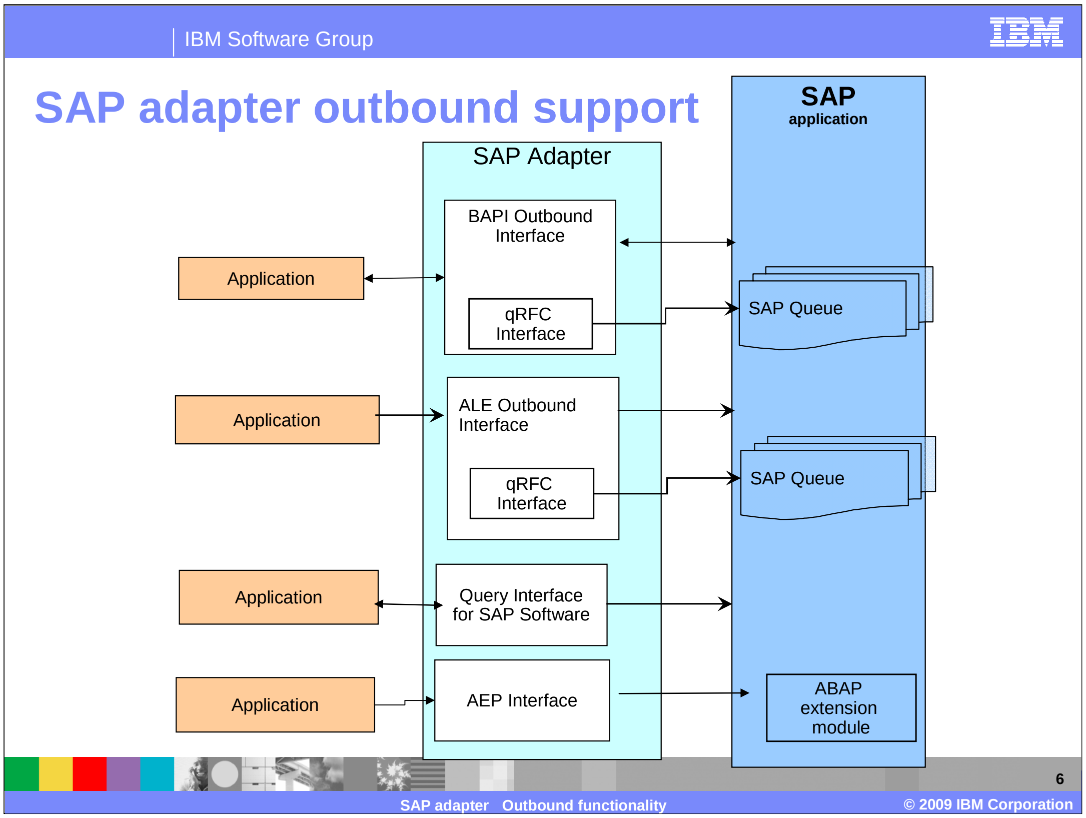
SAP adapter Outbound functionality

© 2009 IBM Corporation

The ALE interface supports passing single or multiple IDocs, which are one-way calls where the IDocs are passed to the SAP application. For ALE pass-through interface, the wrapper business object contains a data stream representing the IDoc. No separate IDoc business object exists for pass-through IDocs. The adapter uses the IDoc wrapper business object to populate the appropriate RFC-enabled function call used by the ALE interface.

With the advanced event processing interface, the adapter makes use of the advanced business application programming (ABAP) handlers.

With the QISS interface, you can directly query the SAP application tables and retrieve the data in the hierarchical form you define.



IBM Software Group

SAP adapter outbound support

SAP Adapter

BAPI Outbound Interface

qRFC Interface

ALE Outbound Interface

qRFC Interface

Query Interface for SAP Software

AEP Interface

Application

Application

Application

Application

SAP application

SAP Queue

SAP Queue

ABAP extension module

SAP adapter   Outbound functionality

6

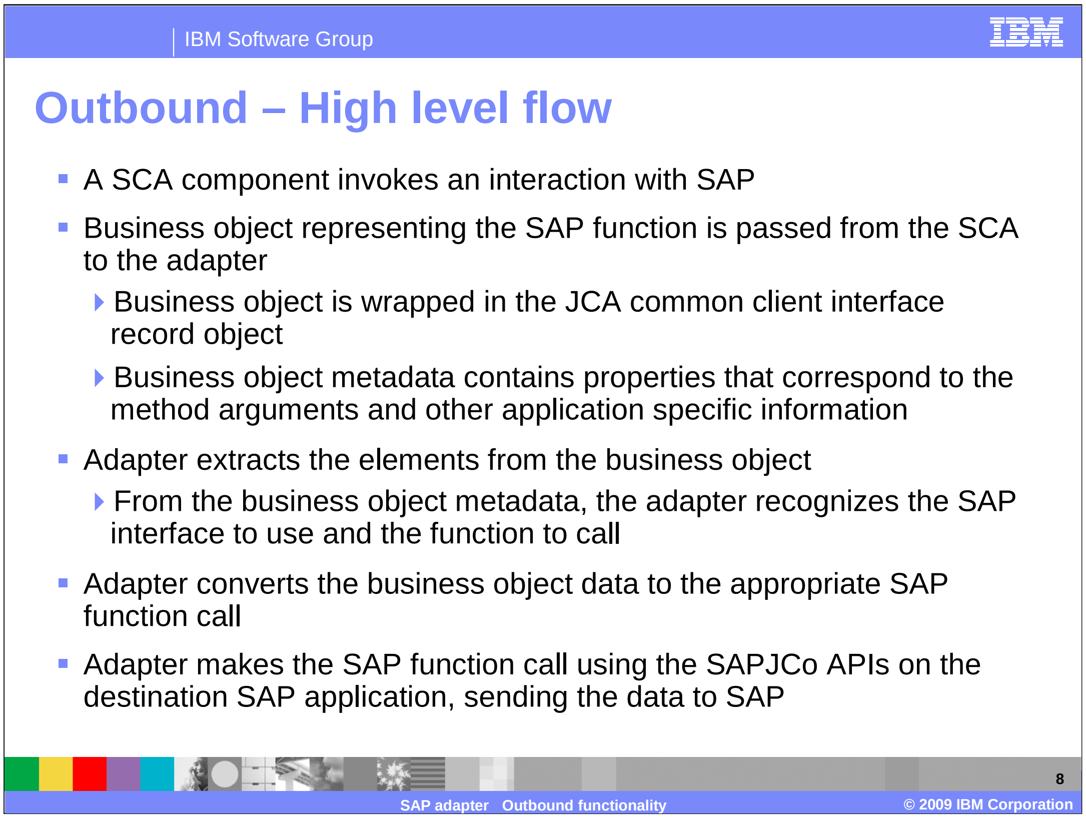
© 2009 IBM Corporation

This diagram shows the high level flow and the components in outbound calls. The various interfaces that the adapter supports for outbound are shown in the diagram above. You can use the adapter to make BAPI calls, exchange data using IDocs over ALE interface, interact directly with application tables using query interface for SAP software interface or can call ABAP handlers when using the advanced event processing interface.

Section

SAP adapter
high level functional overview

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

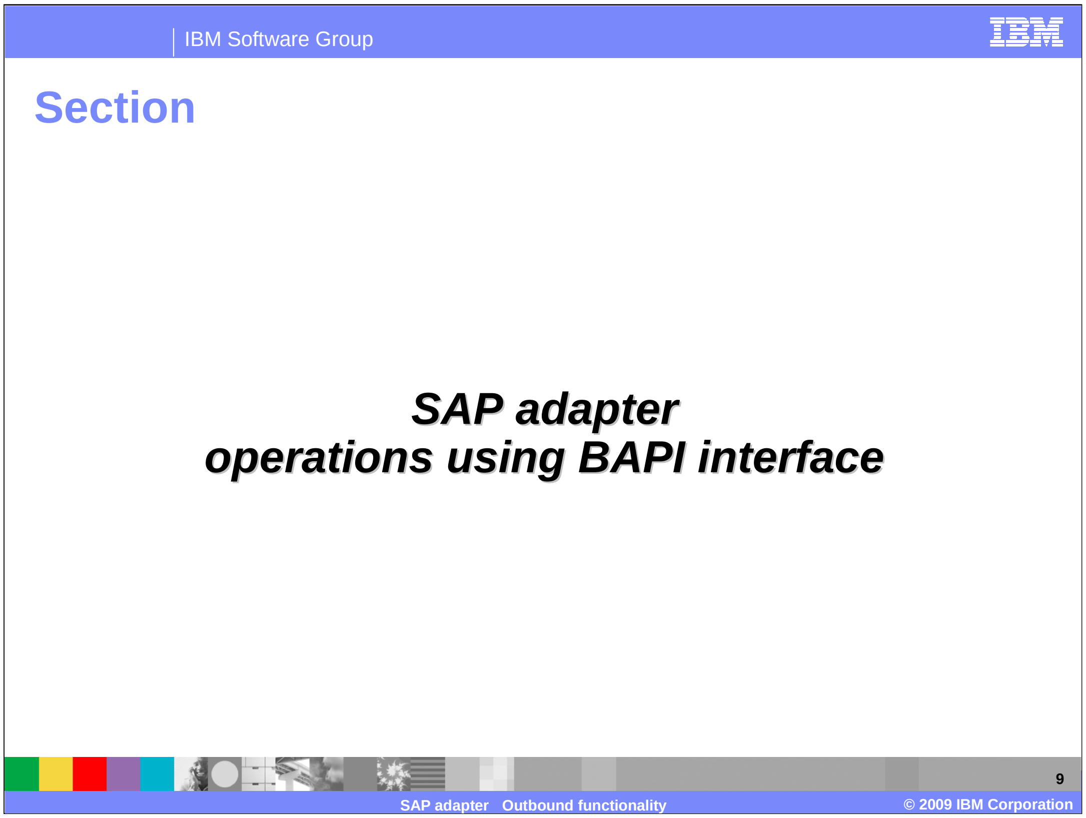


Outbound – High level flow

- A 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 common client interface 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 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 either the BAPI or ALE interface is described here.

