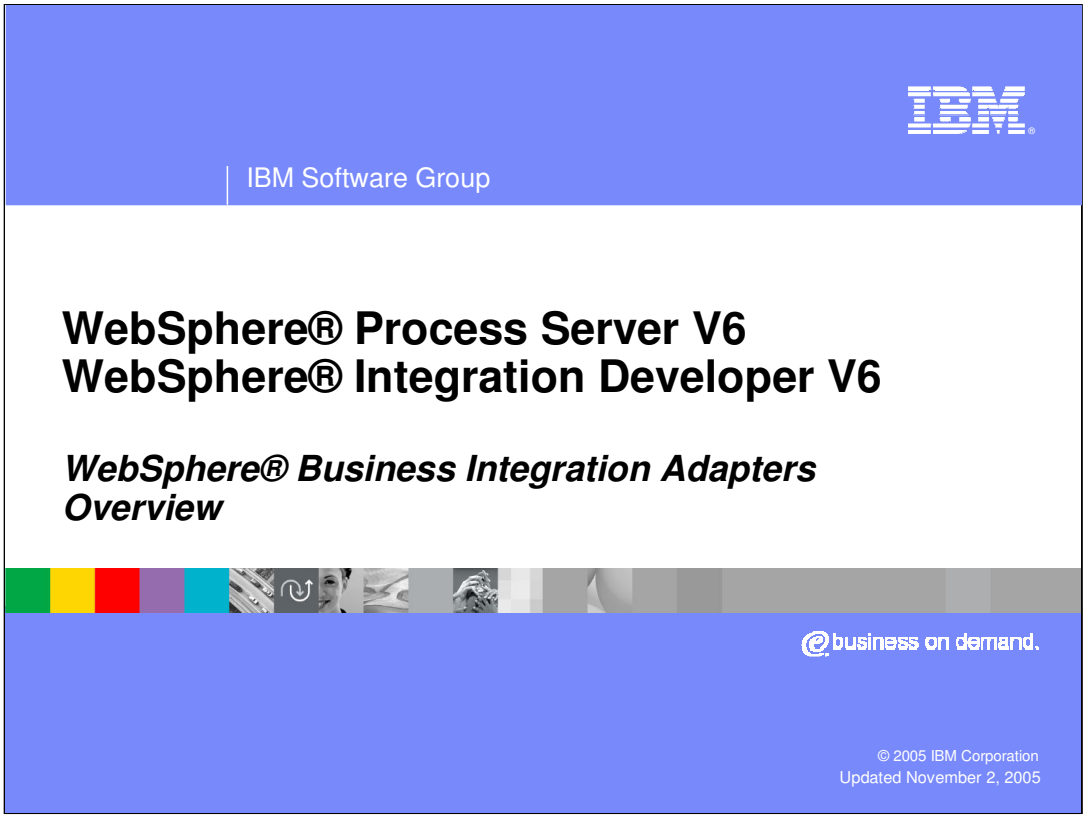


This presentation will cover the WebSphere Business Integration Adapters.

Goals

- Provide Overview of WebSphere Business Integration Adapters with WebSphere Process Server
- Understand Development Tools
- Provide Overview of Administration using WebSphere Process Server Administrative Console

The goals of this presentation are to provide an overview of WebSphere Business Integration Adapters and how they integrate with WebSphere Process Server, discuss development tools, and administration of adapters using the integrated administration capability of the WebSphere Process Server administrative console.

Agenda

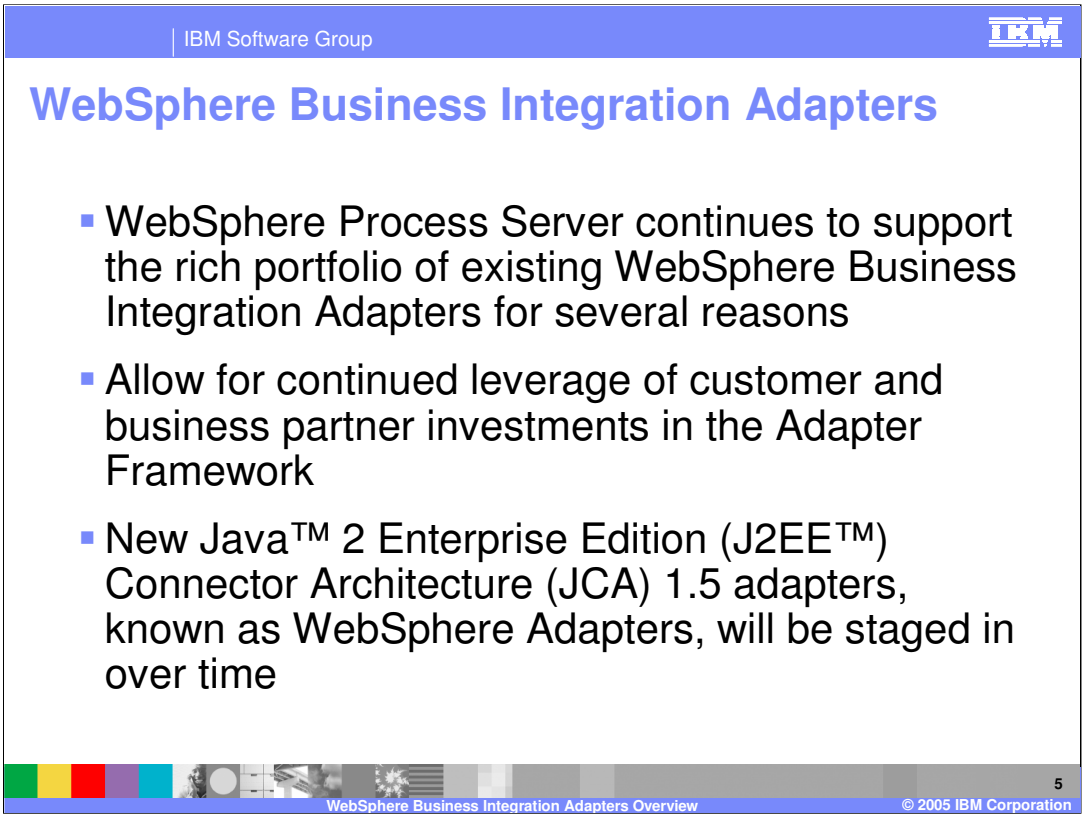
- **Overview**
- Architecture / Big-Picture
- Development Tools
- Administration
- Event Monitoring
- Logging and Tracing
- Summary

