



IBM Software Group

WebSphere® Message Broker V6.1

WebSphere Adapter for Siebel Business Applications



@business on demand.

© 2008 IBM Corporation
Updated January 29, 2008

This presentation will focus on the WebSphere Adapter for Siebel Business Applications in WebSphere Message Broker V6.1.

Agenda

- Terminology and overview
- Adapter connection wizard
- Outbound operations
- Inbound operations and event manager
- Error handling
- Transactions
- Problem determination
- Summary

The agenda is as shown on this page.

Section

Siebel terminology and Siebel adapter nodes overview

This section covers the overview of the WebSphere Adapter for Siebel.

Siebel application overview

- Siebel application architecture is a layered structure containing
 - ▶ 1. User interface objects layer
 - ▶ 2. Business objects layer
 - ▶ 3. Data objects layer
- Business object layer includes business services, business components, business objects and integration objects
- Adapter supports business objects and components, business services and integration objects in this release
- The adapter uses the Java™ APIs provided by the Siebel Java data bean to communicate with the Siebel object manager for data exchange

4

Siebel application architecture has three layers.

The user interface object layer deals with the visual elements that users interact with.

The second one is the business objects layer. This includes the business services, business components, business objects and Siebel integration objects. A Siebel business service is an entity in Siebel that encapsulates and simplifies the use of a set of functionality.

The third layer is the data objects layer.

The Siebel adapter nodes support business objects and components, business services, and any integration objects through the business services.

To allow external applications to exchange data with the Siebel application, Siebel provides a set of interfaces known as Siebel Object Interfaces. The APIs, in the Siebel-provided Java data bean layer jar file, handle the exchange of data by using the Siebel business objects and business components. The adapter uses the code in the jar file to communicate with the Siebel object manager for data exchange.

Siebel business service

- Siebel business service
 - ▶ an entity in Siebel that encapsulates and simplifies the use of a set of functionality
 - ▶ like an object in an object-oriented programming language.
 - ▶ Not tied to specific objects, but rather operate on objects to achieve a particular goal
 - ▶ has properties and methods (with arguments) and maintains a state

The Siebel business service is like an object in an object-oriented programming language. It has a set of functions that can be invoked. A Siebel business service is not tied to a specific object but operates on one or more objects to achieve a particular function.

Siebel business service (continued)

- Siebel business services are based on following types of higher level classes:
 - ▶ CSSEIDataSyncService – Used for data synchronization services
 - ▶ CSSService – Any business service. This is typically extended for by other services
 - An example is the CSSEISiebelAdapter class which inherits from CSSService class and maps to the EAI Siebel Adapter business service.
 - ▶ CSSEIDTScriptService - Used for data transformation services
 - ▶ CSSEISiebelAdapter – Used for the Enterprise Application Integration (EAI) Siebel Adapter
- The adapter only supports business services that are of the above specified class types
 - ▶ These business services includes generic business services such as
 - the 'EAI Siebel Adapter',
 - Applications Services Interfaces such as 'Siebel Contact' and
 - other built-in or custom business services

Siebel business services are based on the four types of higher level classes shown on this page.

The CSSEIDataSyncService is used for data synchronization services.

The CSSService represents any business service. Other services extend this class.

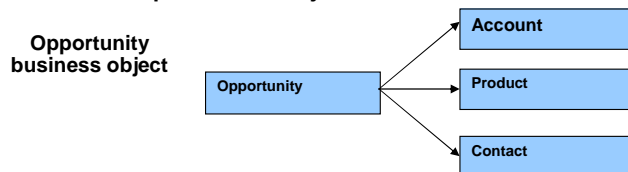
The CSSEIDTScriptService is used for data transformation services

The CSSEISiebelAdapter is used by the “EAI Siebel Adapter,” which is a generic business service in Siebel.

The adapter only supports business services that are of the above specified class types. Any service that does not fall into one of the four classes, is not supported by the adapter.