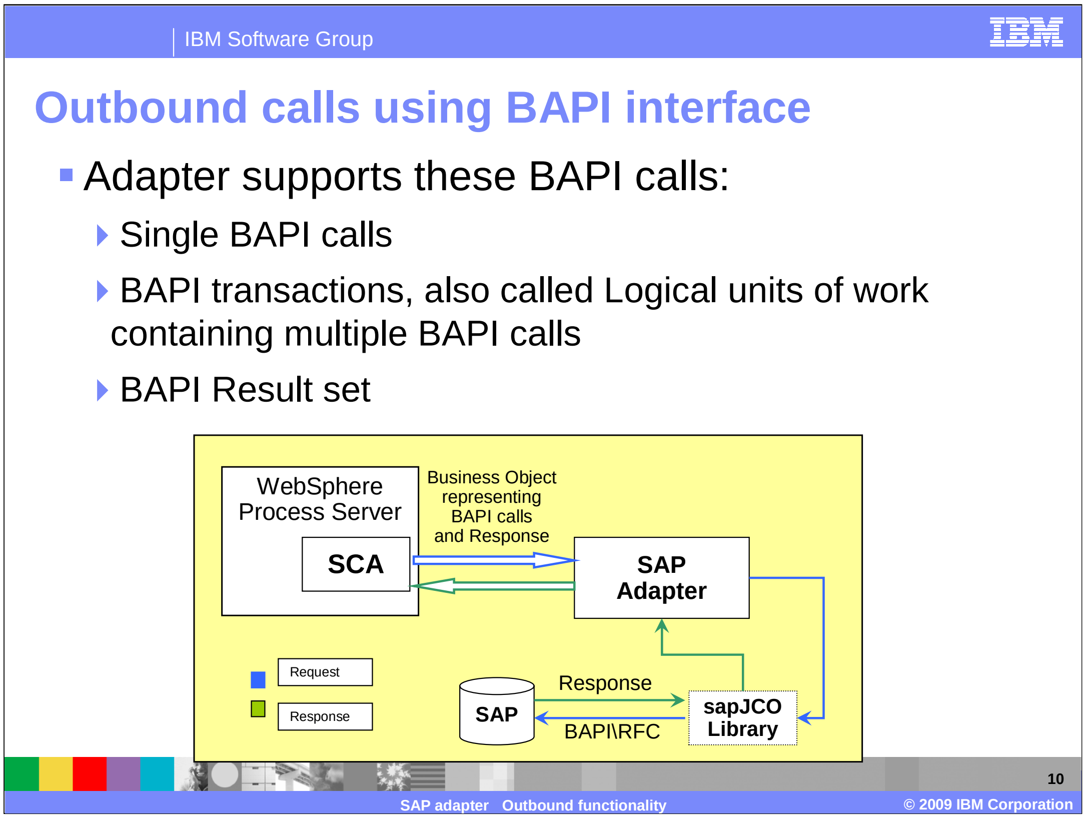
The SAP function call and its parameters and other attributes are modeled using a business object. The business objects were generated by the external service wizard within WebSphere Integration Developer. The SCA client wraps the business object in a JCA common client interface record object. The adapter extracts the business object and its metadata from the common client interface record object. It then determines the type of call, (BAPI, ALE, QISS or AEP), and 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 returned to the calling component.



Section

SAP adapter
operations using BAPI interface

This section covers outbound operations using the BAPI interface.



IBM Software Group

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 Result set

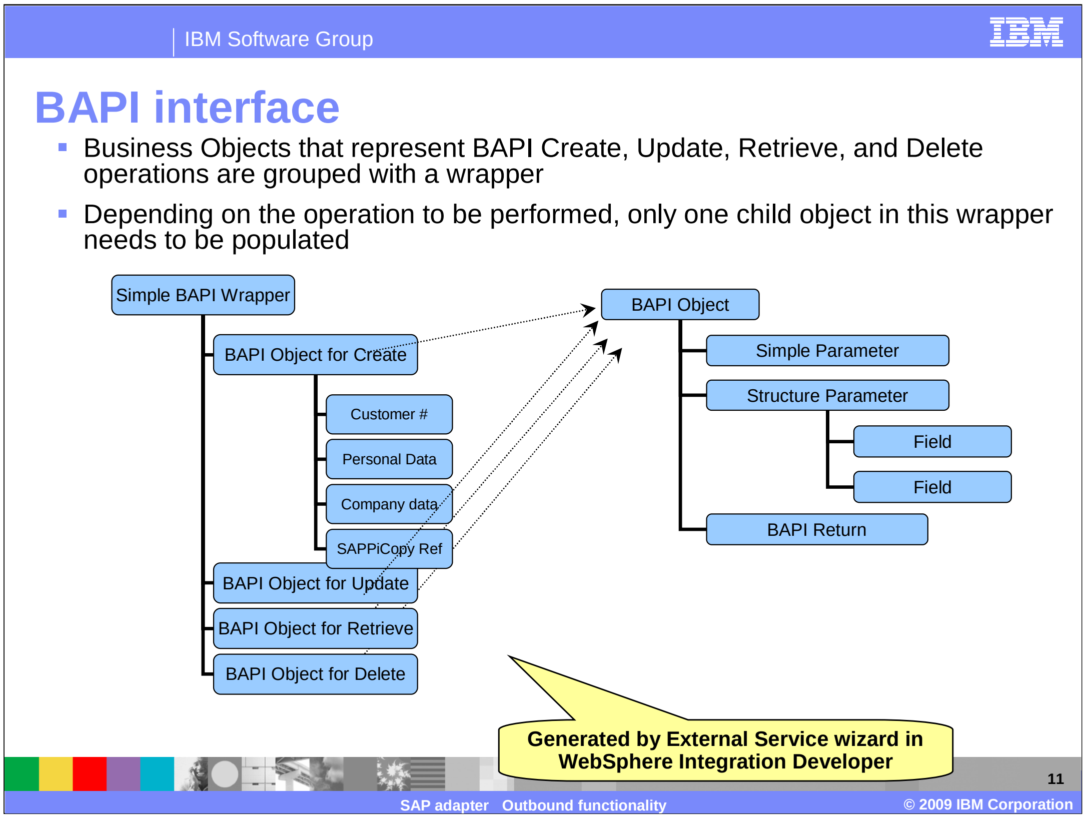
WebSphere Process Server
SCA
Business Object representing BAPI calls and Response
SAP Adapter
🟦 Request
🟩 Response
SAP
Response
BAPI\RFC
sapJCO Library

10

SAP adapter Outbound functionality

© 2009 IBM Corporation

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 result set where “retrieveall” is the only supported operation.



IBM Software Group

BAPI interface

- Business Objects that represent BAPI Create, Update, Retrieve, and Delete operations are grouped with a wrapper
- Depending on the operation to be performed, only one child object in this wrapper needs to be populated

Simple BAPI Wrapper
BAPI Object for Create
Customer #
Personal Data
Company data
SAPPiCopy Ref
BAPI Object for Update
BAPI Object for Retrieve
BAPI Object for Delete

BAPI Object
Simple Parameter
Structure Parameter
Field
Field
BAPI Return

Generated by External Service wizard in WebSphere Integration Developer

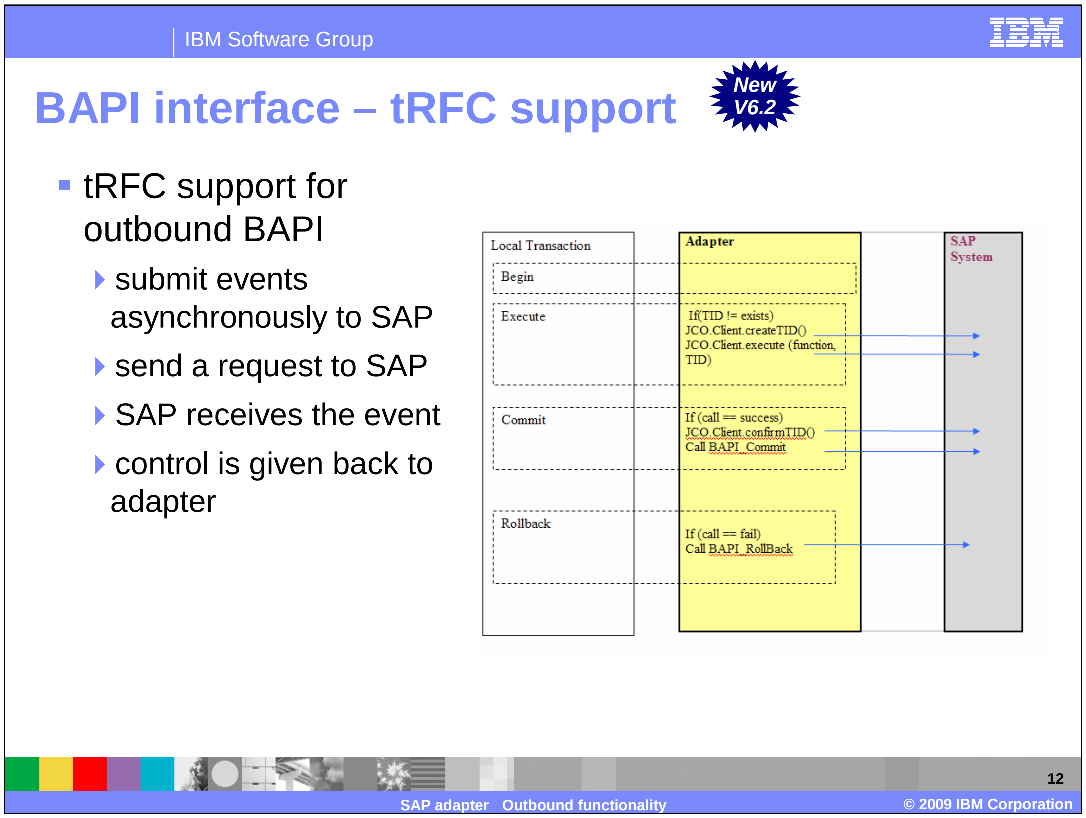
11

SAP adapter Outbound functionality

SAP adapter supports single BAPI calls by representing each call with a single business object. The adapter looks at the metadata and properties that are part of business object in the incoming request 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 business object, which is passed on to the calling component. This diagram shows the business object structure for a single BAPI call. These business object definitions are generated by the external wizard with the WebSphere Integration Developer. The BAPI calls that represent the operations, namely, create, update, retrieve and delete are grouped and wrapped within a wrapper. 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.

If you select Synchronous RFC (the default) during configuration for a simple BAPI, these processing steps occur. The adapter receives a request from a client application in the form of a BAPI business object. The adapter converts the BAPI business object to an SAP JCo function call. The adapter uses the Remote Function Call (RFC) interface to process the BAPI or RFC function call in the SAP application. After passing the data to the SAP server, the adapter handles the response from SAP and converts it back into the business object format required by the client application. The adapter then sends the response back to the client application.

With V6.2, the adapter supports calling BAPI’s asynchronously. For simple BAPIs, you can request that processing be handled synchronously or asynchronously. For asynchronous calls, the client application does not wait for a response from the SAP server. You make a selection, during configuration, about the type of remote RFC call you want to make. More details on this feature are discussed in the next couple of slides.



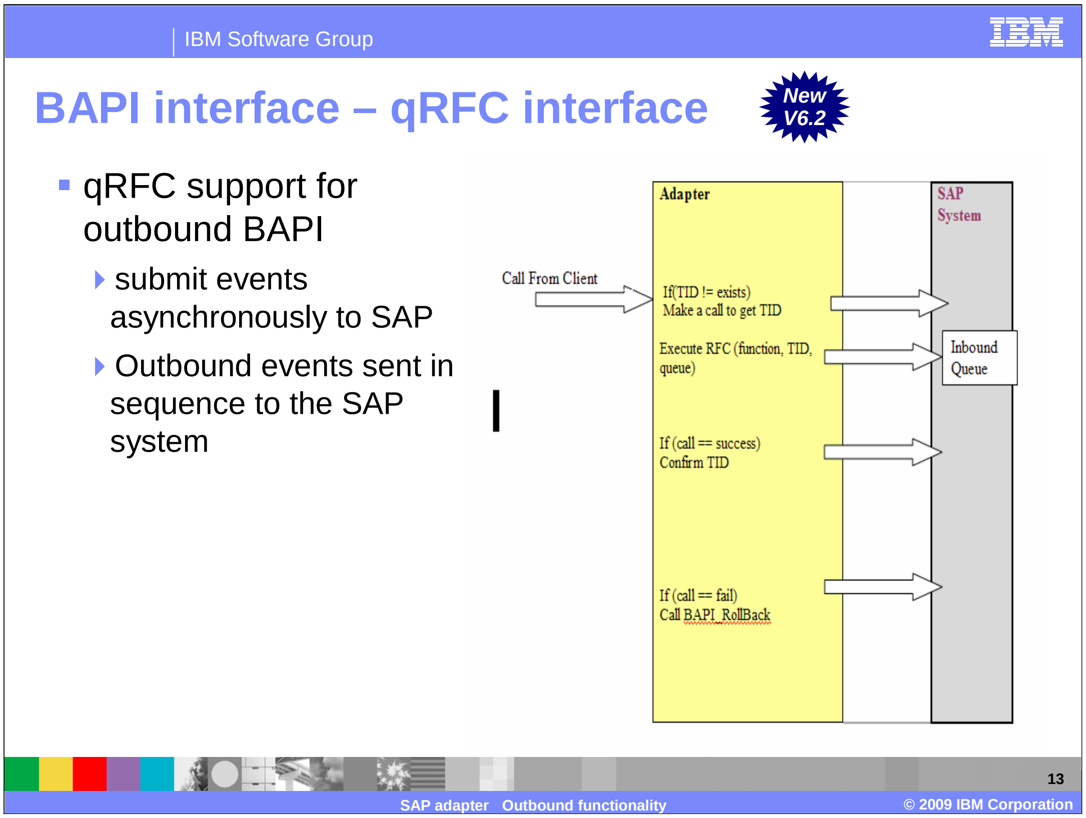
BAPI interface – tRFC support

- tRFC support for outbound BAPI
  - submit events asynchronously to SAP
  - send a request to SAP
  - SAP receives the event
  - control is given back to adapter

If you select asynchronous transactional RFC during configuration, these processing steps occur.