This section will provide an overview of the WebSphere Business Integration Adapters.

WebSphere Business Integration Adapters

- WebSphere Business Integration Adapters
  - consist of a collection of software, Application Programming Interfaces, and tools to enable applications to exchange business data through an integration broker
- Each business application requires its own application-specific adapter to participate in the business integration process

WebSphere Business Integration adapters consist of a collection of software, APIs providing native communication with the backend Enterprise Information System (EIS), and tools that enable you to configure business objects and adapters. Adapters provide communication between the EIS and the integration broker, which in this case is the WebSphere Process Server. While WebSphere Business Integration Adapters are capable of communicating with other WebSphere broker products, this presentation will focus on WebSphere Process Server. Each back-end system or business application requires a specific adapter.



WebSphere Process Server continues to support the rich portfolio of WebSphere Business Integration Adapters, allowing customers and business partners to leverage their investment and continued use. New J2EE Connector Architecture (JCA) 1.5 WebSphere Adapters will be staged in over time.

WebSphere Business Integration Adapters

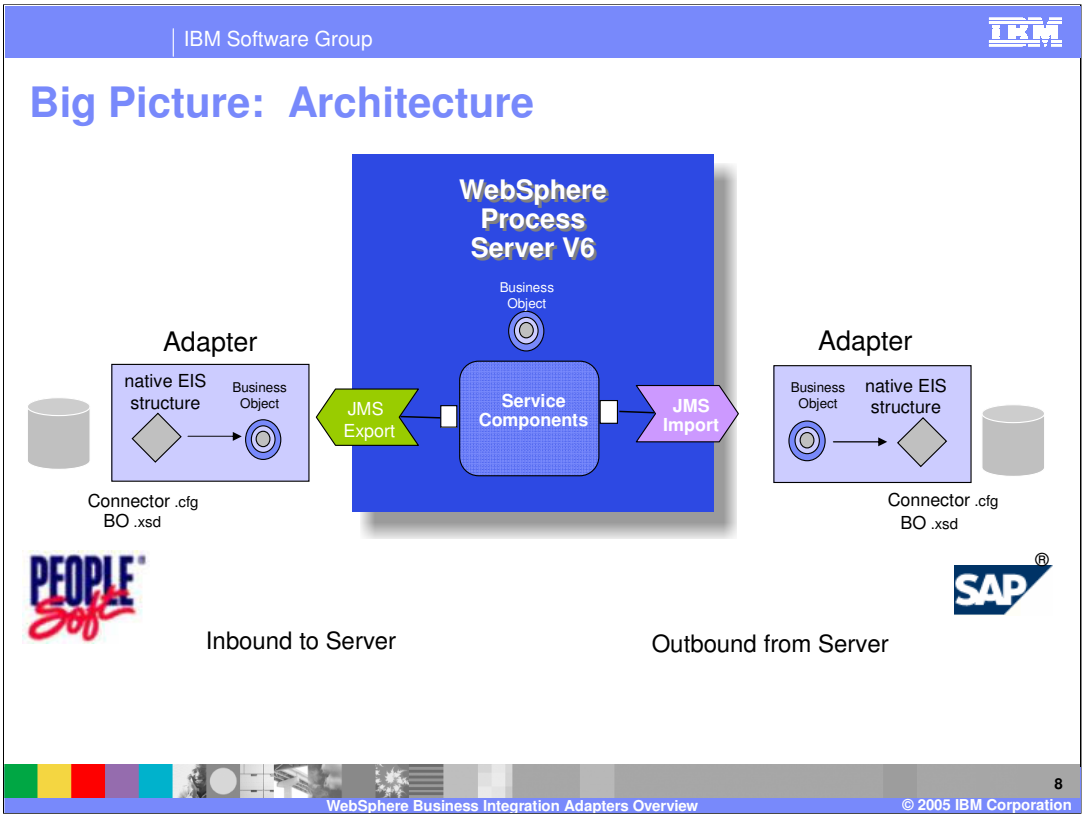
- WebSphere Business Integration Adapters offer support for connectivity to older versions of business applications

Another important aspect of the support for existing WebSphere Business Integration Adapters is the capability to provide connectivity to older versions of business applications.

