



IBM Software Group

## WebSphere® Message Broker Version 6.1

*Communicating with enterprise information systems using WebSphere Adapter nodes*



@business on demand.

© 2008 IBM Corporation  
Updated January 23, 2008

This presentation provides an overview of the WebSphere Adapter nodes in IBM WebSphere Message Broker version 6.1.

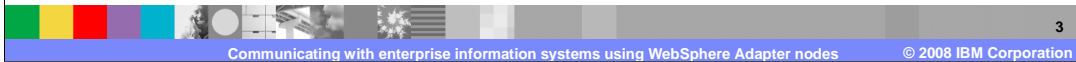
## Agenda

- Introduction
- Scenarios
- Configuring Message Broker for use with enterprise information systems
  - ▶ Discovery of business object information
  - ▶ Development of message broker artifacts
  - ▶ Setting up the broker runtime
  - ▶ Deployment and administration
  - ▶ Problem determination

This presentation covers a general overview of the adapters. It starts with a recap of the previously available adapters and then introduces the new adapters. The presentation also reviews some of the real time scenarios where adapters can be used, and the configuration of adapters. At the end, there will be an overview of problem determination.

## Introduction

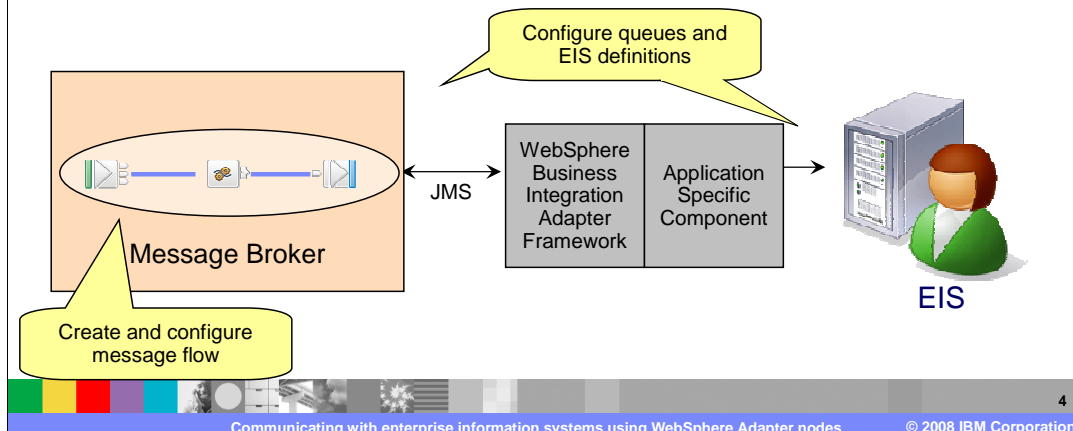
- WebSphere Message Broker V6.1 supports bi-directional interaction with major enterprise information systems:
  - ▶ SAP
  - ▶ Siebel
  - ▶ PeopleSoft
- Based on WebSphere Adapters technology



With V6.1, WebSphere Message Broker supports bi-directional interaction with SAP, Siebel and PeopleSoft Enterprise Information Systems using the WebSphere Adapters Technology.

## WebSphere Business Integration adapters

- Stand-alone product that enables WebSphere Message Broker V6.0 and earlier to work with enterprise information systems (EIS)
- Need to manage two sets of EIS definitions

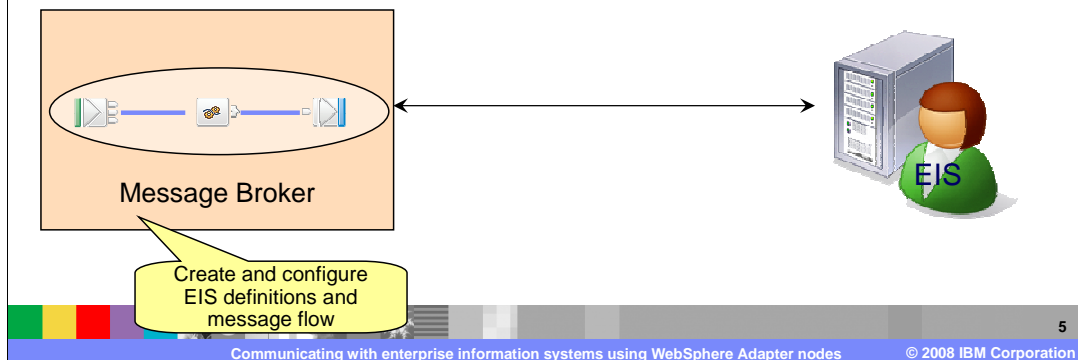


IBM WebSphere Business Integration adapters are a suite of ready-to-use adapters for interfacing to enterprise information systems, or EISs. These adapters are based on a common framework. This allows you to configure and operate the adapters in a consistent manner, while using them in different deployment configurations within the WebSphere software platform to meet your business integration needs.

These legacy adapters are not integrated directly into the Message Broker environment. They are stand-alone adapters running outside of the broker runtime. You use the WebSphere Message Broker Toolkit to design and develop your message flows and use the JMS binding to communicate with the stand-alone WebSphere Business Integration adapters. This means that you have to configure and manage two integration technologies.

## WebSphere Adapters

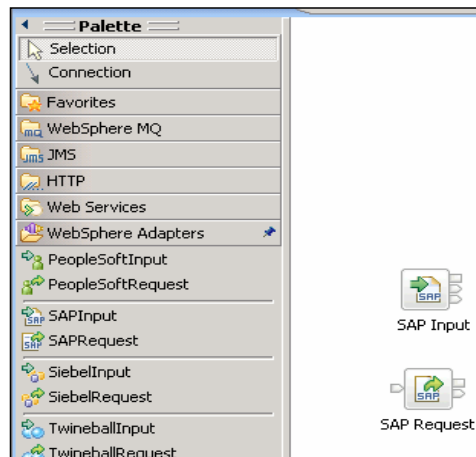
- Integrated set of libraries that enables EIS connectivity from within WebSphere Process Server and WebSphere Enterprise Service Bus
  - ▶ ...and now, WebSphere Message Broker V6.1
- Based on the JCA Framework 1.5



WebSphere adapters are enhanced to make development, deployment, and integration of adapters simple. Enhancements include the ability to discover EIS objects and services, ensure “assured once” inbound delivery of events and other capabilities that are discussed in this presentation. The adapter uses common foundation classes across each of the adapters. These classes enhance the JCA 1.5 functions and encapsulate many common adapter functions to simplify development tasks.