The adapter receives a request from a client application in the form of a BAPI business object. The adapter checks the business object to see whether the SAPTransactionID attribute has a value assigned. If the SAPTransactionID (TID) has a value, the adapter uses that value during processing. If the attribute does not have a value, the adapter makes a call to the SAP server and gets a transaction ID from the SAP server. The adapter converts the BAPI business object to an SAP JCo function call. The adapter uses the transactional Remote Function Call (tRFC) protocol to make the call to the SAP server. The adapter does not wait for a response from the SAP server. After the function data is passed to the SAP application, control returns to the adapter.

If the call to the SAP server fails, the SAP server returns an exception. If the call to the SAP server succeeds but contains incorrect data, no exception is returned to the adapter. For example, if the adapter sends a request that contains a non-existing customer number, the adapter does not respond with an exception indicating that no such customer exists. The adapter passes the TID information to the client.



IBM Software Group

BAPI interface – qRFC interface

New V6.2

- qRFC support for outbound BAPI
  - submit events asynchronously to SAP
  - Outbound events sent in sequence to the SAP system

13

SAP adapter Outbound functionality

© 2009 IBM Corporation

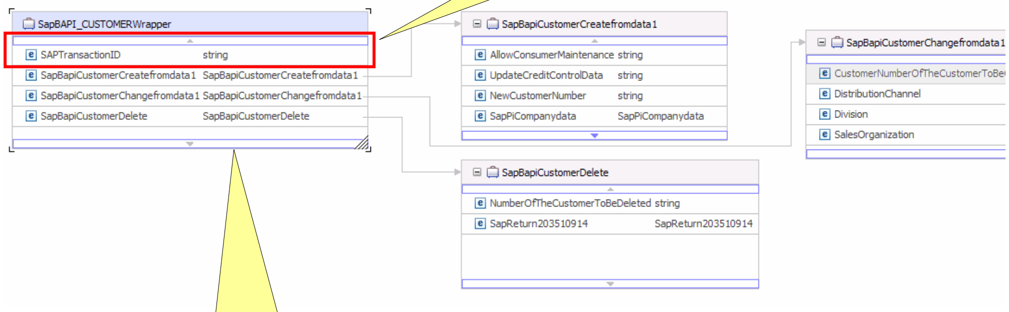
With V6.2 the adapter also supports queued RFC (qRFC) support for making BAPI calls. With the support of qRFC protocol (queued transactional RFC) the adapter will be able to submit events asynchronously to a queue configured in the SAP system. Delivering events to a queue will ensure that sequence in which the calls were made by the client are retained.

You will need to select the “asynchronous queued RFC” option when configuring the adapter using the external service wizard. You also need to provide the queue name to which you want to deliver the call. The queue should already be configured on the EIS before you can select it.

The adapter handles the qRFC the same as the tRFC call which was described in the previous slide. The only difference is the adapter uses the tRFC protocol to make the call to the specified **queue** on the SAP server.

Business object structure

Transaction ID attribute added to the wrapper when you select the asynchronous RFC support for BAPI calls



SapBAPI_CUSTOMERWrapper

SAPTransactionID	string
SapBapiCustomerCreatefromdata1	SapBapiCustomerCreatefromdata1
SapBapiCustomerChangefromdata1	SapBapiCustomerChangefromdata1
SapBapiCustomerDelete	SapBapiCustomerDelete

SapBapiCustomerCreatefromdata1

AllowConsumerMaintenance	string
UpdateCreditControlData	string
NewCustomerNumber	string
SapPiCompanydata	SapPiCompanydata

SapBapiCustomerChangefromdata1

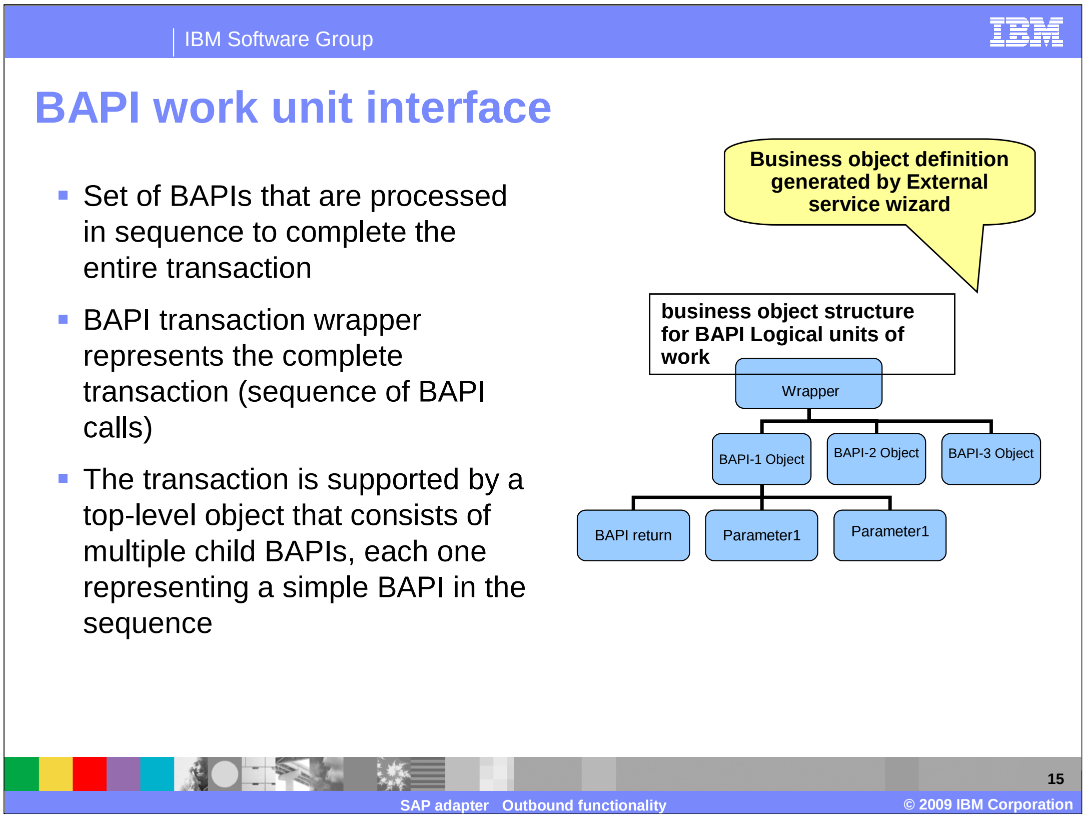
CustomerNumberOfTheCustomerToBe
DistributionChannel
Division
SalesOrganization

SapBapiCustomerDelete

NumberOfTheCustomerToBeDeleted	string
SapReturn203510914	SapReturn203510914

Business object wrapper that contains references to individual business objects that represent the operations you configure

This diagram shows the business object wrapper that contains the transaction ID attribute used for asynchronous RFC support and the business objects that represent each operation you configure. The example above shows an adapter that is configured for three operations: create, update, and delete. Each operation represents a business object that is part of the wrapper. Based on which operation you are invoking from the adapter you only need to populate that business object in the wrapper that represents the operation.



IBM Software Group

BAPI work unit interface

- 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

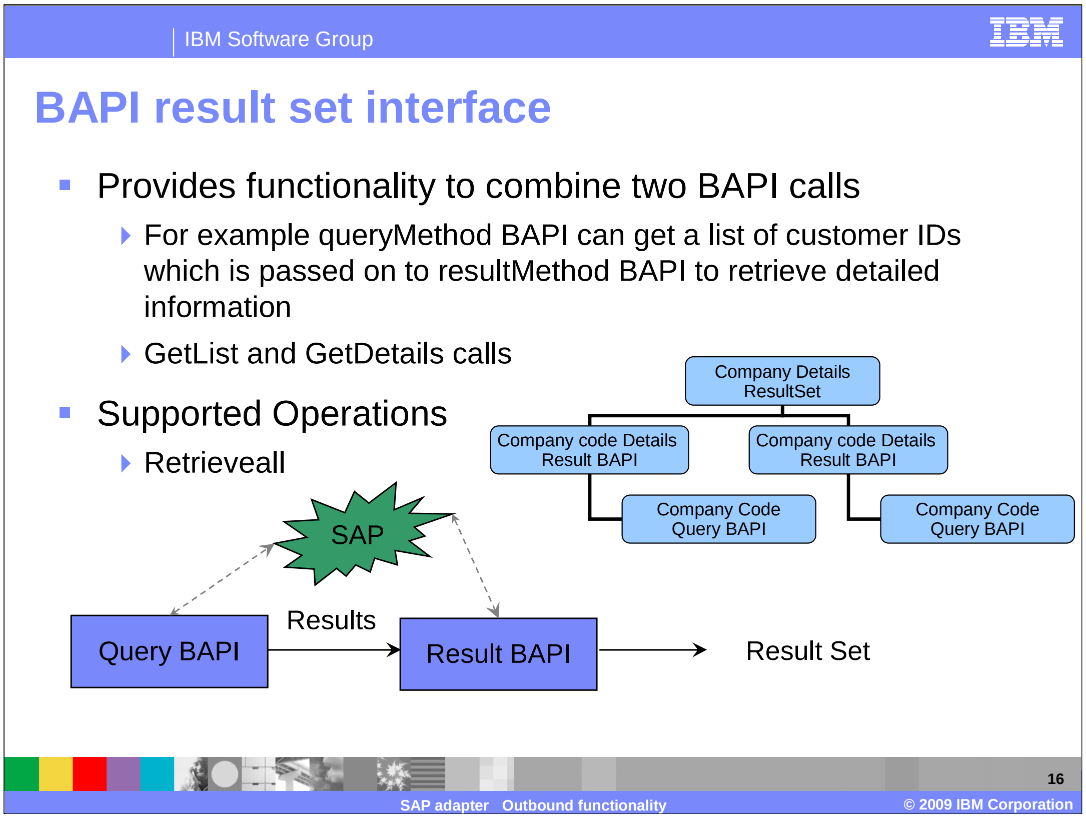
Business object definition generated by External service wizard
business object structure for BAPI Logical units of work
Wrapper
BAPI-1 Object
BAPI-2 Object
BAPI-3 Object
BAPI return
Parameter1
Parameter1

15

SAP adapter Outbound functionality

© 2009 IBM Corporation

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.