Siebel business objects and components

- Siebel business components are logical abstractions of one or more database tables
 - ▶ Defines structure, behavior and information related to a particular subject
- Business objects are customizable, object-oriented building blocks of Siebel applications.
 - ▶ Define the relationships between different business component objects and contain semantic information .



7

A business component defines the structure, the behavior, and the information displayed by a particular subject such as a product, contact, or account.

Siebel business components are logical abstractions of one or more database tables.

The information stored in a business component is typically specific to a particular subject and is typically not dependent on other business components.

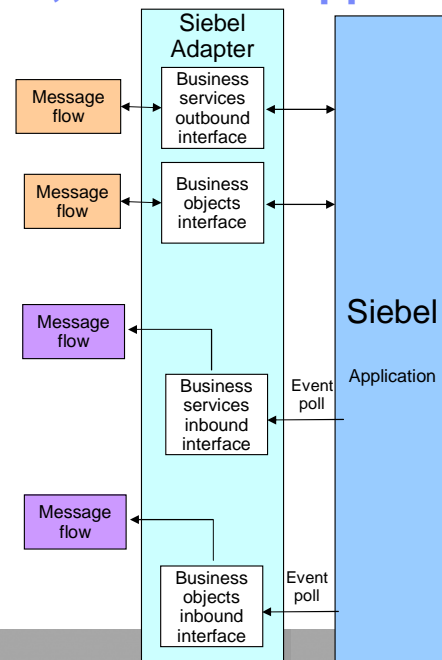
Business objects are customizable, object-oriented building blocks of Siebel applications.

Business objects define the relationships between different business component objects and contain semantic information. Siebel business objects group one or more business components into a logical unit of information

In the picture shown here, the “Opportunity” business object consists of opportunity, contact, and product business components. The “Opportunity” business component, which is the primary business component, dictates the information of the other business components in a parent-child relationship.

Siebel adapter node outbound, inbound support

- Outbound and inbound calls supported using:
 - ▶ Business services interface
 - ▶ Business objects interface



The diagram on this page shows the high level flow and the components for the outbound and inbound calls to and from Siebel. Siebel adapter nodes support both business services and business objects for outbound and inbound calls.

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

Adding dependent libraries to toolkit & runtime

- Toolkit
 - ▶ For Siebel 7.7 , 7.8, add Siebel.jar and Siebel_enu.jar to Java build path of connector project
 - ▶ For Siebel 7.5 are SiebelJI_Common.jar, SiebelJI_enu.jar and SiebelJI.jar
- Runtime
 - ▶ Configurable service is the recommended approach for adding the dependent jar files required to access the EIS.
- Use “mqsichangeproperties” to setup/report the dependencies:
 - ▶ To set up the Siebel.jar file required to connect to Siebel system

```
mqsichangeproperties brk6 -c EISProviders -o Siebel  
-n jarsURL -v <jarfile_location>
```

- ▶ To report the dependencies:

```
mqsireportproperties brk6 -c EISProviders -o Siebel -r
```

There are a set of dependent files required by the adapter at runtime and during design time when working with the adapter connection wizard. The wizard prompts you for the files required during design time. Once you finish the development of the application, before you deploy the application to the broker runtime, you need to set the dependency files required by the adapter to connect to the enterprise information system. You can make use of the mqsichangeproperties command to accomplish this task. The slide shows the format of the command for setting the Siebel.jar file. These dependent files are used by the adapter to connect and communicate with the Siebel system. Based on the version of Siebel you are using, different files are needed as dependencies

Section

Siebel Request node operations high level flow

This section covers the outbound operations high level overview for the Siebel Request node

Outbound – High level flow

- Siebel request node receives a message representing the Siebel 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
 - ▶ From the metadata, the Adapter determines if the message represents a business service or a business component
 - ▶ If it is a business service, the adapter converts the message data to the appropriate Siebel function call.
 - ▶ If it is a business object, the resource adapter either creates, updates or deletes in the Siebel application based on operation specified

11

The high level flow of the outbound operations is discussed in this slide.