Adapters are exposed as nodes and can be configured directly using the WebSphere Message Broker Toolkit. The adapter nodes used in your message flow run in the broker runtime and can be administered directly using the broker tools.

## Implementation



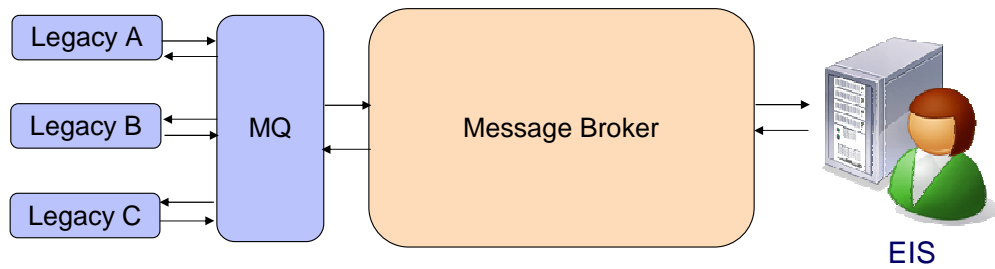
- A set of input and request nodes
  - ▶ SAPInput, SAPRequest
  - ▶ SiebelInput, SiebelRequest
  - ▶ PeopleSoftInput, PeopleSoftRequest
  - ▶ TwineballInput, TwineballRequest
- A wizard to discover metadata information from these systems and generate message definitions and other artifacts

This slide shows a screen capture from the WebSphere Message Broker Toolkit. The Adapters are available on the palette in the message flow editor. Use a Request node when you want to make function calls from your message flow to the enterprise information system. Input nodes are used when you are listening for any events that happen on the EIS. The Adapter retrieves the event information, and sends the event to the rest of the message flow. The Twineball adapter is a sample provided with the toolkit. WebSphere Message Broker Toolkit provides an adapter connection wizard to configure the adapters; the adapter connection wizard is discussed later on in the presentation

## Scenarios [1]

### Creating a legacy interface to an EIS

- A new EIS needs to be integrated into an existing infrastructure.



The next couple of slides review the scenarios where you can use the adapter nodes to connect to the supported enterprise information systems. In this first scenario, an internal infrastructure communicates between different components using COBOL copy book structures over WebSphere MQ. A new EIS needs to be integrated into this infrastructure. With the support in V6.1 to configure and use the adapter nodes in the toolkit, you can use the adapter nodes in your message flow to integrate with the new EIS

## Scenarios [2]

### Creating a Web services interface to an EIS



8

Communicating with enterprise information systems using WebSphere Adapter nodes

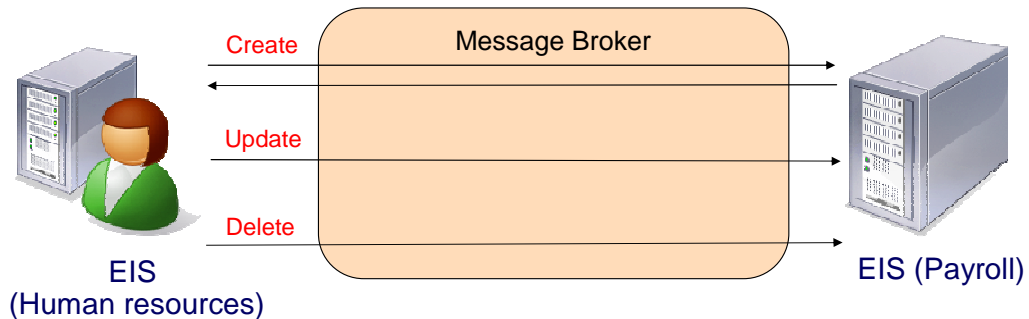
© 2008 IBM Corporation

Another scenario is that you want to expose capability in your EIS through a Web services interface. By using the HTTP and SOAP nodes in conjunction with the EIS Adapter nodes, it is straightforward to enable this.



## Scenarios [3]

### Synchronizing EISs



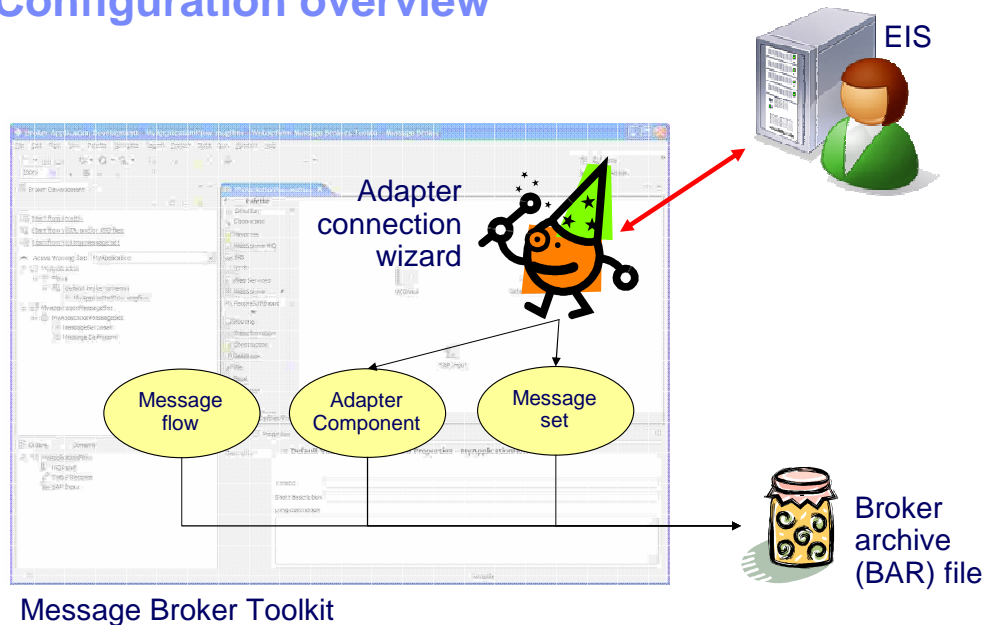
9

Communicating with enterprise information systems using WebSphere Adapter nodes

© 2008 IBM Corporation

The scenario shown in this slide requires ensuring that human resources and payroll systems contain identical information. For example, creating a new employee record on the human resources system should automatically update the payroll system with the same details. The adapter request node can be used to create an entry in human resources system which can result in an event that is monitored and processed by Adapter input nodes to create the employee entry in the payroll system.