IBM Software Group

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

Company Details ResultSet

Company code Details Result BAPI

Company code Details Result BAPI

Company Code Query BAPI

Company Code Query BAPI

SAP

Query BAPI

Results

Result BAPI

Result Set

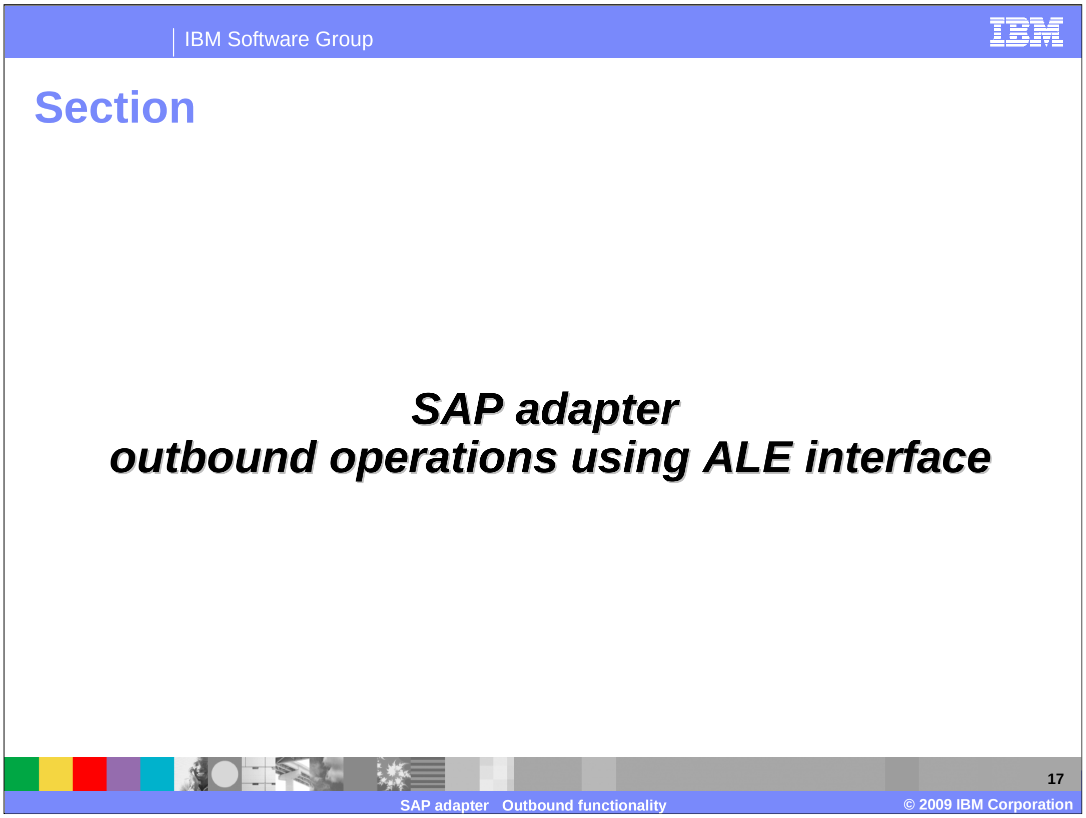
16

SAP adapter Outbound functionality

© 2009 IBM Corporation

This feature provides a new approach to build result set data by combining two BAPI calls. Out of two methods, one method will act as a query, here referred to as "queryMethod ", this method will fetch all the keys which are based on data passed to this method . The second method will fetch all the relevant data, referred to as "resultMethod" method; this method will fetch all the data-details based on key. The queryMethod and resultMethod are combined to form a result set. The methods are represented as a business object.

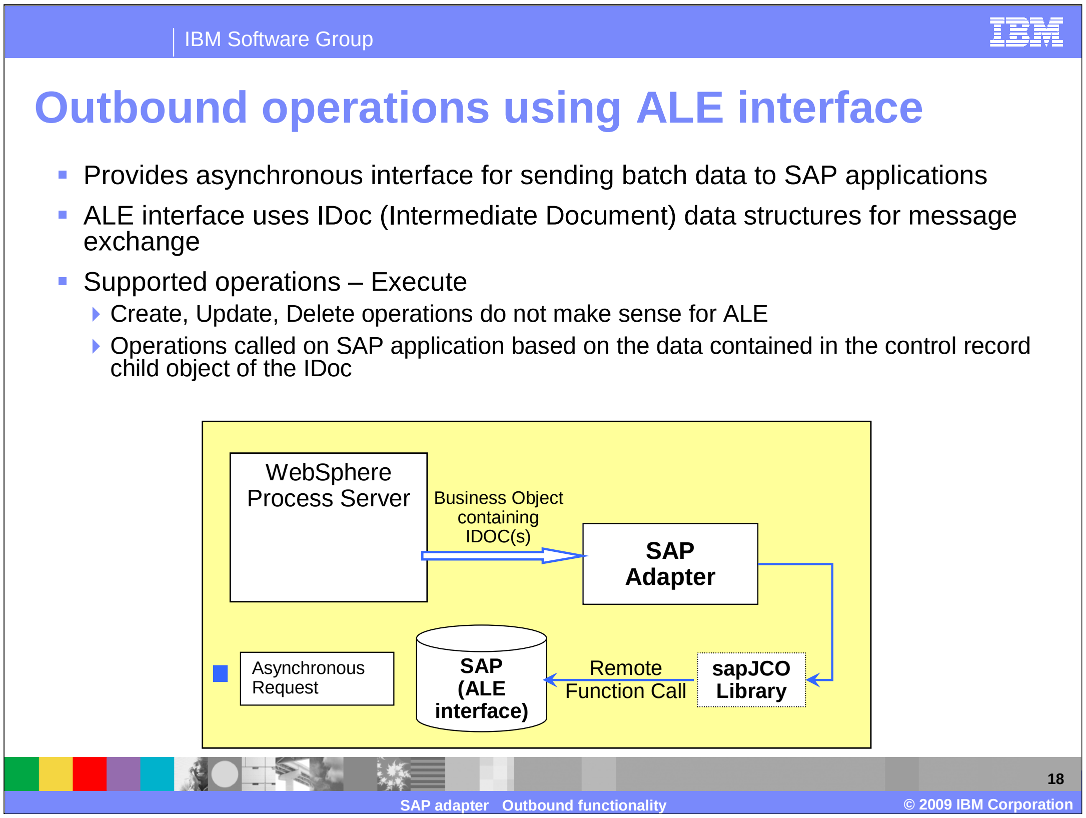
The result set query "queryMethod" method is represented as child of "resultMethod" business object. The "resultMethod" property level application specification information (ASI) provides the foreign-key relationship with "queryMethod" properties. This relationship is used to map key data from queryMethod method to resultMethod method using foreign key. One by one all the key data is set into resultMethod business object.



Section

SAP adapter
outbound operations using ALE interface

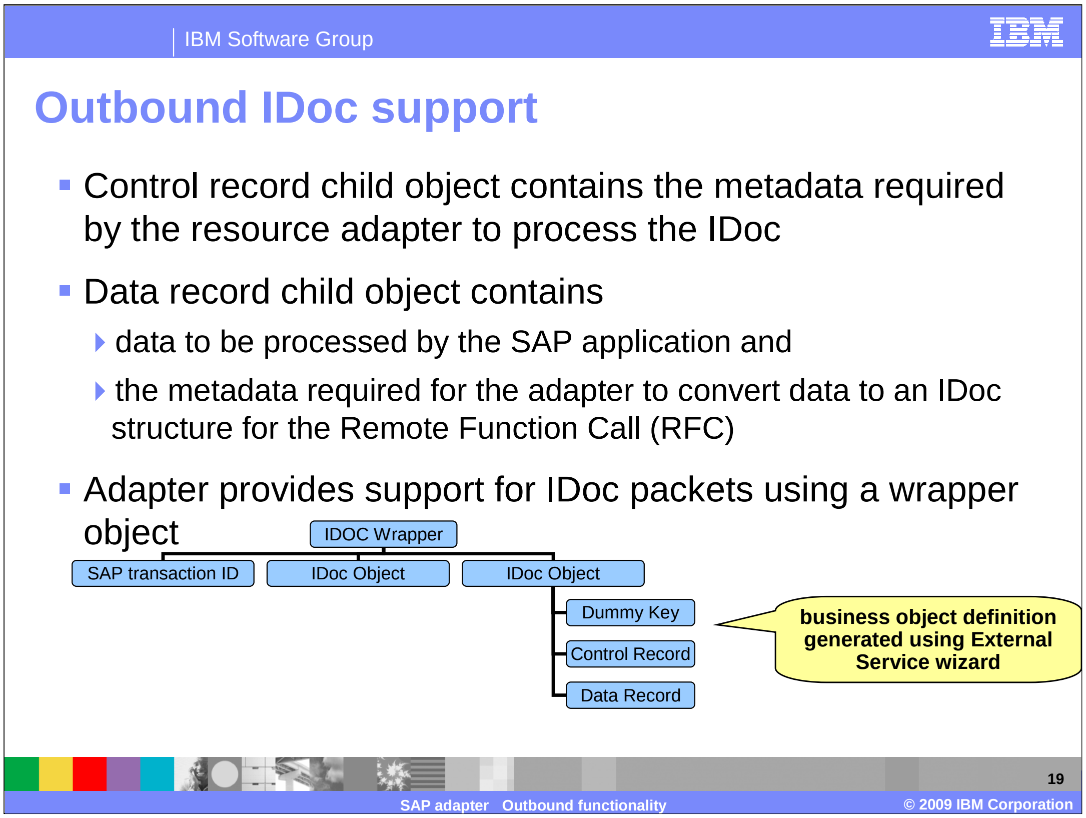
This section covers outbound operations using the ALE interface.



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 business object and then uses SAP JCo function calls to convert the business object 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



IBM Software Group

Outbound IDoc support

- Control record child object contains the metadata required by the resource adapter to process the IDoc
- 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

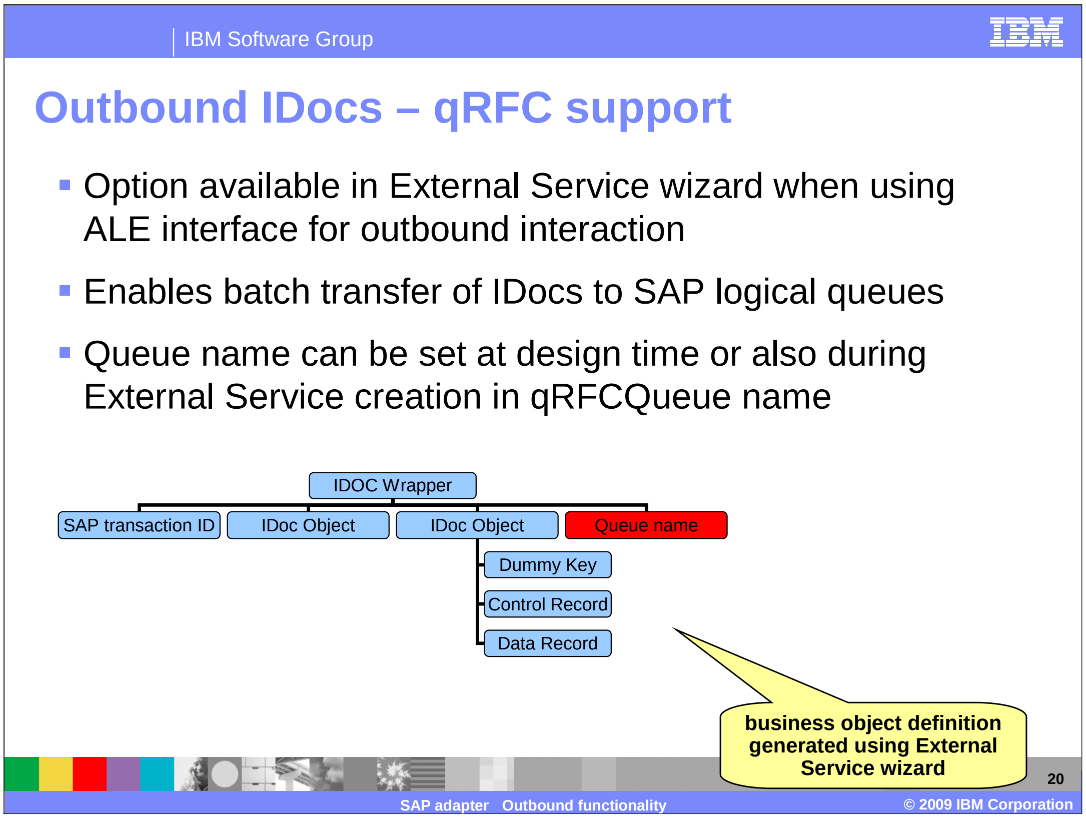
business object definition generated using External Service wizard

19

SAP adapter Outbound functionality

© 2009 IBM Corporation

This diagram shows the support for IDoc packets, where each IDoc is represented by the IDoc object. The wrapper business object 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.