Agenda

- Overview
- **Architecture / Big-Picture**
- Development Tools
- Administration
- Event Monitoring
- Summary

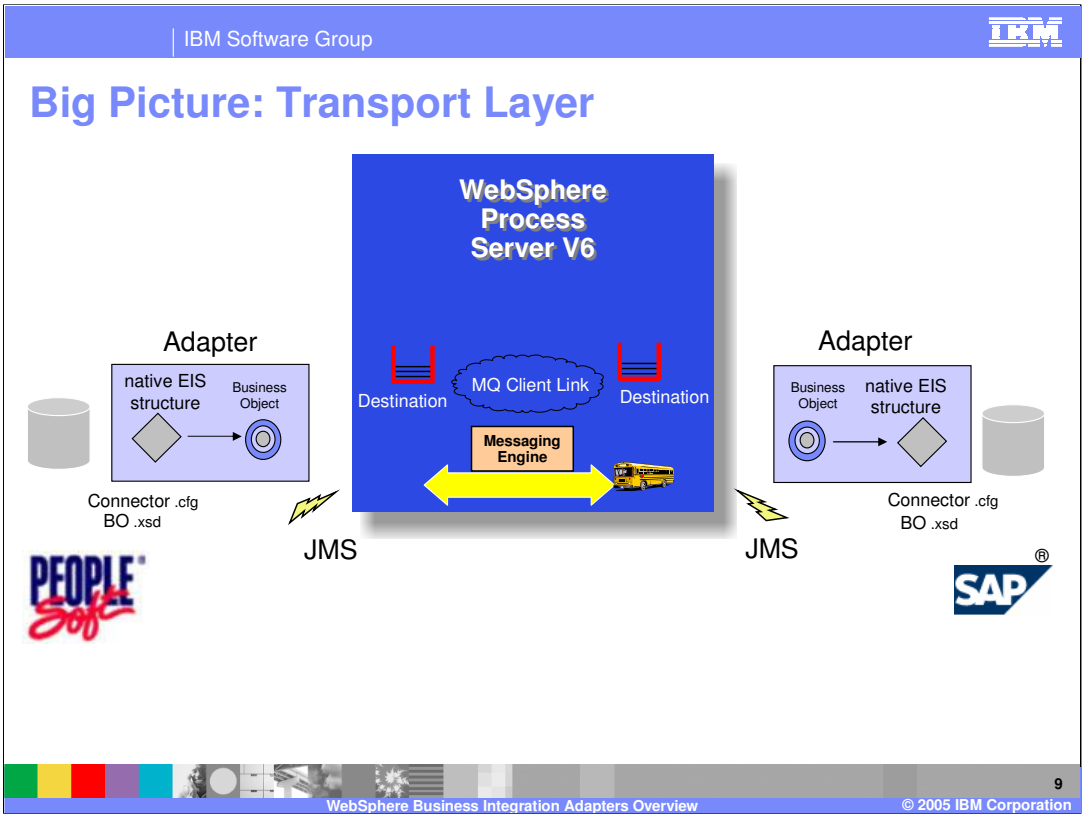
This section will cover WebSphere Business Integration Adapter architecture.



A common pattern used in business integration involves the need to synchronize semantically similar data among various back-end business application systems. This scenario is illustrated here with two different back-end systems, each integrated with the business integration application running on the WebSphere Process Server, through the use of WebSphere Business Integration Adapters. Note that integration of these adapters is done using the same components and programming model as other Service Component Architecture (SCA) integration applications.

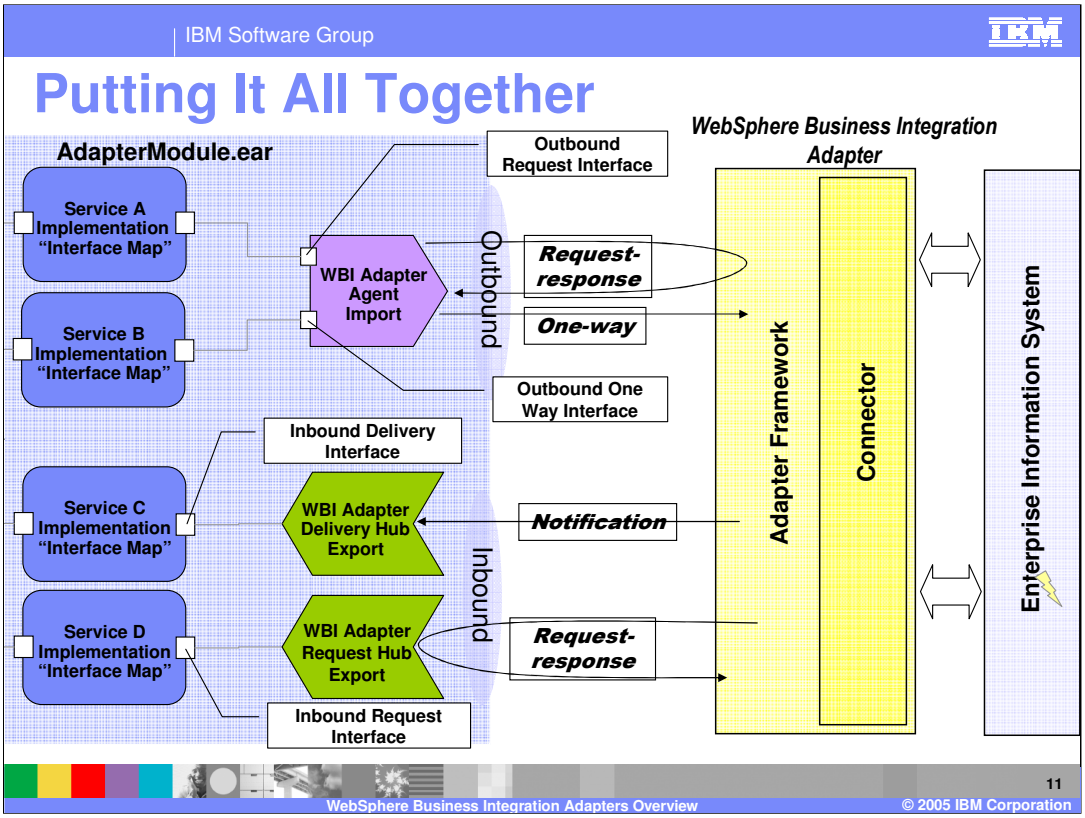
The center of this graphic represents the WebSphere Process Server with a business integration application. The business integration application is made available for invocation to other services outside of the SCA module, through a Java Messaging System (JMS) export. The business integration application is able invoke other services outside of the SCA module through the use of a JMS import. The adapters communicate with the back-end systems using the application specific data structure or business object, and are configured using the connector configuration file, or connector.cfg. When a business object is passed inbound to the WebSphere Process Server through the export, it is converted to a format understood by the WebSphere Process Server by a data binding that is part of the export. When a business object is passed outbound to the adapter, it is converted to a format understood by the adapter by a data binding that is part of the import.

