



IBM Software Group

**WebSphere® Process Server V6.0**  
**WebSphere® Integration Developer V6.0**  
**WebSphere® Adapters V6.0**

***WebSphere Adapters – Common Details***



@business on demand.

© 2005 IBM Corporation  
Updated November 2, 2005

This presentation covers some of the common details that apply to all the WebSphere Adapters V6.0.

## Agenda

- General Overview of IBM WebSphere Adapters
- Interaction Styles
- Event Management for Inbound Operations
- Business Objects
- Generic Configuration for all Adapters
- Common Problem Determination
- Summary



The agenda of this presentation is to provide an overview of adapters, and discuss interaction styles, event management for inbound operations, business objects, generic adapter configuration, and problem determination.

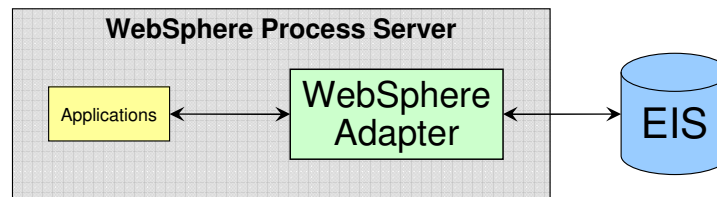
## Section

# ***General Overview of WebSphere Adapters***

This section will provide a general overview of WebSphere Adapters.

## Overview - Recap

- New IBM WebSphere Adapters
  - ▶ IBM WebSphere Adapter For Flat Files V6.0
  - ▶ IBM WebSphere Adapter for JDBC V6.0
  - ▶ IBM WebSphere Adapter for SAP Applications V6.0
  - ▶ IBM WebSphere Adapter for PeopleSoft Enterprise V6.0
  - ▶ IBM WebSphere Adapter for Siebel Business Applications V6.0
- J2EE™ Connector Architecture (JCA 1.5) compliant
- Integrated with WebSphere Process Server V6.0
- Uses Adapter Foundation Classes that enhances JCA 1.5 and encapsulates many common adapter functions



The WebSphere Process Integration product contains several new enhanced WebSphere Adapters based on the JCA 1.5 specification and enhanced to make development and deployment of adapters simple within an SOA application. These adapters run within applications in WebSphere Process Server V6

The adapter uses common foundation classes that enhances the JCA 1.5 functions and encapsulates many common adapter functions to simplify the adapter developer tasks.

JCA 1.5 specification provides open J2EE standard for EIS connectivity, providing bi-Directional interaction with EIS in a managed framework that provides Qualities of Service thru J2EE Application Server, like Connection life-cycle management, Connection pooling, Security, and Transactions.

In this release, there are a total of 5 WebSphere adapters available as part of the product release. There are technology adapters for JDBC and Flat File, and application adapters for SAP, PeopleSoft and Siebel EIS.

## Section

# *Interaction Styles*

This section will briefly cover the Interaction Styles

## Interaction Styles

- There are 2 types of interactions:
  - ▶ **Outbound**: Client initiated
  - ▶ **Inbound**: EIS-initiated
- There can be two modes of interaction for each interaction style
  - ▶ Request-Response – For Outbound interaction
    - Request/Response interaction takes a request and returns a response
  - ▶ One-way - For Inbound interaction
    - Takes a request, but does not return a response
- Interaction style is specified during discovery service within WID
  - The generated request-response method for outbound has both input and output specified, whereas for One-way method for inbound interaction has only input arguments specified



A outbound request or interaction is initiated by a client. The client could be a SCA component within the same module or some other module, or could be some external client like a JSP.

An inbound request is initiated by the backend EIS based on some event on the EIS business object, like create, update or delete. The adapter gets the event either by push or poll, based on how the specific adapter interacts with the EIS, and then fetches the business object from EIS before sending that to the listener that will handle the event.

The outbound interaction is a request-response interaction, where usually the copy of the business service is returned.

The inbound interaction is one way only where the event gets passed to the adapter.

As discussed before, the interaction style (inbound or outbound) is selected during the discovery service process within WID.

## Section

# ***Inbound Support, Event Manager and Event Handling***



This section will cover support for inbound events through the event manager.

## Inbound Support

- An application client receives events from the resource adapter for changes in the EIS components/objects
- Application server passes an Endpoint Factory along with an ActivationSpec to the resource adapter that outlines the endpoint-specific configuration
- The inbound event delivery mechanism uses a listener interface through which the resource adapter delivers events
  - ▶ Listener interface are different for different adapters
  - ▶ Resource adapter “listener” is the interface that endpoints are expected to expose and through which the resource adapter will deliver events to the endpoint
- The Adapter Foundation classes standardizes the inbound message listener and the event delivery mechanism for the different adapters



The JCA 1.5 specification added support for inbound events to the adapter. Clients interested in receiving the events register with the adapter using the ActivationSpec of the adapter.