IBM Software Group

Outbound IDocs – qRFC support

- Option available in External Service 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 External Service creation in qRFCQueue name

IDOC Wrapper

SAP transaction ID

IDoc Object

IDoc Object

Queue name

Dummy Key

Control Record

Data Record

business object definition generated using External Service wizard

20

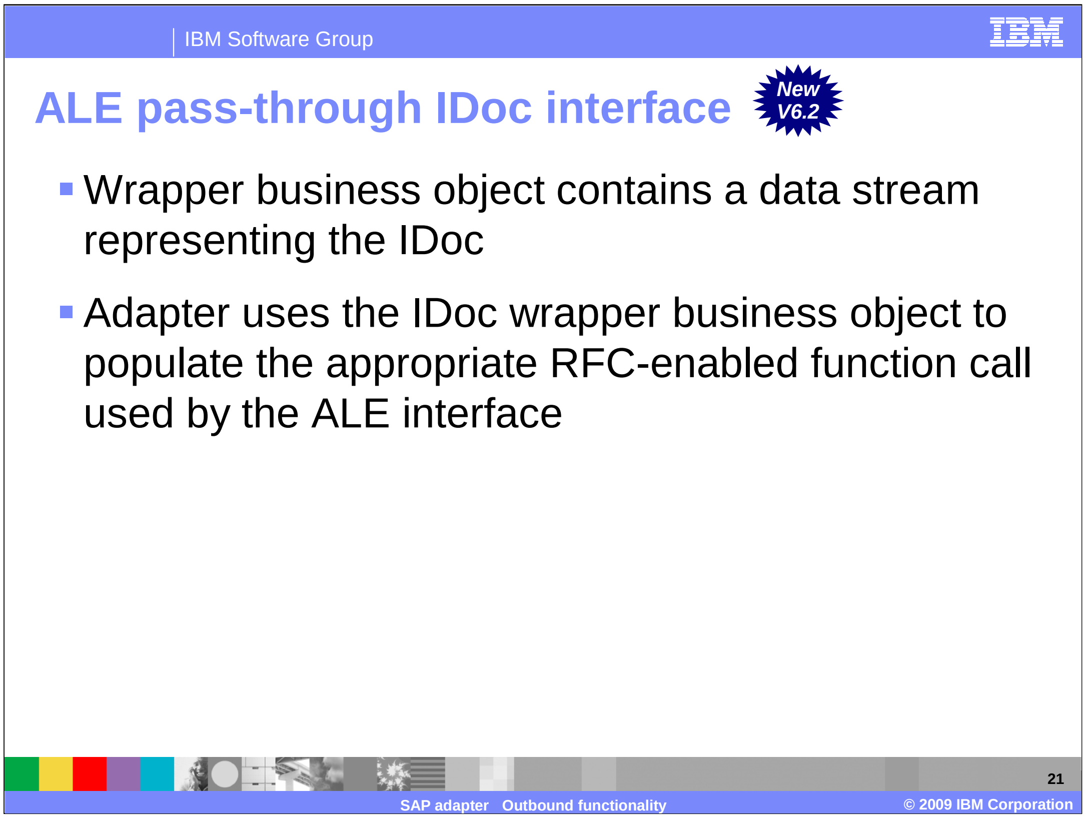
SAP adapter Outbound functionality

© 2009 IBM Corporation

The adapter uses the qRFC protocol to deliver a set of ordered IDocs to a queue on the SAP server or to receive them from a queue on the SAP server. The IDocs are then processed in the order in which they appear in the queue.

The adapter receives a request from a client application. Adapter uses the business object to populate the appropriate RFC-enabled function call used by the ALE interface. The adapter establishes an RFC connection to the ALE interface and passes the IDoc data, in the order in which it is specified in the business graph, to the specified queue on the SAP system. After passing the data to SAP, the adapter returns the transaction ID.

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 external service 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 business object. The adapter finds the list of queues available at runtime and SAP uses them to determine if the queue name is valid.



ALE pass-through IDoc interface

New V6.2

- Wrapper business object contains a data stream representing the IDoc
- Adapter uses the IDoc wrapper business object to populate the appropriate RFC-enabled function call used by the ALE interface

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

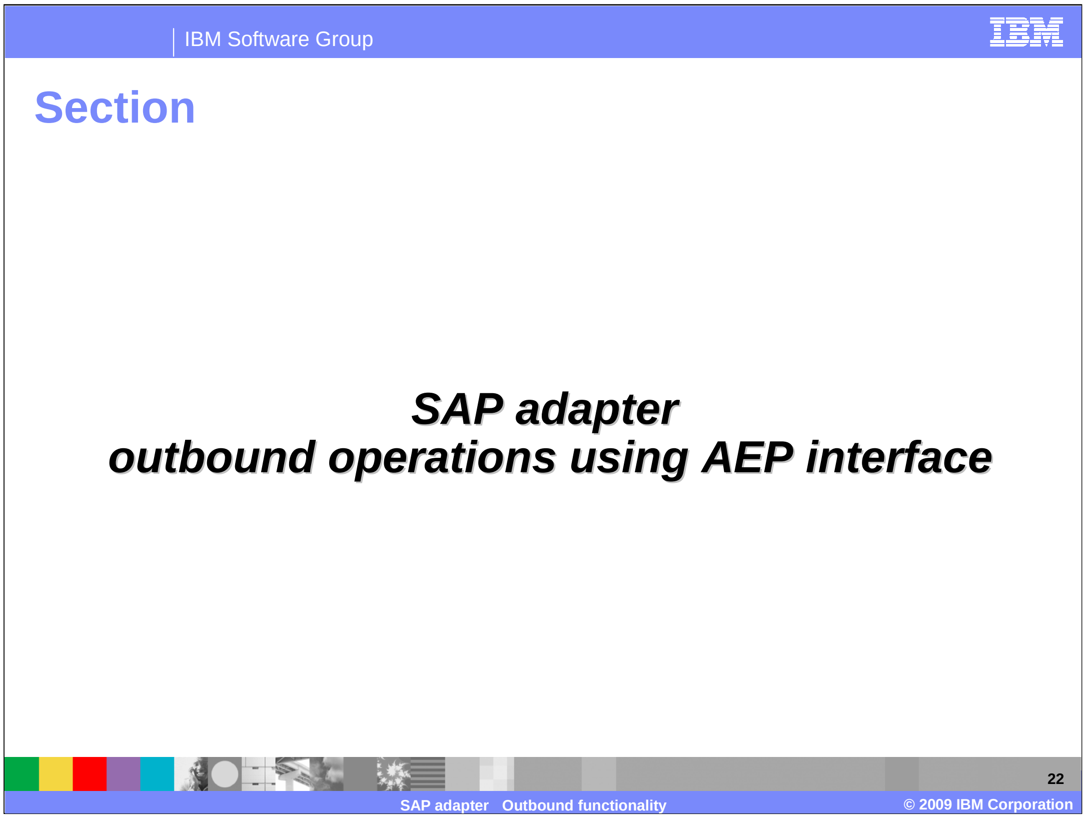
Note that the client application that makes the request uses the interface information that was generated by the external service wizard.

The adapter receives a request, which includes a wrapper business object, from a client application. The wrapper business object contains a data stream representing the IDoc. No separate IDoc business object exists for pass-through IDocs. The adapter uses the IDoc wrapper business object to populate the appropriate RFC-enabled function call used by the ALE interface. The adapter establishes an RFC connection to the ALE interface and passes the IDoc data to the SAP system. If you are using the qRFC protocol, the adapter passes the IDoc data in the order specified in the wrapper business object to the specified queue on the SAP server.

The adapter handles the call in two different ways depending on if the call is managed by local transaction or not.

If the call is not managed by a J2C local transaction, the adapter releases the connection to SAP and does not return any data to the caller. When no exceptions are raised, the outbound transaction is considered successful. You can verify whether the data is incorporated into the SAP application by inspecting the IDocs that have been generated in SAP.

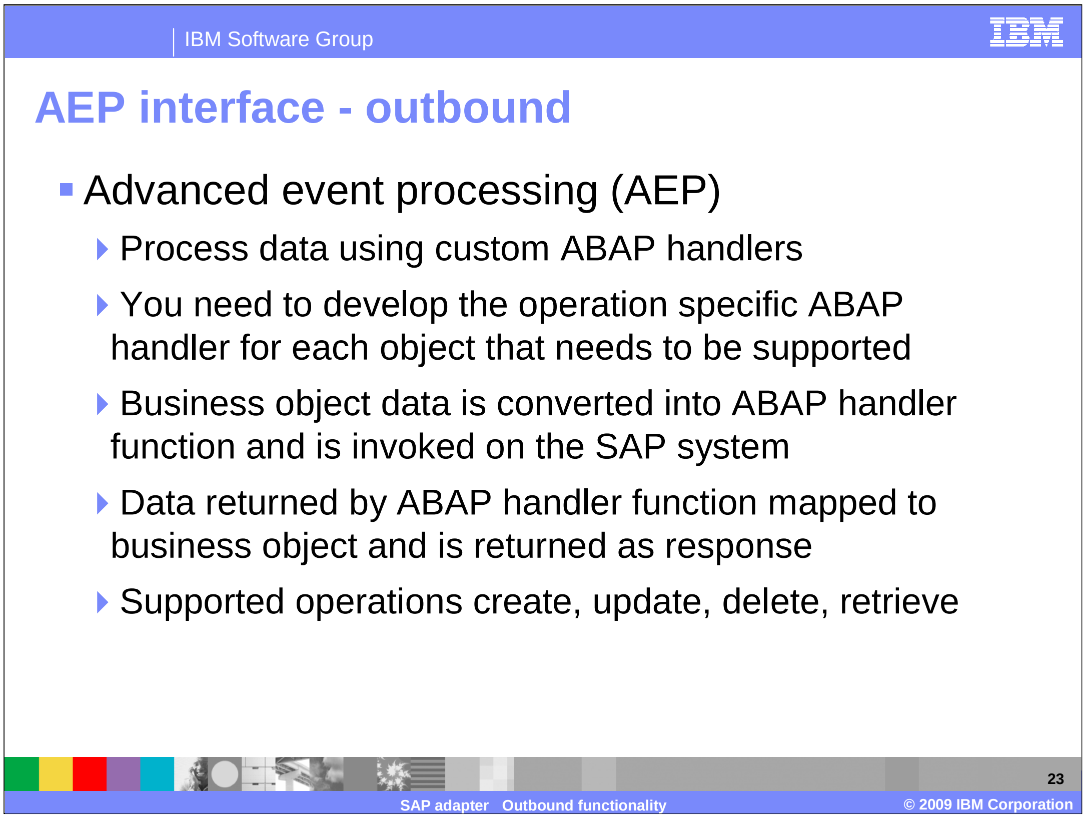
If the call is managed by a J2C local transaction, the adapter returns the transaction ID. The adapter uses the tRFC protocol to support J2C local transactions.