This data synchronization pattern can also incorporate mapping of the business object from an application specific format to a generic format. See the presentation on interface and business object mapping for additional information.



WebSphere Business Integration Adapters communicate with WebSphere Process Server using the JMS protocol. The connector configuration file is configured to use WebSphere Application Server as the broker type. WebSphere Process Server's default messaging support, Service Integration Technologies messaging, using MQClientLink is automatically configured the first time an Enterprise Application (EAR) containing WebSphere Integration Adapter artifacts is installed. Destinations specific to each adapter are automatically configured.

Imports and exports are generated by WebSphere Integration Developer. They are used to facilitate communication between the adapters and the WebSphere Process Server. Exports listen for incoming events on the JMS delivery destination (queue) or synchronous request destination (queue). Through the export and data bindings, the incoming business object is converted from the WebSphere Business Integration format to the WebSphere Process Server business object or business graph format. The business object is then passed on to the SCA component. Imports receive request business object or business graphs from an SCA component, convert the object from the WebSphere Process Server format to the WebSphere Business Integration business object format, and write the message to the JMS request destination (queue).



Taking a look at a "big picture" view of the WebSphere Business Integration Adapter environment, you see four interaction patterns are supported. The AdapterModule application installed to the WebSphere Process Server is shown on the left. The WebSphere Business Integration Adapter, consisting of the adapter framework and connector specific to the back-end application, are pictured on the right communicating with the Enterprise Information System. Outbound communication from the WebSphere Process Server application is enabled through the use of an import file that contains an outbound request interface. Both request/response and one-way type communications with the EIS are supported using the WebSphere Business Integration Adapter. Inbound communication from the EIS to the WebSphere Process Server, is enabled through the use of an export file that contains an inbound request interface. Both request/response and one-way notification type communications are supported using the WebSphere Business Integration Adapter.

Agenda

- Overview
- Architecture / Big-Picture
- **Development Tools**
- Administration
- Event Monitoring
- Logging and Tracing
- Summary

This section will cover development tools.

Development Tools

- Use WebSphere Business Integration Adapter Framework and System Manager tools to:
  - Create Business Objects
    - Business Object Designer
    - Object Discovery Agent (ODA), ODA Development Kit
  - Create Connector Configuration
    - Connector Configuration
  - Test
    - Visual Test Connector

The WebSphere Business Integration Adapter Framework Development Kit and System Manager tools continue to be used to create business objects, create connector configurations, and test the WebSphere Business Integration Adapter. These tools include the business object designer, object discovery agent, connector configurator, and visual test connector.

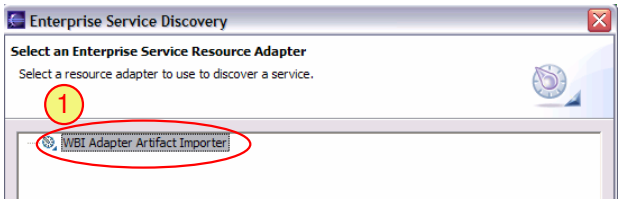
Development Tools

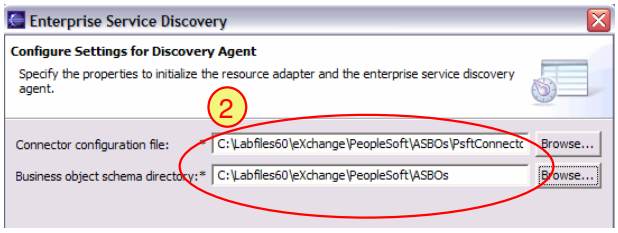
- Use WebSphere Integration Developer tools
  - Discover existing artifacts (Business Objects and Connector configuration)
  - Generate SCA artifacts such as import and export files, interface WSDLs, and convert Business Objects to WebSphere Process Server Business Object format
  - Configure WebSphere Process Server Service Integration Technologies messaging support
  - Assemble Service Component Architecture (SCA) solution

WebSphere Integration Developer tools are used to create the necessary artifacts for integration with the WebSphere Process Server. Using WebSphere Integration Developer and the enterprise service discovery wizard, discover existing WebSphere Business Integration business objects and connector configuration and generate the SCA artifacts such as import and export files, WSDL interfaces and business objects or business graphs in a format compatible with the WebSphere Process Server. Configuration of messaging support will be examined in detail. Once all the artifacts are created, you will assemble components into your integration solution, ultimately resulting in an Enterprise Application (EAR) file that can be exported and installed on the WebSphere Process Server.