## Configuration overview



Message Broker Toolkit

10

Communicating with enterprise information systems using WebSphere Adapter nodes

© 2008 IBM Corporation

This slide depicts the steps involved in configuring the adapter nodes and creating a deployable application. WebSphere Message Broker Toolkit provides an adapter connection wizard that connects to the EIS, introspects and imports the metadata information and creates the necessary artifacts for you to build your message flow. The artifacts generated by the wizard are the message set projects that contain all the message definitions for the metadata selected for import and the adapter component file. The adapter component created is different based on the interaction style you choose when configuring using the wizard. The supported interaction styles are inbound and outbound. Once the artifacts are created for you, you can use these generated artifacts to configure the adapter nodes and include them in the message flow based on your application logic. The finished message flow project along with adapter component and the message set project become the components of the bar file that you create to deploy to the broker runtime.

## Configuration steps

- **Discovery:** Discovering EIS metadata and automatic creation of components to access the EIS
- **Development:** Creating message flows which make use of the discovered components
- **Enablement:** Configuring the broker with the location of the EIS provider jars/native libraries
- **Administration:** Deploy all required components to Message Broker, and administer them



11

Communicating with enterprise information systems using WebSphere Adapter nodes

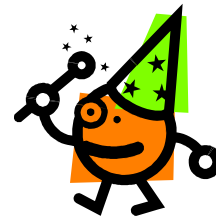
© 2008 IBM Corporation

This slide summarizes the steps involved in using adapter nodes as part of your message flow and how to administer them. At a high level the interaction steps can be broken down into four steps. The first step is the Discovery phase where you use the adapter connection wizard to configure the adapter and generate the necessary artifacts. The next step is the development phase where you use the generated artifacts to create the adapter node and use it as part of your message flow. The third step is the Enablement step where you specify the necessary dependency files required at runtime by the adapter. For example the SAP adapter uses the sapjco.jar file to communicate with the SAP system. The last step is to create the Broker Archive (BAR) file, deploy it to the execution group and administer the application.

## Discovery

Discovery  
Development  
Enablement  
Administration

- Discovery and import
  - ▶ Wizard to connect to the EIS and discover object metadata and create all required message sets and connection information
- Generates:
  - ▶ A message set definition (DataObject domain)
  - ▶ Adapter component
- High-level steps
  - ▶ Select Interaction style
  - ▶ Connect to EIS
  - ▶ Select services
  - ▶ Select business objects
  - ▶ Select operations
  - ▶ Refine connection properties for broker runtime
  - ▶ Generate message set



12

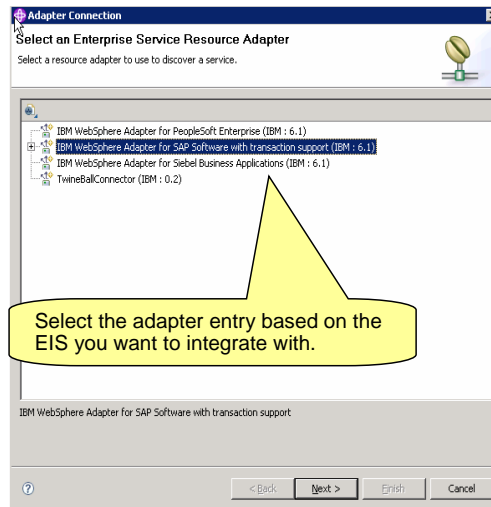
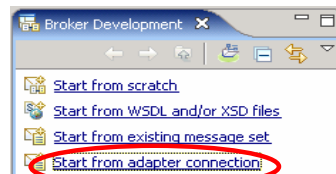
Communicating with enterprise information systems using WebSphere Adapter nodes

© 2008 IBM Corporation

The next few slides will give you a brief overview of the steps involved in the first step which is the Discovery. The adapter connection wizard lets you introspect the metadata information on the EIS and select the objects or services you want to work with. The wizard will then create an adapter component file and message set containing the message definitions based on the selections you make. The high level steps involved in the discovery and import process when using the wizard are listed on this slide.

## Discovery steps 1 - Launch wizard

- Adapter Connection Wizard can be invoked from
  - ▶ From Menu option
    - File → New → Adapter Connection
  - ▶ From Broker Development View
    - Right click, New → Adapter Connection
  - ▶ Using Quick start wizard



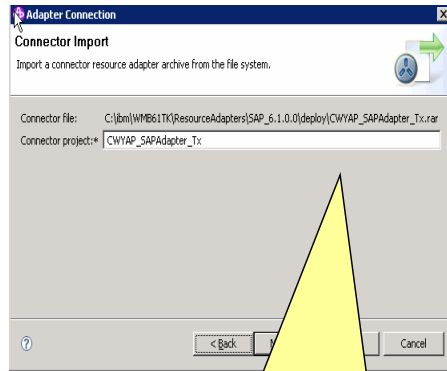
13

You can launch the adapter connection wizard from several different locations in the toolkit. The first option is to invoke from the menu by selecting the File → New → Adapter Connection. The second one is to right click in the Broker Development view and select New → Adapter Connection. The third option is to invoke from the quick start link “Start from Adapter Connection” in the Broker Development view.

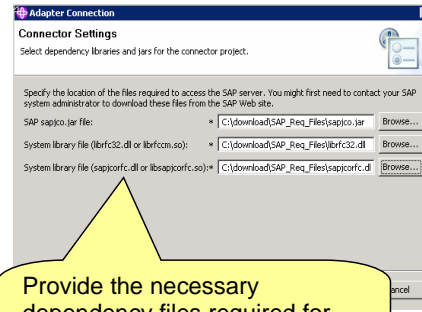
Any of the above three options will launch the wizard shown in the slide. Based on your requirements for the application and which backend system you are connecting to, you can select the adapter entry from the list and click next. The presentation does not cover all the panels in the wizard but discusses the important ones.