Section

SAP adapter
outbound operations using AEP interface

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



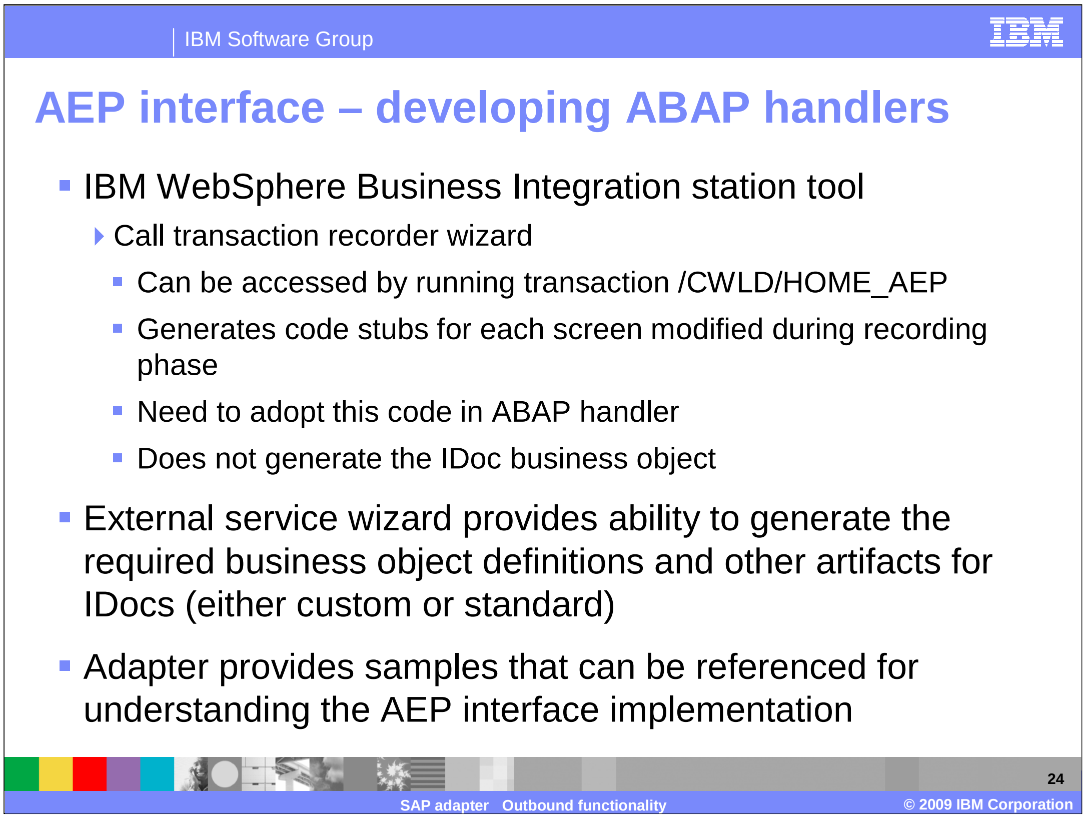
AEP interface - outbound

- 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
  - Business object 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 external service wizard is used to generate the business object 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.

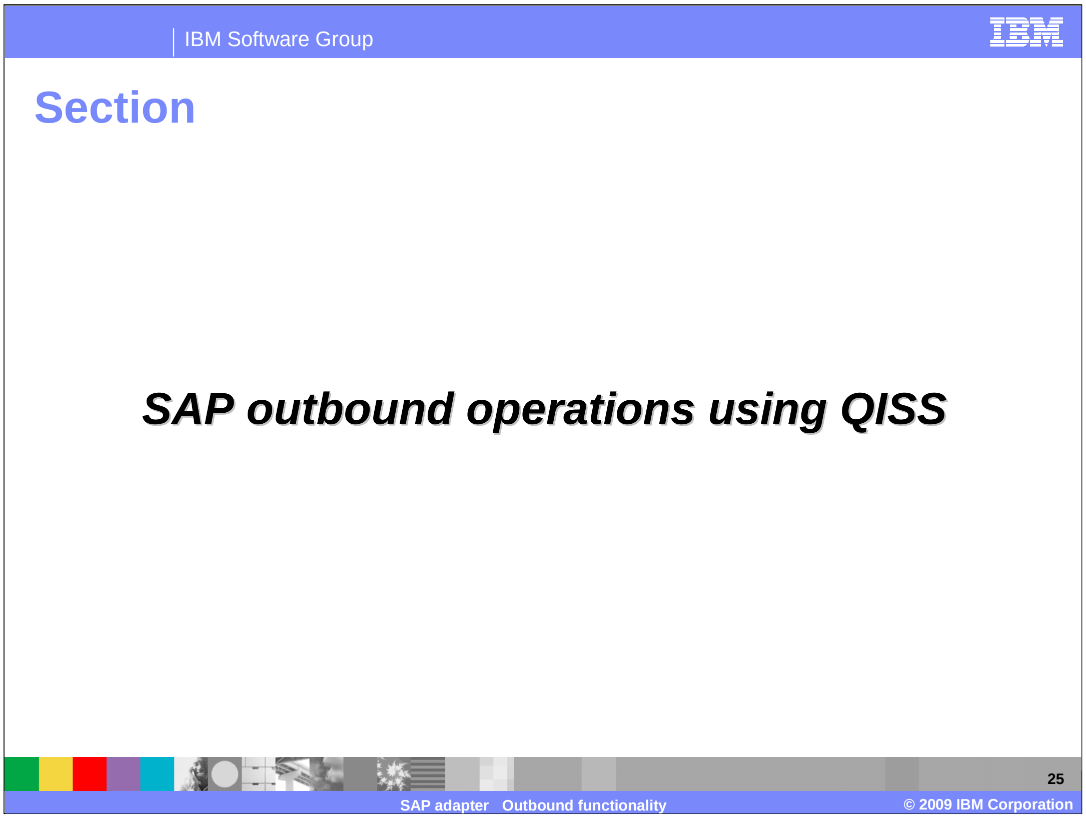
In the outbound processing, the adapter receives the AEP record as a business object. The business object contains the business data along with the metadata. Depending on the operation set, the adapter looks up the ABAP handler to be invoked from the operation application specific information. The adapter converts the business data to handler data format and passes it along with the ABAP handler information to the SAP 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. ABAP component 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 business object 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
- External service wizard provides ability to generate the required business object definitions and other artifacts for IDocs (either custom or standard)
- Adapter provides samples that can be referenced for understanding the AEP interface implementation

SAP adapter 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 Integration Developer provides the external service wizard, which can generate the business object definitions for the custom IDocs you created on the SAP backend.



IBM Software Group

Section

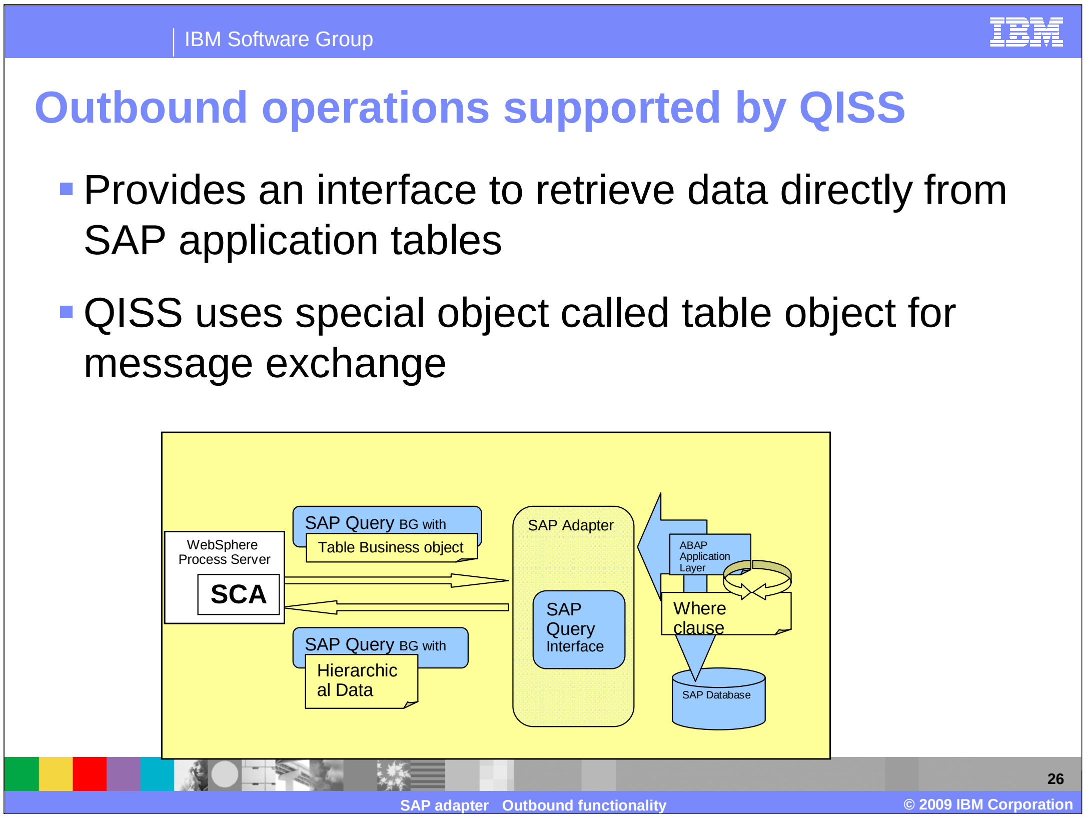
SAP outbound operations using QISS

25

SAP adapter   Outbound functionality

© 2009 IBM Corporation

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



IBM Software Group

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

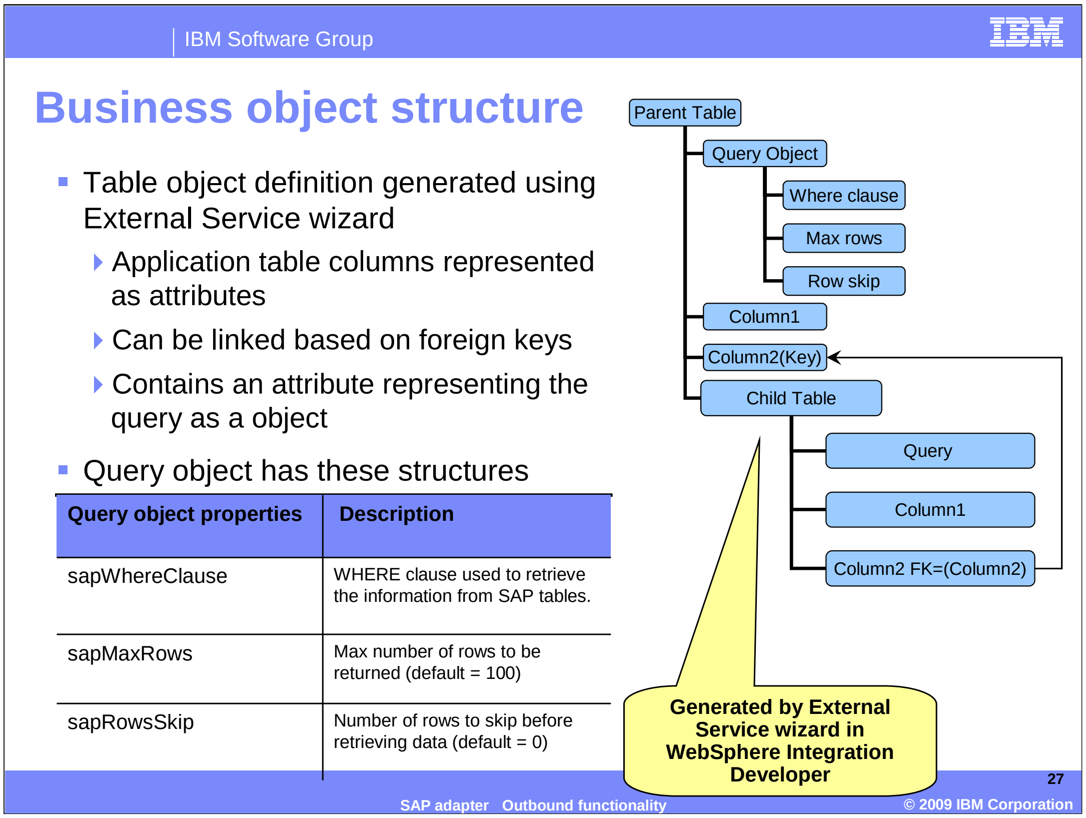
26

SAP adapter Outbound functionality

© 2009 IBM Corporation

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 adapter receives the business object with table information. 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.