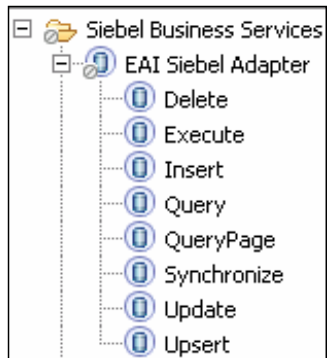
The adapter models the Siebel function calls as messages. The adapter extracts the data from the message, and determines if the message represents a business service or a business component.

If the incoming message represents a service, the function name specified in the interaction spec is used as a business service method. The adapter creates a copy of the input message, which is populated with results after the function call. The adapter builds the required Siebel PropertySets based on the input message and invokes the business service. The output Siebel PropertySets are then populated in the output message.

If the incoming message represents a business component, the adapter creates a copy of the input message, which is populated with results after running the function you specified. The adapter either creates, updates or deletes in the Siebel Application based on the method name mentioned. Any result is then populated in the output message accordingly.

Siebel business service: 'EAI Siebel Adapter'

- It is a general purpose business service that allows reading and writing integration objects



EAI Siebel Adapter method	Description
Synchronize	Makes the values in the application match those in the business object
Insert	Creates the integration object
Update	Updates the integration object, and synchronizes all children
Upsert	Performs synchronize, but no deletes are done.
Delete	Performs a delete of the integration object
Query	Retrieves the integrations objects that match the input business object
QueryPage	Performs a query but only returns a set of records.
Execute	Performs a combination of operations on the components of the integration object

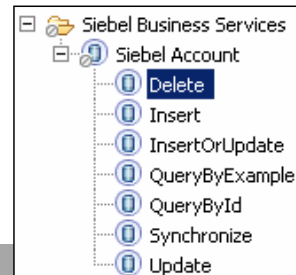
The table here lists the methods supported by the EAI Siebel Adapter business service. As indicated earlier, the EAI Siebel Adapter is a general purpose business service that allows reading and writing other Siebel integration objects.

Siebel application service interface

- Built-in business services that provide integration object specific services – Some examples are as follows:

- ▶ Siebel Account – Integrates with the Account Interface integration object
- ▶ Siebel Contact – Integrates with the Contact Interface integration object

Built-in Business Service Method	Description
QueryByExample	Retrieves the integration object based on non key values
QueryById	Retrieves the integration object based on the primary rowID
InsertOrUpdate	Inserts or updates the integration object depending on if it already exists



13

The methods for the built-in business services are listed in the table. For the EAI Siebel Adapter general purpose business service, either the QueryByExample or QueryById method can be used to retrieve the specific Siebel integration object.

Siebel business object and component support

- Incoming message represent the primary business component of a business object.

Operation	Description
Create	'Create' operation creates the business component
Update	'Update' updates the Siebel application with the incoming object.
Retrieve	In order to see the values of the business component 'Retrieve' operation can be used.
RetrieveAll	'RetrieveAll' can retrieve multiple instances of the same business component and populates it as the container message and returns it.
Delete	'Delete' deletes the business component and its children from Siebel. By default it is a cascade delete in Siebel.
Exists	'Exists' operation checks for the existence of the incoming business objects in Siebel. Output business object 'ExistsResult' is returned with the Boolean value populated.

The incoming message represents the primary business component of a business object. It is hierarchical if the business component has different child relationships defined.

The adapter supports the create, update, retrieve, retrieve all, delete and exists operations. For the Create operation, the adapter always creates a new entry for the business component in the Siebel application. Each of the children is created first and then the parent. After all the children are created, the parent is saved. Saving after creating all the children has the advantage of rolling back all the child creates in case the top level create fails. The Update operation involves comparing the business object retrieved from Siebel with the incoming message. Creating, updating and deleting the children is done based on the comparison. This amounts to processing all the children in the Siebel application and making them to be the same as the incoming message. The Retrieve operation involves retrieving the Siebel business component corresponding to the specified key in the incoming message. The Delete operation does a cascade delete by default. The Exists operation checks for the existence of the incoming message representing an object in Siebel. The output business object is returned populated with a Boolean value.

Section

Siebel Input node operations high level flow

This section covers the inbound operations.