## Discovery steps 2 – Provide dependencies

- “Application” refers to Message Broker



Provide the connector module name or accept default.

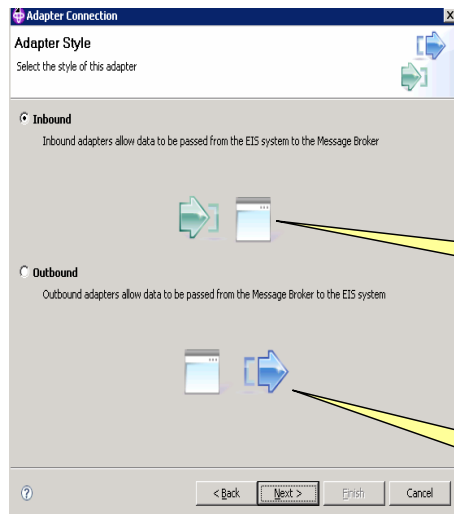


Provide the necessary dependency files required for the tool to make the connection to the backend system.

In the “Connector Import” panel, you can either provide a name or accept the default value and continue to the next panel by clicking next. The wizard will import the resource archive (RAR) file and will generate a connector module using the name specified in the previous panel. In the Connector Settings panel, provide the path to the dependent files required by the tool to make the connection to the Enterprise Information System. The screen capture uses SAP adapter as an example.

## Discovery steps 2 - Select “Adapter Style”

- “Application” refers to Message Broker



Inbound - An event from The EIS which is sent to Message Broker.

Outbound - A request sent from Message Broker to the EIS

In the adapter style selection screen, you can choose the type of interaction style. It can be inbound or outbound. Inbound corresponds to an event that occurs on the EIS which is sent to the Message Broker. Outbound corresponds to the request made from an adapter node from the Message Broker application to the EIS.

## Discovery steps 3 - Configure settings

- Provide properties to connect to EIS
  - ▶ Authentication properties like username and password.
  - ▶ EIS specific properties like host name, port based on the EIS
  - ▶ Interaction style and Interface selection

16

Once you select the interaction style the next screen prompts you to provide the necessary properties required for the adapter connection tool to access the EIS metadata. The properties that you normally see in this panel are the username, password and the EIS specific properties like the host name, port. You also have an option to modify the interaction style and select the appropriate interface supported for the interaction style. For example, SAP supports Business Application Programming Interface (BAPI), and Application Link Enabling(ALE) interfaces for outbound. At the bottom of the screen, you can see the attribute for log file location which can be edited. The log file is used to log any events related to discovery and import of metadata using the Adapter Connection wizard.



## Discovery steps 4 – Filter services

- Discover and filter the EIS data

Use this button to specify the filter criteria

Specify the search string and the numbers of results to display

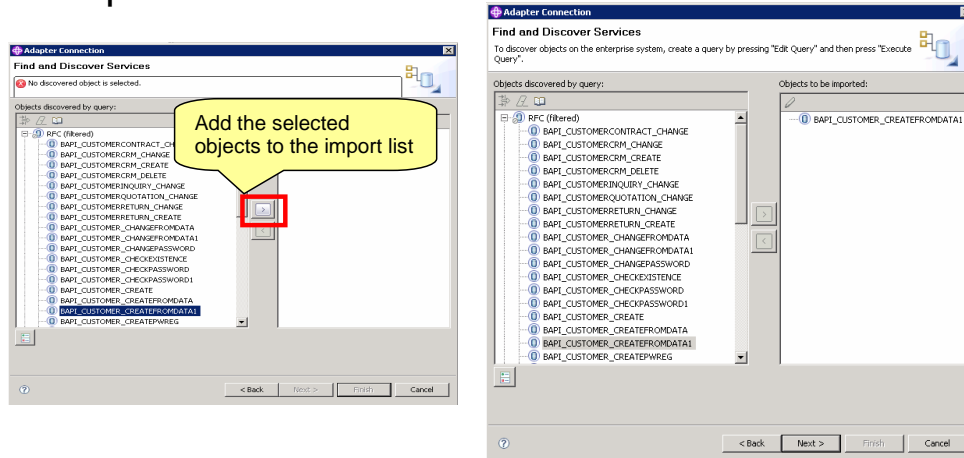
17

Communicating with enterprise information systems using WebSphere Adapter nodes © 2008 IBM Corporation

In the Find and Discover Services panel, you can specify your filter criteria to limit the search results when querying the metadata from the Enterprise Information System.

## Discovery steps 5 - Select objects for import

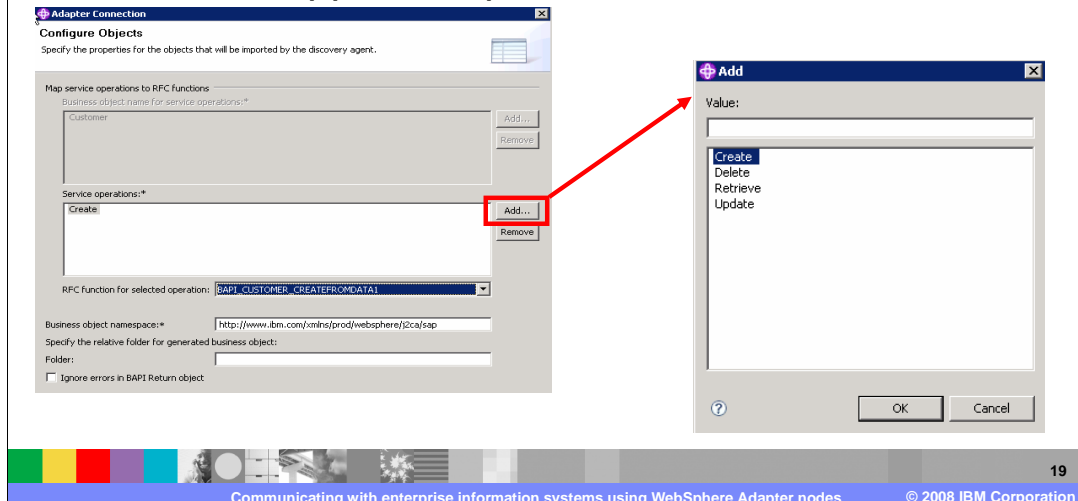
- Select the EIS object from the filtered list and add to import list



From the filtered list, you can select the objects or services that you want to work with and add them to the imported list. Based on the EIS you are working with, the objects and services supported by that adapter vary. The example used for this presentation is using the Business Application Programming Interface (BAPI) calls.