Enterprise Service Discovery

- File > New > Enterprise Service Discovery
  - WBI Adapter Artifact Importer
- Configure Settings for Discovery Agent
  - Browse to existing connector configuration file
  - Browse to existing business object schema directory





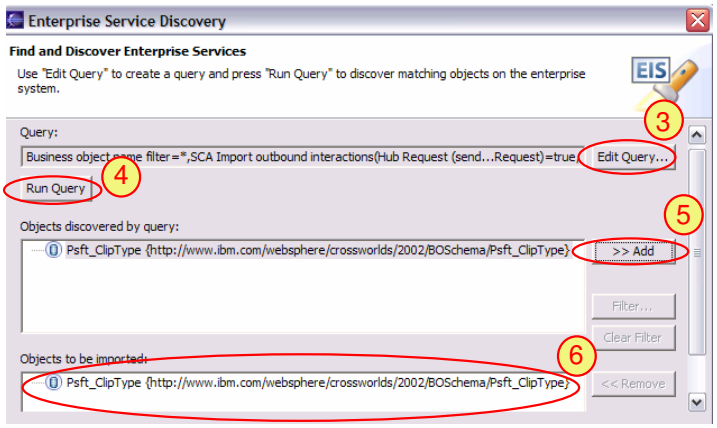
Enterprise Service Discovery can be initiated from the File > New > Enterprise Service Discovery menu item in the business integration perspective. Select WBI Adapter Artifact Importer from the panel and then specify the connector configuration file and business object schema directory. The configuration file is used by WebSphere Integration Developer to create a properties file that is included in the resulting EAR file. When the EAR file is installed to WebSphere Process Server, this properties file is used to automatically create and configure the JMS messaging support.

Enterprise Service Discovery

- Find and Discover Enterprise Services
  - Edit query to select the Business Objects and interaction patterns to support (filter)
  - Run Query, Select Meta data objects, Select Configuration Parameters

- Generates appropriate interfaces based on interaction selected

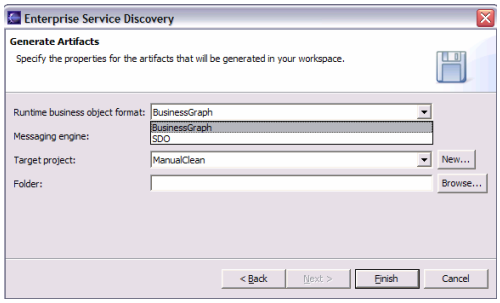
- Outbound (request response – one way)
- Inbound (request response – one way)



Once you have specified the configuration file and the location of business objects, the next step is to select which business objects you want to include in the application. The Edit Query button provides options to filter the resulting list of business objects discovered and the interaction mode in which they will be used; such as outbound or inbound and request/response, or one way. This determines the appropriate WSDL interface(s) that are created. Select the Run Query button to run the query, after which the objects discovered by the query are displayed. Select the objects wanted and click the >> Add button to add them to the Objects to be imported window.

Enterprise Service Discovery

- Specify Business Object format
  - BusinessGraph
    - Formatted in the WebSphere Process Server Business Graph format
  - SDO
    - Formatted in the WebSphere Process Server Business Object format



Use the Generate Artifacts panel to specify the required format, such as business graph or SDO. The business graph provides the verb, change summary, and event summary information and wrappers a business object. In many scenarios involving WebSphere Business Integration Adapters, the business graph will be the preferred format; however this will depend on the specific application requirements. Select the target project and Folder name and click finish to proceed.

Artifacts Generated

Artifact	Description
*.xsd	▪ Business Object xsd file
*BG.xsd	▪ Business Graph of the Supported Business Objects. These are the “migrated” versions of the original xsd files. (verb and objecteventID moved)
<ApplicationName>DeliveryHub.export	▪ SCA export file that specifies the export for Adapter Delivery to Hub communications.
<ApplicationName>RequestHub.export	▪ SCA export file that specifies the export for Adapter Request to Hub communications.
<ApplicationName>Agent.import	▪ This is the SCA import file that specifies the import for Hub to Adapter communications.
<ApplicationName>.wsdl	▪ The wsdl file identifies the operations and messages that are used. (Interface)
<ApplicationName>Wsadmin.wbia	▪ The file used as input to automatically configure MQClientLink and JMS Queues to allow communication with the server and the adapter

This table lists artifacts that are generated from the adapter creation process. The *.xsd and *BG.xsd (if you elected to generate a business graph) would be available in the business integration module. There are either exports or an import depending on the type of interaction pattern you selected on the Edit Query panel. The WSDL interface file identifies the operations and messages used. The <ApplicationName>Wsadmin.wbia properties file is used by the WebSphere Process Server for the automatic configuration of the Service Integration Technologies default JMS messaging configuration.

Configure JMS Communications

- Example of <ApplicationName>Wsadmin.wbia file
- Created from original connector.cfg
- Note highlighted properties set for WebSphere Process Server integrated messaging configuration
- WebSphere Process Server provides the QueueManager support