Inbound operations: High level flow

- Inbound operations are based on events being triggered when any record in the Siebel Business application is created, updated or deleted
- Event notification requires the creation of event in the Siebel database
 - ▶ “IBM Events” Siebel business component is the default table used to track events happening in the Siebel EIS
 - ▶ component and table name can be customized, but not the schema.
 - ▶ Event component lists the type of event, the corresponding business graph and the status of the event
 - ▶ Triggers need to be placed on the base Siebel business objects. Siebel supports Visual Basic® scripts and Siebel eScript embedded in the Siebel Business Component event handlers to populate the event table

Inbound operations are based on events being triggered when a Siebel record or object is created, modified or deleted. The Siebel developer must create a custom Siebel business component to store the events that are being triggered as a result of the updates to the Siebel records. The schema for the custom Siebel business component cannot be changed. A sample custom Siebel business component, called, “IBM Events” is provided with the adapter.

Triggers need to be inserted in the Siebel business objects that need to be watched for events. Sample triggers are also provided. Siebel supports Visual Basic scripts and Siebel eScript for writing event triggers.

The rest of the flow is in the next page.

Inbound operations: High level flow (continued)

- Adapter polls the Siebel event component for new events and fetches the events
 - ▶ “Once and only once” event delivery is supported using the Event Persistence database
- If the event corresponds to business component, adapter retrieves the business component and creates a message that is sent to the rest of message flow
- If the event corresponds to business service, adapter retrieves the integration object and creates the message which is sent to the rest of message flow

The adapter polls for the events looking in the custom Siebel business component that was created to store events within Siebel. When a new event is detected by the adapter, based on the event status, the adapter retrieves it and makes an entry in its event persistence store. If the event corresponds to a business component, the adapter retrieves the business component by making use of the retrieve operation and creates a message conforming to the message definitions created using the adapter connection wizard. The adapter retrieves the integration object represented by the event in case of business service and creates a message that is sent to rest of message flow.

Event table schema

Fields	Description	Example
XID	This field replaces the EDT table	
Object Key	The unique identifier which identifies the business object row for which the event was created.	AccountId=1-20RT
Object Name	The object for which the event was detected.	EAIAccountInterface
Event Type	The type of the event	Create
Priority	Event priority	1
Status	Event Status. Initially set to READY_FOR_POLL.	0
Description	Any comment associated with the event – Not required	Account Interface Event
Event Id	Id of the event row.	
Event Timestamp	The time stamp for the event. Format MM/dd/yyyy HH:mm:ss	02/24/2005 11:37:56

18

Listed on this page is the table schema for the custom Siebel business component that is created to store events within Siebel. These events are created by the triggers inserted in Siebel business objects that need to be watched for events.

Prerequisite

- Siebel EIS setup for event management
 - ▶ Create custom event project
 - Sample provided
 - ▶ Insert triggers
 - Visual Basic and Siebel eScript supported
 - Samples provided

Setting up Siebel EIS includes preparing the custom event project needed to store the events within Siebel. There are two main tasks. First, the custom event project must be created in Siebel. This is where the generated events are stored within Siebel. The adapter looks at this event project for new events or changing the status of existing events. Sample event project is provided in the samples directory. The schema of the custom event project must be the one provided in the user guide. The name of the custom event project can be changed, but the schema must be the one expected by the adapter. The event generated triggers must be inserted in the components whose events can be monitored by the adapter. Siebel supports Visual Basic scripts and Siebel eScripts. Sample triggers are provided by the Adapter. To Create the Event table in Siebel, you must be familiar with Siebel tools.