## Discovery steps 6 – Select operations

- Provide the name and location for the objects to be imported
- Select the supported operations



Once you finish selecting the objects or services you want to work with, the next step is to associate the objects with the operations supported by the adapter. In the example shown here, the BAPI call selected is related to customer creation, so you select the create method from the list of supported operations. This means that when you invoke the create operation for the adapter, BAPI\_CUSTOMER\_CREATEFROMDATA1 is called on the SAP system. Depending on the adapter you are working with, different operations are supported.

## Discovery steps 7 - Runtime connection properties

Adapter Connection

**Publishing Object Configuration Properties**

Warning: Sensitive values, such as passwords, should not be saved.

Service operations  
If you want to modify the names, or add a description to the operations to be generated in the interface file, press the "Edit Operations" button.

Deployment properties  
Specify the connection properties which will be used to connect to the Enterprise Information System at runtime:

Connection properties

SAP system connection information

Host name: 9.26.248.125

System number: 00

Client: 812

Language code: EN (English) Select...

Code page: 1100

User name: LUONGJ

Password: \*\*\*\*\*

Advanced >>

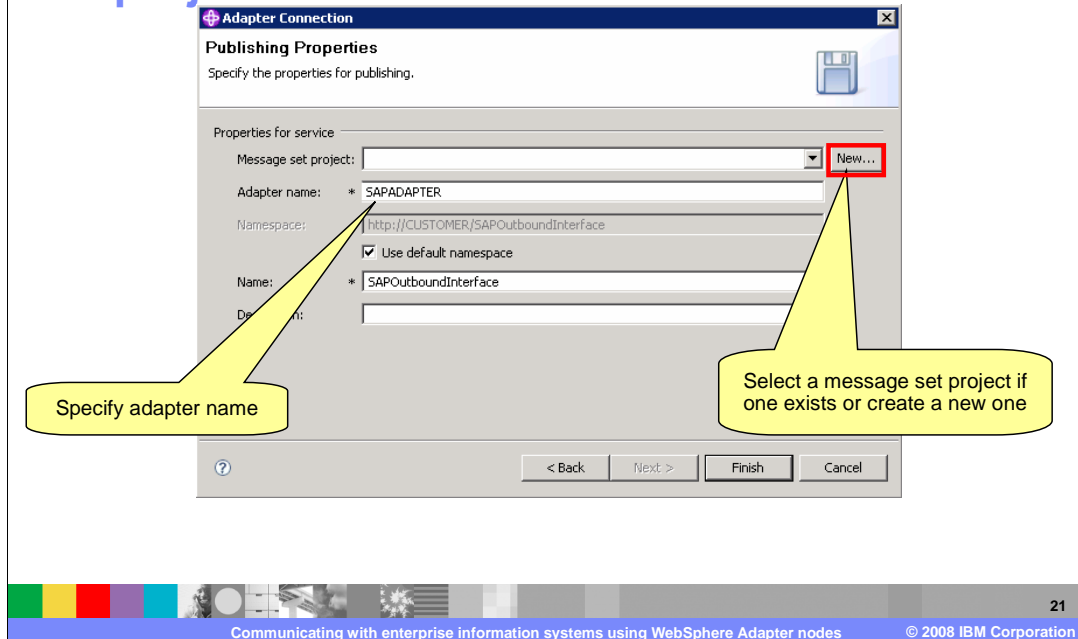
< Back Next > Finish Cancel

20

Communicating with enterprise information systems using WebSphere Adapter nodes © 2008 IBM Corporation

In the Publishing Object Configuration Properties panel, you provide the configuration properties used at runtime by the adapter. The initial screen where you provided some of the same values you see on the panel shown here are used by the adapter connection wizard to connect to the EIS and do the discovery and import of metadata

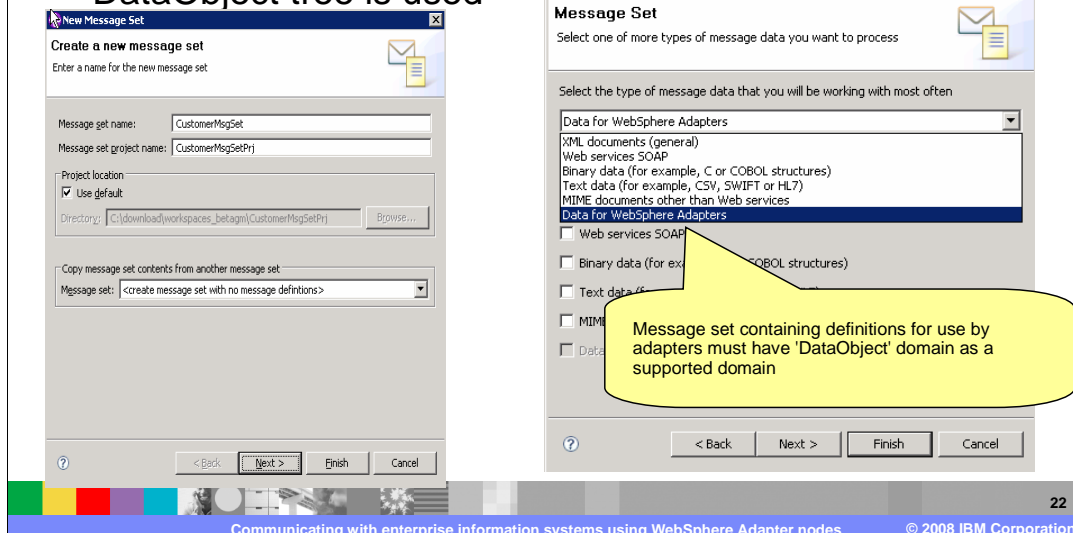
## Discovery steps 8 - Select (or create) message set project



The last task in the wizard is to create the message set project that contains the message definitions and provides the name for the adapter component that is to be created. If you already have an existing message set project created in your workspace, you can use that or you can create a new message set project.