SiebelConnectorWsadmin.wbia

```
AdminInQueue=SIEBELCONNECTOR/ADMININQUEUE
AdminOutQueue=SIEBELCONNECTOR/ADMINOUTQUEUE
DeliveryQueue=SIEBELCONNECTOR/DELIVERYQUEUE
FaultQueue=SIEBELCONNECTOR/FAULTQUEUE
RequestQueue=SIEBELCONNECTOR/REQUESTQUEUE
ResponseQueue=SIEBELCONNECTOR/RESPONSEQUEUE
SynchronousRequestQueue=SIEBELCONNECTOR/SYNCHRONOUSREQUESTQUEUE
SynchronousResponseQueue=SIEBELCONNECTOR/SYNCHRONOUSRESPONSEQUEUE
AdminInJNDI=SIEBELCONNECTOR/ADMININQUEUE
AdminOutJNDI=SIEBELCONNECTOR/ADMINOUTQUEUE
DeliveryJNDI=SIEBELCONNECTOR/DELIVERYQUEUE
RequestJNDI=SIEBELCONNECTOR/REQUESTQUEUE
ResponseJNDI=SIEBELCONNECTOR/RESPONSEQUEUE
SyncRequestJNDI=SIEBELCONNECTOR/SYNCHRONOUSREQUESTQUEUE
SyncResponseJNDI=SIEBELCONNECTOR/SYNCHRONOUSRESPONSEQUEUE
AdminInJMS=SIEBELCONNECTOR.ADMININQUEUE
AdminOutJMS=SIEBELCONNECTOR.ADMINOUTQUEUE
DeliveryJMS=SIEBELCONNECTOR.DELIVERYQUEUE
RequestJMS=SIEBELCONNECTOR.REQUESTQUEUE
ResponseJMS=SIEBELCONNECTOR.RESPONSEQUEUE
SyncRequestJMS=SIEBELCONNECTOR.SYNCHRONOUSREQUESTQUEUE
SyncResponseJMS=SIEBELCONNECTOR.SYNCHRONOUSRESPONSEQUEUE
QueueManager=WBIA_QM
ChannelName=WBIA.JMS.SVRCONN
ClientLink=wbia.mqclientlink
SIBusName=<default>
DebugTrace=true
SaveAdminConfig=true
```


Shown here is an example of the <ApplicationName>Wsadmin.wbia file that is created from the existing connector configuration file information. Note the highlighted properties. To complete the JMS configuration for integration with WebSphere Process Server, you will need to make some minor modifications to the existing connector configuration file. The WebSphere Process Server messaging will now be providing the message broker or queue manager support for the WebSphere Business Integration Adapter. Highlighted here are the QueueManager name, ChannelName, ClientLink, and the SIBusName parameters.

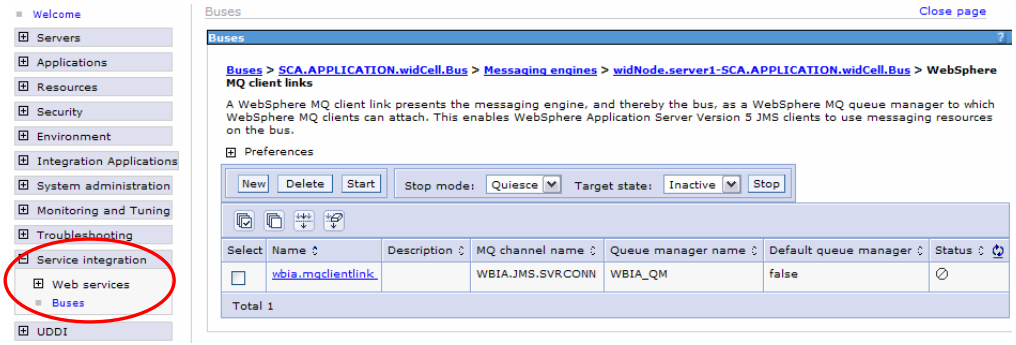
Configure JMS Communications

- Automatic configuration of MQClientLink and other WebSphere Process Server JMS communications artifacts
  - Configuration is automatically created at installation of Enterprise Application (EAR) containing WBI Adapter artifacts

Configuration of the messaging support is automatic and happens the first time an EAR file containing the <ApplicationName>Wsadmin.wbia file is installed on the WebSphere Process Server.

Configure JMS Communications

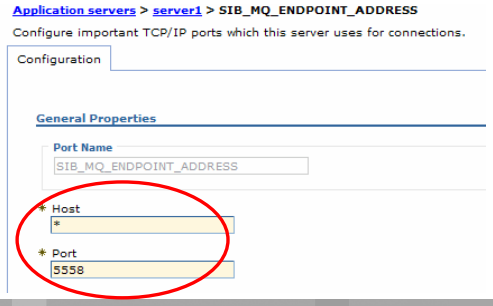
- WebSphere Administrative Console displays service integration buses



Shown here is an example of the Service Integration messaging support configured for use with WebSphere Business Integration Adapters, displayed in the WebSphere Process Server administrative console. To view this panel in the administrative console, expand the Service Integration options, and select the Buses option. Next, select the particular SCA Application messaging bus, messaging engines, SCA Application messaging engine, WebSphere MQ client links to view the wbia.mqclientlink object configured, or Queue points to view the queues created.

Configure JMS Communications

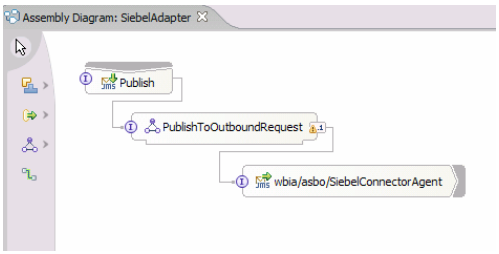
- Update existing connector.cfg file to match MQClientLink
  - BrokerType - WAS
  - jms.MessageBrokerName - WBIA_QM:WBIA.JMS.SVRCONN:localhost:5558
    - Assumes adapter and server are on the same machine and default port assignments were configured on the server during profile creation
  - jms.Password and jms.Username –
    - Used to authenticate for JMS communications if security is enabled



You will need to edit the existing connector configuration file that the WebSphere Business Integration Adapter is using and modify some of the parameters because the message broker is now the WebSphere Process Server messaging support. In the connector.cfg file, ensure the broker type parameter is set to WAS. You should also check the value for the jms.MessageBrokerName parameter and ensure that it is correct, including the Host name and Port number. The jms.Password and jms.Username should also be verified to ensure they are valid if security is enabled on the WebSphere Process Server.