IBM Software Group

Business object structure

- Table object definition generated using External Service wizard
  - Application table columns represented as attributes
  - Can be linked based on foreign keys
  - Contains an attribute representing the query as a object
- Query object has these structures

Query 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)

Parent Table
Query Object
Where clause
Max rows
Row skip
Column1
Column2(Key)
Child Table
Query
Column1
Column2 FK=(Column2)

Generated by External Service wizard in WebSphere Integration Developer

27

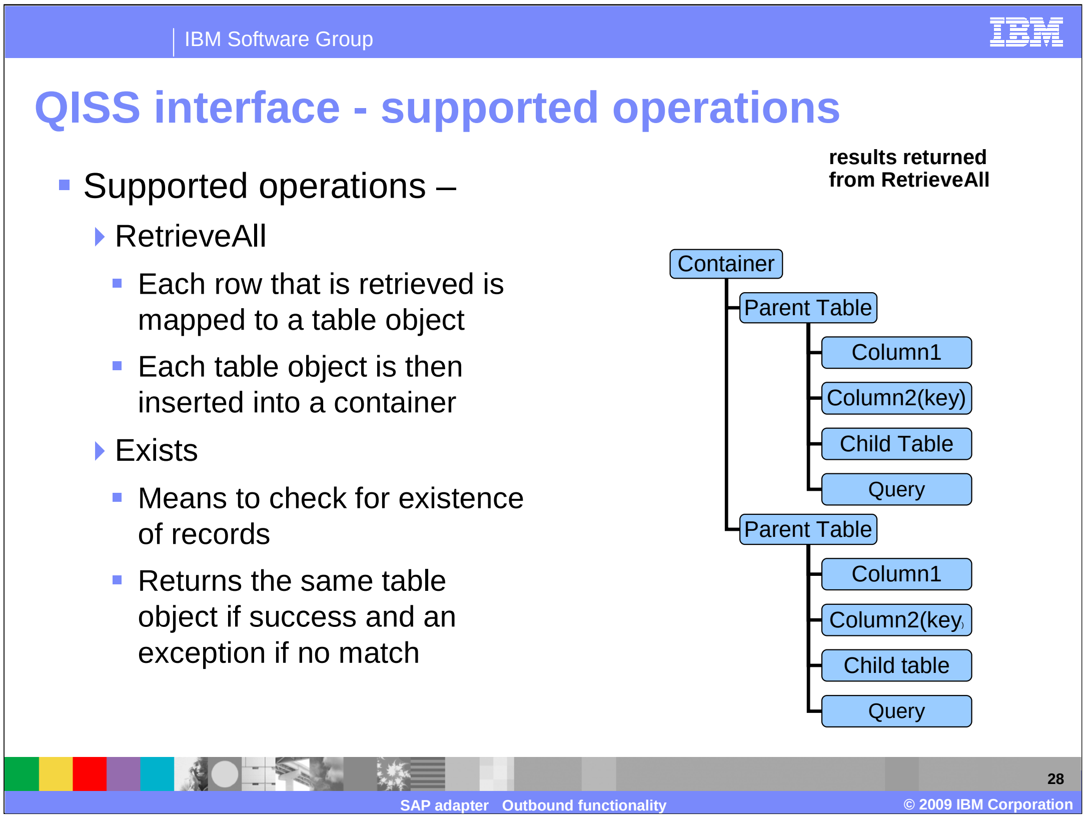
SAP adapter Outbound functionality

© 2009 IBM Corporation

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 are mapped as attributes of the table object. Another important attribute of the table object is the query object. Using the external service 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 external service 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.



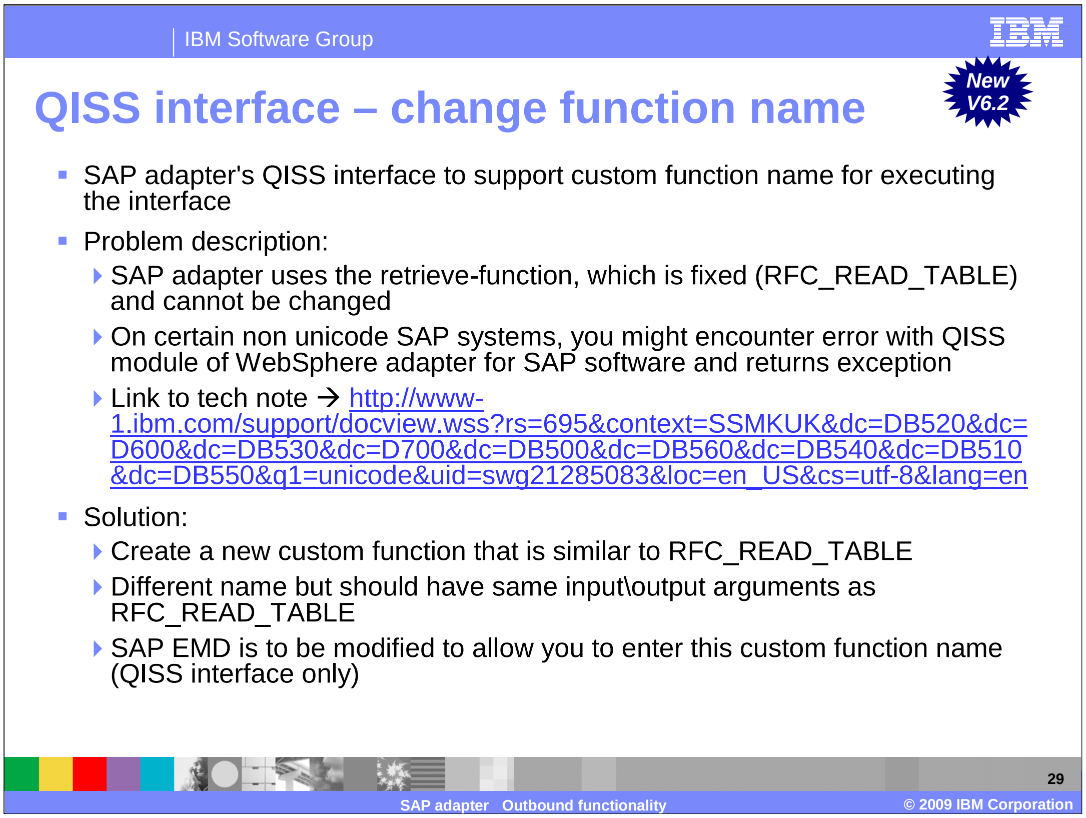
QISS interface - supported operations

- 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

- Container
  - Parent Table
    - Column1
    - Column2(key)
    - Child Table
    - Query
  - Parent Table
    - Column1
    - Column2(key)
    - Child table
    - Query

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 business object with a boolean value.



QISS interface – change function name

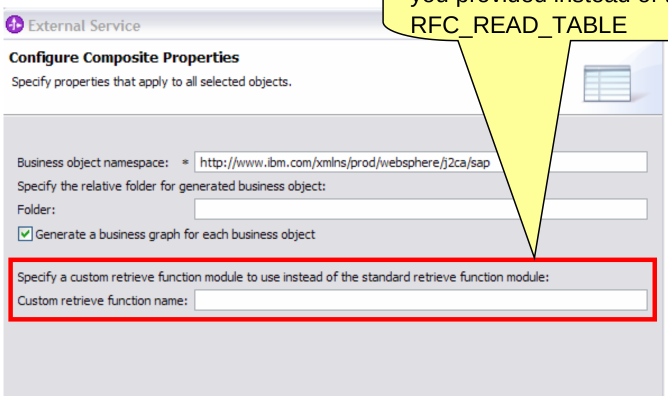
New V6.2

- SAP adapter's QISS interface to support custom function name for executing the interface
- Problem description:
  - SAP adapter uses the retrieve-function, which is fixed (RFC_READ_TABLE) and cannot be changed
  - On certain non unicode SAP systems, you might encounter error with QISS module of WebSphere adapter for SAP software and returns exception
  - Link to tech note → http://www-1.ibm.com/support/docview.wss?rs=695&context=SSMKUK&dc=DB520&dc=D600&dc=DB530&dc=D700&dc=DB500&dc=DB560&dc=DB540&dc=DB510&dc=DB550&q1=unicode&uid=swg21285083&loc=en_US&cs=utf-8&lang=en
- Solution:
  - Create a new custom function that is similar to RFC_READ_TABLE
  - Different name but should have same input\output arguments as RFC_READ_TABLE
  - SAP EMD is to be modified to allow you to enter this custom function name (QISS interface only)

On certain non-UNICODE SAP systems, you can encounter an error with QISS module of WebSphere Adapter for SAP software and returns exception. Currently there is a technote describing this problem. A link is provided on this slide.

SAP adapter uses the RFC_READ_TABLE to retrieve data from SAP tables. To get around the issue mentioned above, the RFC_READ_TABLE function needs to be modified. However, it is not recommended to change the original system function in SAP. The user should create a new custom function that is similar to RFC_READ_TABLE. The new custom function will have a different name but should have the same input and output arguments as RFC_READ_TABLE. In V6.2, external service wizard for SAP adapter provides the means to prompt you for this custom function name.

SAP adapter



External Service

Configure Composite Properties

Specify properties that apply to all selected objects.