## Discovery steps 9 - New message set

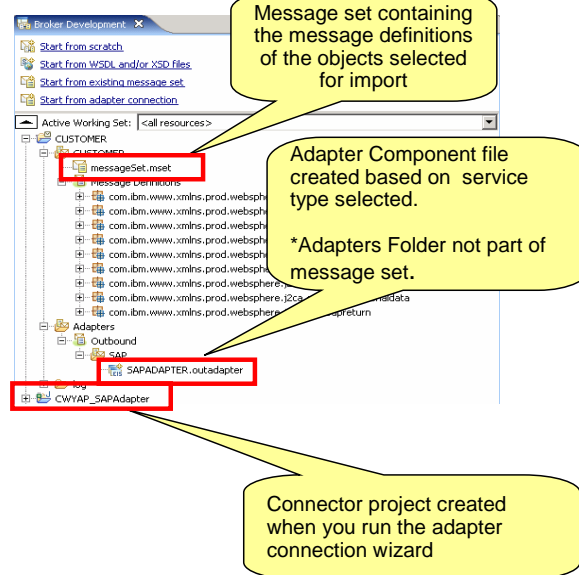
- DataObject domain - new domain for adapters
  - ▶ When used with adapter input or request nodes, DataObject tree is used



In the “Create a new message set” panel, provide the name of message set and message set project and continue to the next panel by clicking Next. In the next panel, by default, Data for WebSphere Adapters is selected for message data. When working with SAP, Siebel and PeopleSoft adapters, accept the default value of “Data for WebSphere Adapters”. This is the required parser when using the WebSphere Adapter nodes.

## Discovery steps – End result

- **Generated Artifacts**
  - ▶ **Adapter Component**
    - .inadapter file for inbound adapter style
    - .outadapter file for outbound interaction style
  - ▶ **Message set project**
    - Message set
    - Adapters folder – not part of message set
- **Runtime connection properties can be edited using the Adapter editor**



Once you complete the discovery process using the adapter connection wizard, you should see the message set project which contains the message definitions and a folder called Adapters. The Adapters folder (if one does not exist) is created under the message set project, at the same level as the message set folder. A file with an extension of '.inadapter' or '.outadapter' is created by the adapter connection wizard. The '.inadapter' extension is used for the adapter component when you choose the adapter style as inbound; '.outadapter' is used when the Adapter style is outbound. The '.inadapter' and '.outadapter' files are placed in the Inbound or Outbound adapter type folder. You can edit the connection properties used by the adapter at runtime by opening the adapter component file using the WebSphere Adapters editor.

## Discovery steps – Message example

SapBapiCustomerCreatefromdata1.mxsd			
Structure	Type	Min Occurs	Max Occurs
[-] SapBapiCustomerCreatefromdata1.mxsd			
[-] Messages			
[-] SapBapiCustomerCreatefromdata1	SapBapiCustomerCreatefromdata1		
[-] AllowConsumerMaintenance		0	1
[-] UpdateCreditControlData		0	1
[-] NewCustomerNumber		1	1
[-] SapPiCompanydata	sappicompanydata:SapPiCompanydata	0	1
[-] SapPiCopyreference	sappicopyreference:SapPiCopyrefer...	1	1
[-] SapPiOptCompanydata	sappioptcompanydata:SapPiOptCom...	0	1
[-] TransportationZoneToOrFromWhichTheG		0	1
[-] ReconciliationAccountInGeneralLedger		0	1
[-] TermsOfPaymentKey		0	1
[-] ShippingConditions		0	1
[-] DeliveringPlant		0	1
[-] PartialDeliveryAtItemLevel		0	1
[-] CreditControlArea		0	1
[-] TaxJurisdiction		0	1
[-] SapPiOptPersonaldata	sappioptpersonaldata:SapPiOptPers...	0	1
[-] TransportationZoneToOrFromWhichTheG		0	1
[-] ReconciliationAccountInGeneralLedger		0	1
[-] TermsOfPaymentKey		0	1
[-] ShippingConditions		0	1
[-] DeliveringPlant		0	1
[-] PartialDeliveryAtItemLevel		0	1
[-] CreditControlArea		0	1
[-] TaxJurisdiction		0	1
[-] SapPiPersonaldata	sappipersonaldata:SapPiPersonaldata	1	1
[-] SapReturn	sapreturn:SapReturn	1	1

24

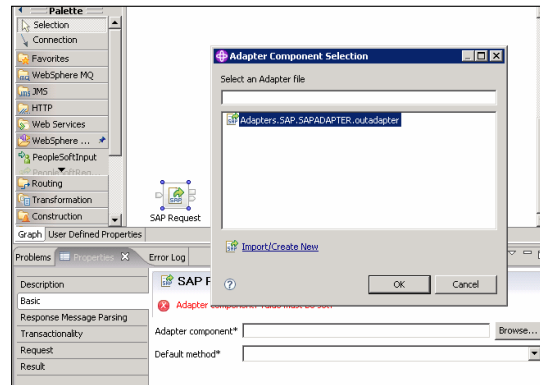
Shown on this slide is the message definition created for the BAPI call that is imported during the discovery process. The definition file describes the structure of the message and contains the application specific information required for processing the message by the adapter nodes.



## Development – Adapter nodes

Discovery  
Development  
Enablement  
Administration

- The next task is to create the message flows that use the EIS nodes
- Set of six nodes to cover EIS inbound and outbound scenarios
- Relatively few properties to configure
  - ▶ Just associate node with the generated adapter component and message set project
- Drag and drop of adapter component onto message flow editor...



25

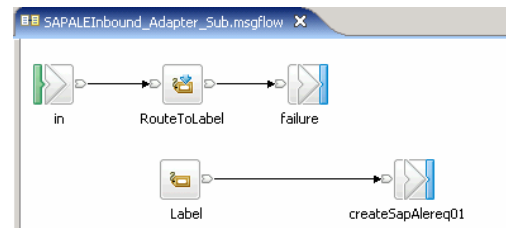
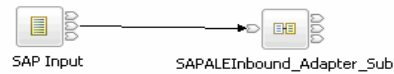
Communicating with enterprise information systems using WebSphere Adapter nodes

© 2008 IBM Corporation

The next logical step after the discovery process is the development phase. You can use the generated artifacts from the Adapter Connection wizard and associate them with the adapter nodes. Once the artifacts are associated with the adapter nodes, you can wire them with the other nodes in your message flow to complete your application.