Assemble WebSphere Business Integration Adapters into Solution

- Open module with Assembly Editor
- Wire to other Components
- Deploy SCA Application



You can use the Assembly Editor to assemble the WebSphere Business Integration Adapter into the business integration solution to complete your SCA application. Upon completion, you can continue with unit testing and ultimately export the application for deployment to WebSphere Process Server.

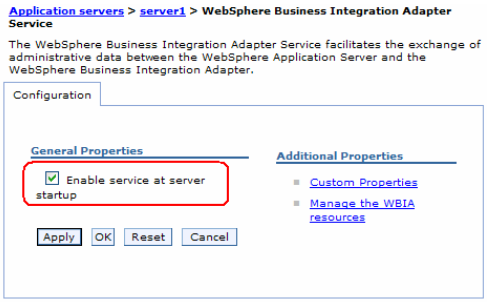
Agenda

- Overview
- Architecture / Big-Picture
- Development Tools
- **Administration**
- Event Monitoring
- Logging and Tracing
- Summary

This section will cover WBI adapter administration.

Administrative Operational Commands

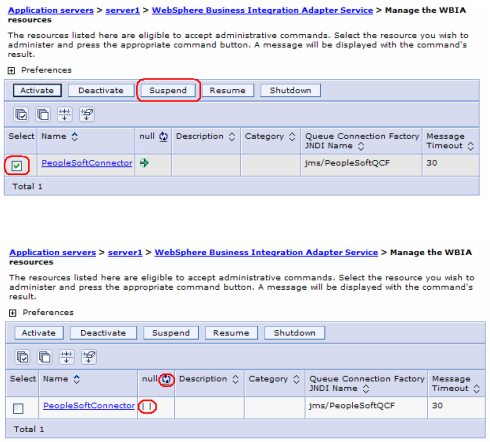
- Operational commands integrated into Administrative Console
- Enable WebSphere Business Integration Adapter Service
  - Servers > Application Servers > server1
- Create JMS Queue Connection Factory at Node level
  - Resources > JMS Providers > Default Messaging > JMS Queue Connection Factory > New
- Create WebSphere Business Integration Adapter Resource



Administration of the WebSphere Business Integration Adapters is available using the WebSphere Process Server administrative console. The WebSphere Business Integration Adapter service that provides support for this functionality is not enabled by default. If you'd like to use the integrated administrative console command support, you must first enable the service. To enable the service, first start the server and the administrative console application. From the administrative console, select Servers, Application Servers, and your specific server. Next, select the WebSphere Business Integration Adapter Service and check the box to enable service at server startup. Stop and restart the server for the change to take effect. Once the server is restarted, there are a few more configuration tasks to do. The WebSphere Process Server submits the operational commands to the WebSphere Business Integration Adapter via the JMS protocol, so you will need to create a JMS queue connection factory. From the administrative console, this is done by selecting Resources, JMS Providers, Default Messaging, JMS queue connection factory, and then New. Specify a name for the queue connection factory and complete the configuration information. Once you have created the JMS queue connection factory, create a WebSphere Business Integration adapter resource, which is also done under Resources, and give the adapter a name so the administrative console can route commands to it.

Administrative Console Operational Commands

- Manage the WebSphere Business Integration Adapter Resources
- Servers > Application Servers > server1 > WBIA Service > Manage WBIA resources
- Commands
  - Deactivate, Activate, Suspend, Resume, Shutdown
- WebSphere Business Integration System Manager tool, Adapter Monitor, may also be used to manage the adapters. It also allows management of the fault queue.



Once you have the service enabled and the configuration items taken care of and are ready to use the functionality, select the option to manage the WebSphere Business Integration adapter resources. Do this by clicking Servers, Application Servers, your specific application server name, WBIA Service, and then Manage WBIA Resources. An example is shown in the top right portion of this graphic. Select the resource that you created and the command you want to run to Activate, Deactivate, Suspend, Resume, or Shutdown the resource. You can also check the status of the adapter.

The commands available and their definitions are as follows:

Deactivate - Changes the status from active or paused to inactive (Connector process is active, however no processing in either direction)

Activate - Changes the status from inactive to active

Suspend - Changes the status from active to paused (no polling for events)

Resume - Changes the status from paused to active

Shutdown - Changes the status from active to “unavailable”. (Ends the active connector process). Note: After issuing a Shutdown, the adapter can only be restarted by invoking a script on the system that will be used to run it.

Adapter Monitor is a tool that ships with the WebSphere Business Integration Adapters and can also be used to manage the adapters. This tool allows for management of the fault queue in addition to the operational commands.

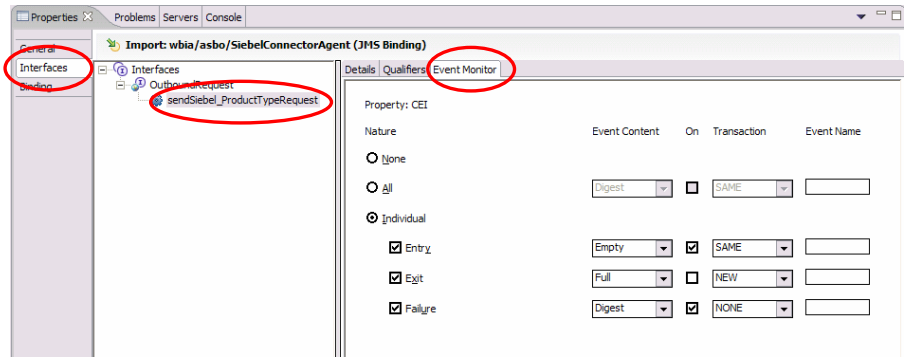
Agenda

- Overview
- Architecture / Big-Picture
- Development Tools
- Administration
- **Event Monitoring**
- Logging and Tracing
- Summary

This section will cover event monitoring.