Section

Configuration: Siebel properties for outbound and inbound

This section covers some of the important properties that you can configure in the adapter connection wizard for outbound and inbound interaction styles

Important outbound properties

Property	Description
User name	User name to login to the Siebel EIS
Password	Password for the corresponding user name
Siebel connect string	<p>Connection string needed to connect to Siebel EIS</p> <p>For Siebel 7.5 (uses default port 2320, hence no port # specified) Protocol://machinename/enterprisename/objectmanager/servername Example: siebel://mysiebel.ibm.com/siebel/SSEObjMgr_enu/sebldev1</p> <p>For Siebel 7.7 (uses port 2321 which is not a default): protocol://machinename:portnumber/enterprisename/objectmanager</p>
Language	Language to be used - Example: enu
resonateSupport	Specifies whether the adapter should use Attach/Detach calls on the SiebelDataBean. Default: True

Some of the specific Siebel Adapter custom properties are shown here. The description for most of them is self explanatory.

The Siebel connect string has its own format, as shown in the table with examples.

Resonate Support is a feature exposed by the Siebel Application to allow for more efficient use of Siebel Connections. Instead of doing expensive login and logoff operations on the connection, a Siebel Client can detach and attach a Siebel Java DataBean (connection). This is transparent to you, and happens when the application server does a clean up of the connection then later on attempts to re-establish a connection. If resonateSupport is set to true, then instead of doing a login or logoff during this phase, the adapter attaches and detaches.

Important inbound properties

Property	Description
User name	User name to login to the Siebel EIS
Password	Password for the corresponding user name
Siebel connection string	<p>Connection string needed to connect to Siebel EIS</p> <p>For Siebel 7.5 (uses default port 2320, hence no port # specified): Protocol://machinename/enterprisename/objectmanager/servername Example: siebel://mysiebel.ibm.com/siebel/SSEObjMgr_enu/sebldev1</p> <p>For Siebel 7.7 (uses port 2321 which is not a default): protocol://machinename:portnumber/enterprisename/objectmanager</p>
Siebel repository	Name of the Siebel repository – Example: "Siebel Repository"
Language	Language to be used - Example: enu
Event component name	Specifies the name of the Siebel Component for the event table
resonateSupport	Specifies whether the adapter should use Attach/Detach calls on the SiebelDataBean. Default: True

The properties for the inbound are shown in the table. They are same as the outbound except the Event Component Name, which specifies the name of the Siebel custom event component that has been created to store the events from the Siebel components.

Transaction and security

- Transaction
 - ▶ Siebel application does not support local or XA transaction
 - ▶ Hence this feature is not supported by the adapter

Siebel application does not support transaction, so there is no end to end transaction available from the adapter to and from the Siebel application

Problem determination

■ Logs

▶ Local error log

- Windows®: Event log
- UNIX®: syslog

▶ Broker logs

- Eclipse error logs: located under <workspace>\.metadata\.log
- Broker domain event logs:
 - log is stored and managed by the Configuration Manager
 - can be information, errors, or warnings
 - view the messages in the editor area of the Broker Administration perspective

■ Trace

▶ User trace

▶ Service trace

On Windows, the local error log is the Windows Event log's application view. On UNIX and Linux systems, the local error log is the *syslog*. Where syslog messages are sent depends on how you configure your UNIX or Linux system. The Eclipse error log captures internal errors that are caused by the operating system or your code and are logged in the .log file under your workspace. Broker domain log information is written to the broker domain Event Log. This log is stored and managed by the Configuration Manager. The Event Log displays messages about events that occur within the broker domain, such as deploying topology or topics hierarchy configuration. The messages can be information, errors, or warnings. You view the messages in the editor area of the Broker Administration perspective. The broker domain Event Log shows errors that are generated by all workbench users in that particular domain. You can enable the user and service traces by using the *mqsichangetrace* command

Section

Summary and references

This section covers the summary and references.

Summary and references

- **Summary**
 - ▶ Discussed Siebel Adapter in details covering the architecture, operations, problem determination
- **References**
 - ▶ Information center
 - ▶ User guide



In summary, this presentation covered the details of Siebel adapter nodes functionality using the business services and the business objects interface.

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_SiebelAdapterNodes_detail.ppt

This module is also available in PDF format at: [./WMBV61_SiebelAdapterNodes_detail.pdf](._/WMBV61_SiebelAdapterNodes_detail.pdf)



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

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

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

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

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.

© Copyright International Business Machines Corporation 2008. All rights reserved.

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