## Development – Inbound message flows

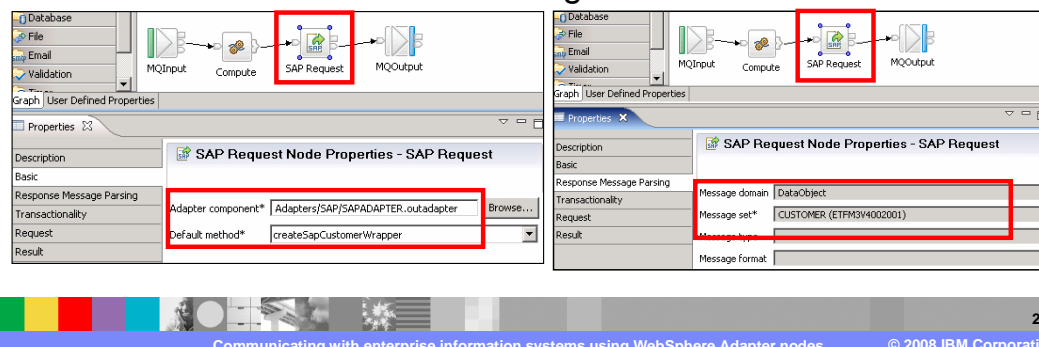
- Drag and drop adapter inadapter component file
  - ▶ Adapter input node is created (for example SAP Input node)
    - Adapter component associated with the input node
    - Destination list set
- A sub flow is created that exposes each operation that was selected during the creation of the adapter definition



Based on the interaction style you choose when configuring the adapter, the adapter connection wizard will create an .inadapter or .outadapter component file. For inbound interaction style the .inadapter component file is created. You can create a message flow project and then drag the .inadapter component from the generated artifacts in the Broker Development view to the message flow editor. This results in the creation of an input node corresponding to the EIS adapter. The input node is wired to a sub flow node. The sub flow node contains a route to a label node, which forwards the incoming message to the appropriate label node. The number of label nodes depends on the number of operations you choose when configuring the adapter. The message set that is created when configuring the adapter is automatically associated with the input node, and the parser is set to DataObject.

## Development – Outbound message flows

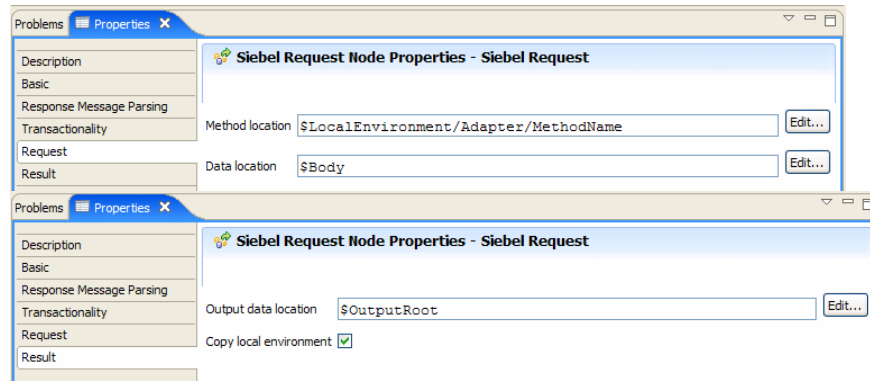
- Drag and drop adapter outbound component file
  - ▶ Adapter request node is created (for example: SAP request node)
  - ▶ Adapter component associated with the request node
  - ▶ Message set associated automatically
  - ▶ Message domain set to DataObject
- Connect to nodes in the message flow



When you choose the interaction style as outbound, an .outadapter component is created for you by the adapter connection wizard. You can drag this .outadapter component file onto the message flow editor which creates a request node relevant to the adapter type. The message set is automatically associated to the request node, which can be viewed in the properties of the node. The message domain is set to DataObject, which is the required domain for adapter nodes. You can then wire this request node appropriately as part of your message flow to complete your application.

## Development – Node properties

- For request nodes, the EIS method to invoke and the location of the output data location can be configured



- For input nodes, *\$LocalEnvironment/Adapter/MethodName* is set to the name of the business method corresponding to the EIS event that triggered the message delivery

The request node has a default method set when you create it. You can change the default method either programmatically by setting the Local Environment variable, or change it in the properties view by setting the default method to the one you want. For the Input nodes, the *\$LocalEnvironment/Adapter/MethodName* variable is set to the name of the business method corresponding to the enterprise information system event that triggered the message delivery.

## Setting up the broker runtime

Discovery  
Development  
**Enablement**  
Administration

- Your EIS provider provides client libraries that the broker requires in order to connect to the EIS
- Using configurable services is the recommended approach for adding the jar files required to access the EIS
  - ▶ This ensures that no other part of message broker has access to the client (and potentially cause conflicts)
- Use “mqsichangeproperties” to set up and report the properties, for example:
  - ▶ To set up the sapjco.jar file required to connect to SAP system

```
mqsichangeproperties B1 -c EISProviders -o SAP  
-n jarsURL -v <jarfile_location>
```

- ▶ To set up native library files required for SAP

```
mqsichangeproperties B1 -c EISProviders -o SAP  
-n nativeLibs -v <nativelib_location>
```

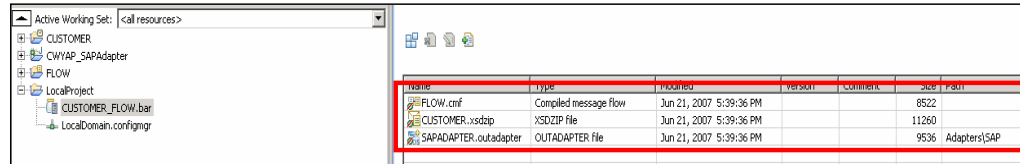
- ▶ To report the properties

```
mqsireportproperties B1 -c EISProviders -o SAP -r
```

The next logical step is the enablement step. 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 examples of the commands for setting SAP dependencies.