Event Monitor: WebSphere Integration Developer

- Event Monitor is provided by the SCA JMS layer
  - Set in WebSphere Integration Developer - open SCA module in assembly editor, select properties pane, select Interface, operation, Event Monitor tab



Event monitoring is supported in integration scenarios involving WebSphere Business Integration Adapters. From the WebSphere Process Server perspective, event monitoring is provided by the SCA JMS layer support. Within WebSphere Integration Developer, you can enable event monitoring on the interface used with the adapter. Open the SCA module in the assembly editor select the import or export, from the properties pane, select interfaces, expand to the operation, and then select the type of event monitoring you want on the Event Monitor tab.

Event Monitor: Adapter Framework V2.6

- WebSphere Business Integration Adapter Framework V2.6 supports Common Event Infrastructure
  - Set in standard properties of connector configuration

6	BrokerType	WAS	String		Broker type
7	CharacterEncoding	ascii7	String		The connector agent will use the character encoding
8	CommonEventInfrastructure	false	Boolean		Specify if Common Event Infrastructure (CEI) should be invoked at runtime
9	ContainerManagedEvents		String		Indicates if container managed events are enabled
10	DeliveryQueue	SIEBELCONNECTOR/	String	QueueName	The logical queue that will be used by the connector to send business data
11	DeliveryTransport	JMS	String		The transport for data flowing between connector and server for subscrip

- Types of events
  - ADAPTER_START
  - ADAPTER_STOP
  - ADAPTER_TIMEOUT
  - REQUEST
  - DELIVERY

Beginning with the Adapter Framework V2.6, the adapters themselves also support event monitoring. Event monitoring is enabled using the options on the connector configuration panel. This is done in the adapter framework connector configurator tool, not in WebSphere Integration Developer. One of the properties in the connector configurator is the CommonEventInfrastructure, which you can enable by setting the value to true. Listed above are the event types that the adapter framework can emit.

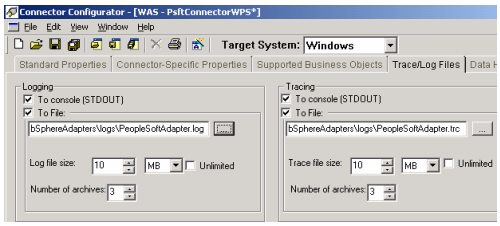
Agenda

- Overview
- Architecture / Big-Picture
- Development Tools
- Administration
- Event Monitoring
- **Logging and Tracing**
- Summary

This section will provide information on available logging and tracing.

Logging/Tracing

- Adapter log/trace files
  - Set in Connector Configurator
  - Log file and trace file
  - Configurable tracing level on Standard Properties tab
  - Configurable file names and locations on Trace/Log Files tab
  - Configurable file size and number of archives on Trace/Log Files tab
- Server log/trace files
  - Set in Administrative Console
  - Several files (trace, system out, native_stdout, native_stderr, activity)
  - Configurable file names and locations
  - Configurable trace string under Logging and Tracing -> servername -> Change Log Detail Levels



For assistance with problem determination, logging and tracing is available. Adapter logging and tracing can be set and configured from within the connector configuration file. WebSphere Process Server logging and tracing can be set and configured from within the server's administrative console.

Agenda

- Overview
- Architecture / Big-Picture
- Development Tools
- Administration
- Event Monitoring
- Logging and Tracing
- **Summary**

This section will provide a summary of topics covered in this presentation.

Summary

- WebSphere Process Server continues support of WebSphere Business Integration Adapters
- Use Adapter Framework toolset for WebSphere Business Adapter development and testing
- Use WebSphere Integration Development toolset for development and integration with WebSphere Process Server solutions
- Basic operational commands are integrated with WebSphere Process Server Administrative Console
- Event Monitoring is supported
- Logging and tracing are available

In summary, WebSphere Process Server continues support for the existing integration adapters. You will continue to use the existing adapter framework toolset for adapter development and testing. You will use WebSphere Integration Developer to integrate the adapters into the WebSphere Process Server and testing of business integration applications. Basic operational commands are integrated into the WebSphere Process Server administrative console. Event monitoring is supported from both the WebSphere Process Server SCA JMS layer, and also through the adapter framework. Logging and tracing can be configured to aid in problem determination analysis.

For More Information

- WebSphere Adapter Information Center

http://www-306.ibm.com/software/integration/wbiadapters/library/infocenter/

- WebSphere Process Integration Information Center

http://publib.boulder.ibm.com/infocenter/dmndhelp/v6rxmx/index.jsp

Shown here are links to information centers useful for reference material.

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	CICS	IMS	MQSeries	Tivoli
IBM(logo)	Cloudscape	Informix	OS/390	WebSphere
e(logo)business	DB2	iSeries	OS/400	xSeries
AIX	DB2 Universal Database	Lotus	pSeries	zSeries

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

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

Intel, ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

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

Linux is a registered trademark of Linus Torvalds.

Other company, product and 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 and/or changes in the product(s) and/or program(s) 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 (e.g., 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.