Business object namespace: * http://www.ibm.com/xmlns/prod/websphere/j2ca/sap

Specify the relative folder for generated business object:

Folder:

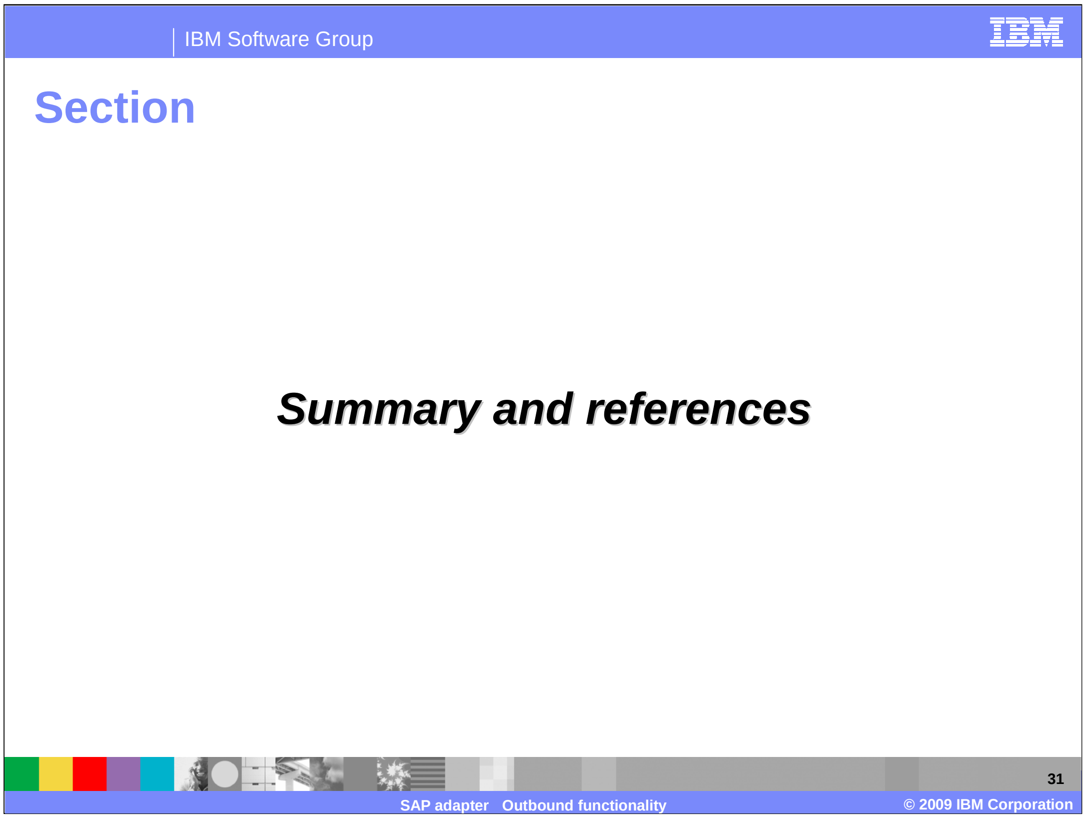
☑ Generate a business graph for each business object

Specify a custom retrieve function module to use instead of the standard retrieve function module:

Custom retrieve function name:

If custom function name is set, adapter will use the custom function you provided instead of the default RFC_READ_TABLE

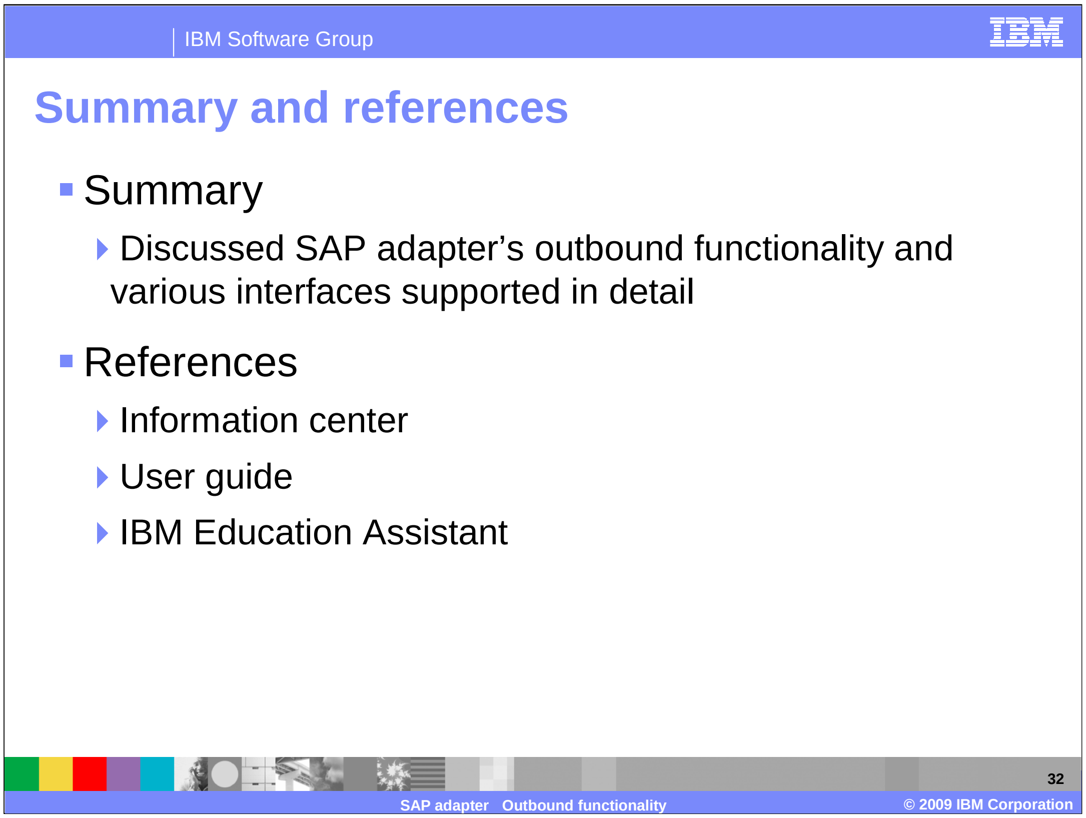
The slide shows the screen capture of the panel from external service wizard where you can provide your custom function name.



Section

Summary and references

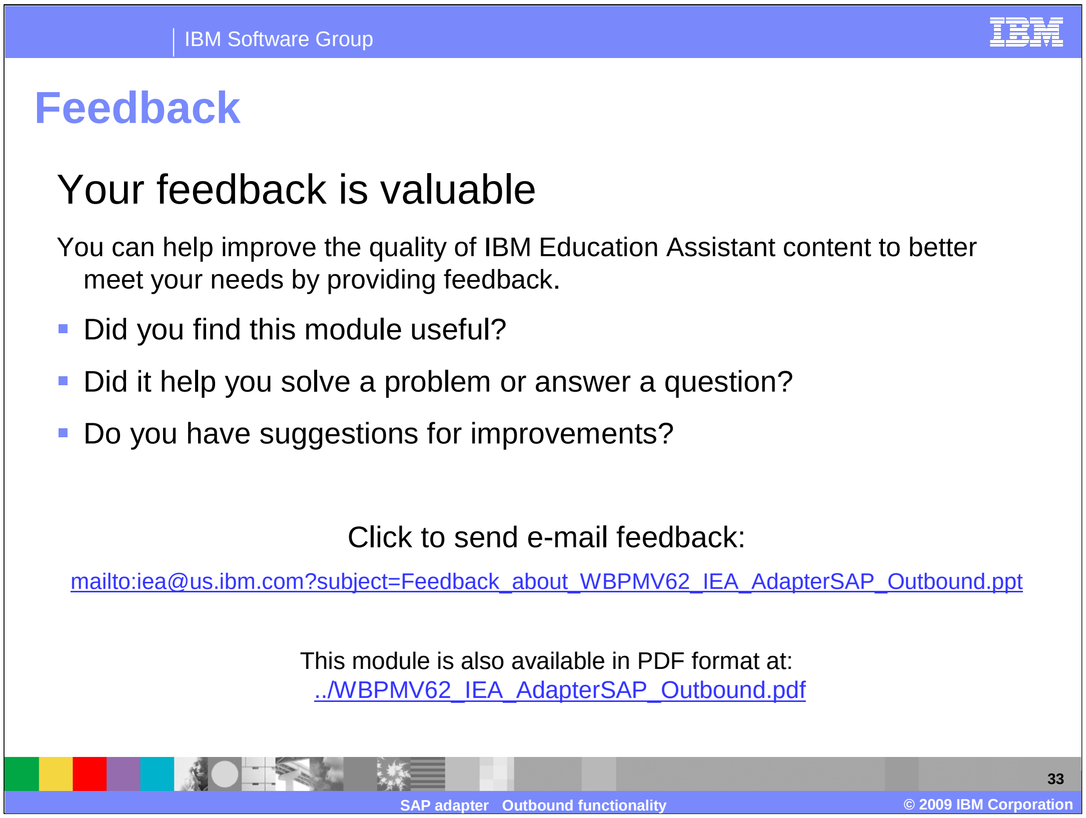
This section provides a summary and references.



Summary and references

- Summary
  - Discussed SAP adapter’s outbound functionality and various interfaces supported in detail
- References
  - Information center
  - User guide
  - IBM Education Assistant

This presentation covered the details of SAP adapter’s outbound 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.



IBM Software Group

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:

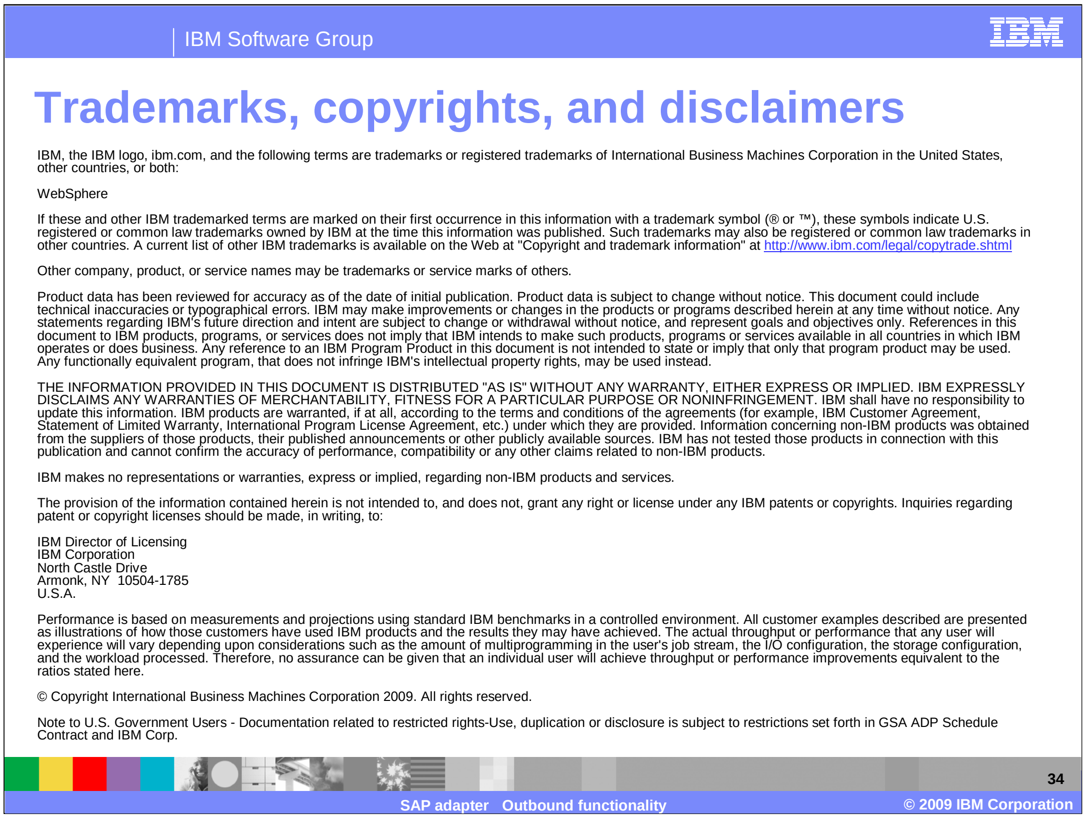
mailto:iea@us.ibm.com?subject=Feedback_about_WBPMV62_IEA_AdapterSAP_Outbound.ppt

This module is also available in PDF format at:
../WBPMV62_IEA_AdapterSAP_Outbound.pdf

33

SAP adapter   Outbound functionality

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



Trademarks, copyrights, and disclaimers

IBM, the IBM logo, ibm.com, and the following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

WebSphere

If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of other IBM trademarks is available on the Web at "Copyright and trademark information" at http://www.ibm.com/legal/copytrade.shtml

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

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements 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.

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.