## Administration - Create BAR file

Discovery  
Development  
Enablement  
**Administration**



Artifact	Type	Created	Size (KB)	Checksum	Source Path
FLOW.cmf	Compiled message flow	Jun 21, 2007 5:39:36 PM	8522		
CUSTOMER.xsdzip	XSDZIP File	Jun 21, 2007 5:39:36 PM	11260		
SAPADAPTER.outadapter	OUTADAPTER file	Jun 21, 2007 5:39:36 PM	9536		Adapters[SAP

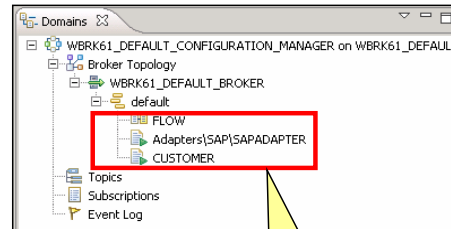
- BAR file contents
  - ▶ Message flow
  - ▶ .xsdzip file
  - ▶ Adapter component
  
- Can deploy adapter component and message set separately from message flow

30

The last step is the administration where you create the broker archive (BAR) file and deploy it to the execution group. You create the BAR file by switching to the Broker Administration view and adding the message set project, adapter component and the message flow. The message set is stored as a file with the extension .xsdzip. You can deploy the message set and the adapter component separately from the message flow.

## Administration – Managing resources

- Can associate only one input node to an .inadapter component file.
- No restrictions on usage of outbound adapter component usage
- The adapter component cannot be removed if a flow using it is running

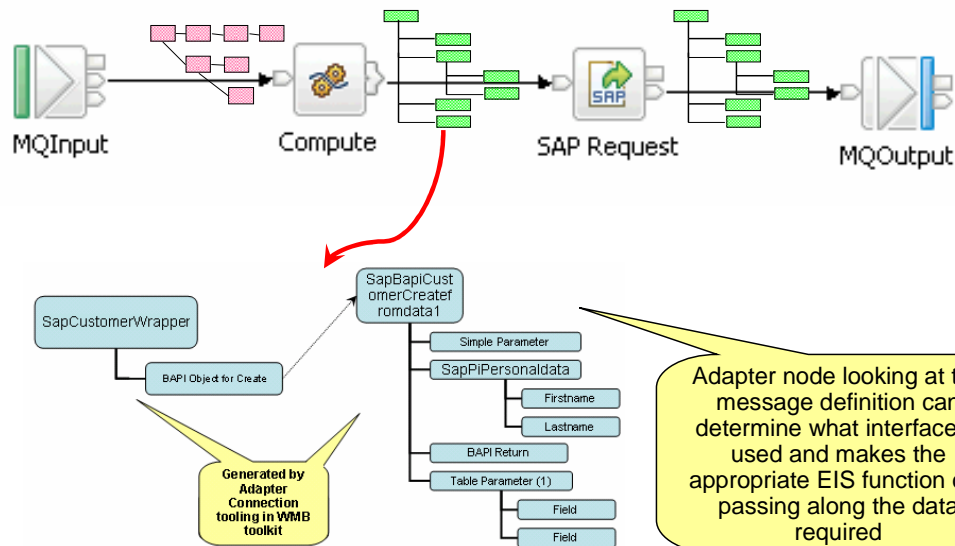


Deployed artifacts in the execution group.

- Message flow
- Message set
- Adapter component

You cannot associate more than one input node to the same .inadapter component. This is to avoid the contention for the events between several input nodes. No such restriction is applicable for .outadapter component files. The .outadapter component can be associated with several request nodes.

## Message flow details



32

Communicating with enterprise information systems using WebSphere Adapter nodes

© 2008 IBM Corporation

The slide summarizes the flow of message in a message flow and how the adapter node processes the message. This example shows an outbound scenario where you are making a call to the backend system. The SAP adapter is configured to make a BAPI call when you invoke the create operation. The message comes into the flow through a queue and is sent to the compute node. In the compute node you retrieve the information in the incoming message and populate the data into a message which conforms to the message definition created for the BAPI call. The message definition is already created by the adapter connection wizard when you configured the adapter and selected to work with the specific BAPI. Once the message is received by the SAP request node, the adapter will retrieve the data and application specific information from the message. It will determine what interface to use and what function call to make on SAP. The adapter then makes the function call to SAP and will send the appropriate attribute information for the BAPI function call. The returned information is then sent to the MQOutput node.



## Administration - Log and trace files

- Logs
  - ▶ Local error log
    - Windows®: Event log
    - UNIX® and Linux®: 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
    - Example: `mqsichangetrace WBRK61_DEFAULT_BROKER -u -e executiongroup -l debug -c 5000 -r`
  - ▶ Service trace
    - Example: `mqsichangetrace WBRK61_DEFAULT_BROKER -e executiongroup -tl debug -rc 100000`

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

## Summary

- Message Broker V6.1 supports bi-directional interaction with major EISs:
  - ▶ SAP
  - ▶ Siebel
  - ▶ PeopleSoft
- Adapter connection wizard to configure adapter connection and message set definitions
- Nodes to make use of EIS connections from within flows

In summary, WebSphere Message Broker V6.1 provides support for bi-directional interaction with the SAP, Siebel and PeopleSoft enterprise information systems. You can configure the adapter by making use of the adapter connection wizard provided in the WebSphere Message Broker toolkit.

## Feedback

### Your feedback is valuable

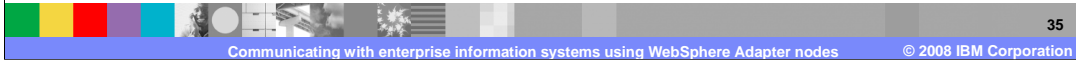
You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- 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\\_AdapterNodes\\_Overview.ppt](mailto:iea@us.ibm.com?subject=Feedback_about_WMBV61_AdapterNodes_Overview.ppt)

This module is also available in PDF format at: [../WMBV61\\_AdapterNodes\\_Overview.pdf](..\\WMBV61_AdapterNodes_Overview.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

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

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

UNIX is a trademark of The Open Group in the United States, other countries, or both.

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.