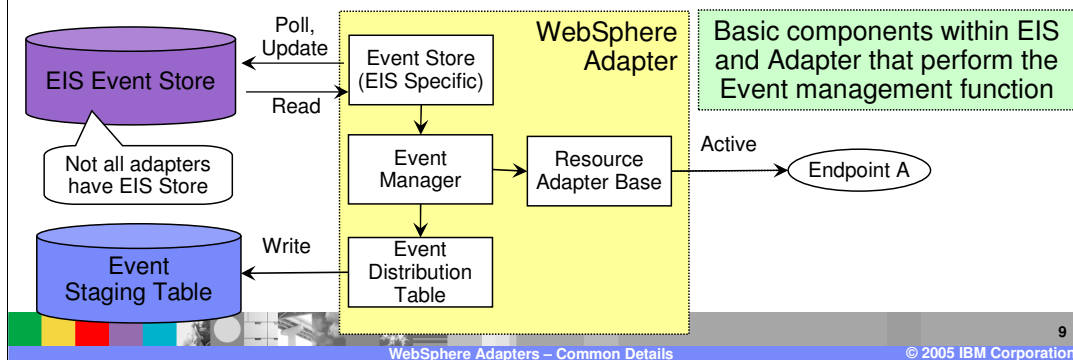
The client endpoint implements the listener interface that the adapter defines. The adapter delivers the events through these endpoints.

The Adapter Foundation classes standardizes the inbound message listener and the event delivery mechanism for the different adapters. It provides the event management framework that can be used by all the Adapters.



## Event Management Foundation

- When an event is retrieved from the EIS, a reference to the event is written to the staging table for each active endpoint
- Each event in the staging table is then delivered to the endpoint as part of a unique XA transaction controlled by the WebSphere Process Server, providing assured “once and only once” delivery
- When the endpoint consumes its event and the transaction is committed, the reference to the event in the staging table is removed for that endpoint and the event in the EIS Event Store is deleted
- Only Asynchronous Event Delivery supported in this release for all Adapters



This diagram shows the different components that enable the event management framework to handle incoming events. This is provided by the Adapter foundation classes for all the adapters to exploit.

The EIS Event Store is the store within the EIS system. Not all EIS back-ends have an EIS store. For example, SAP EIS does not have an EIS Store and lets the adapter manage the events.

The event management framework supports both situations – whether EIS Event store is supported or not.

When adapter gets the event from the EIS either by push or pull, a copy of the event is stored in the Event staging table. The event staging table could be a persistent store like a database or could be in-memory. Persistent event staging tables support server failover.

Events from the staging table are delivered to the configured endpoints. The EIS event store and Event staging table are updated to indicate the status of the events within the adapter, as the event is being processed within the adapter and the endpoint.

The event management framework supports “once and only once” delivery of events. The next few pages will cover the process of how the events are handled through the end to end processes.

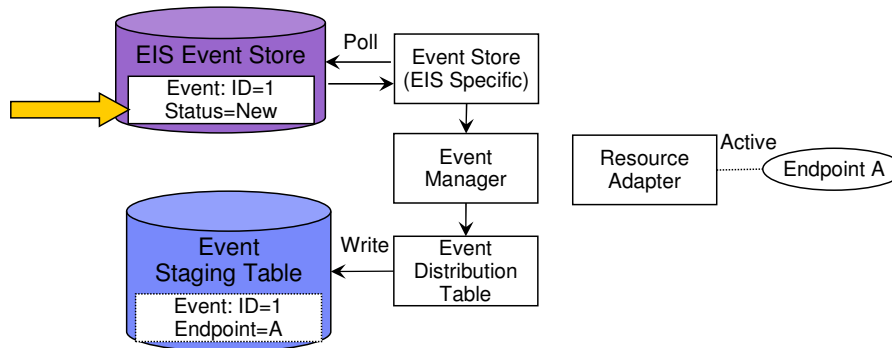
\*IBM Confidential\*

WPSWIDV6\_Adapter\_details.ppt

Page 9 of 41

## Assured Event Delivery – Step 1

Step 1: Event detected in EIS and record added to staging table for the active endpoint



10

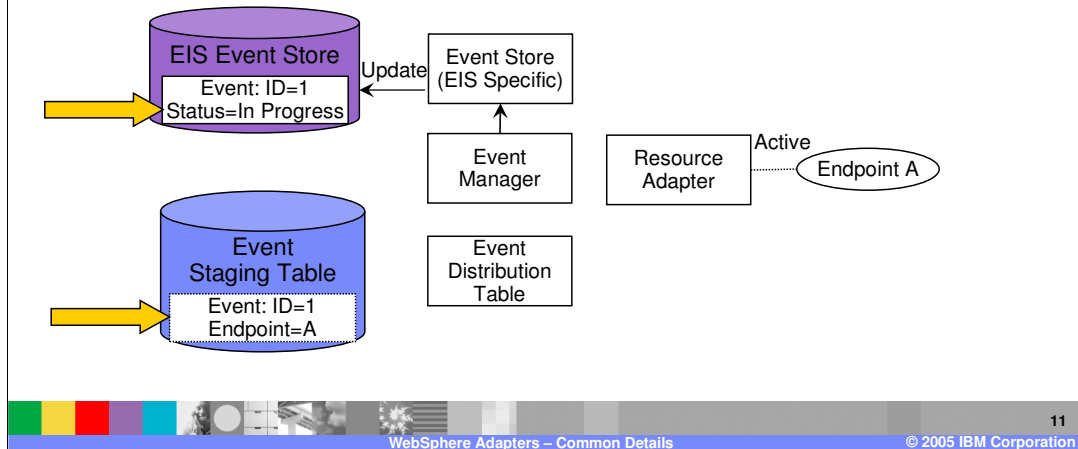
WebSphere Adapters – Common Details

© 2005 IBM Corporation

In step 1, a new event is added to the EIS event store and the status of the event is marked as “New”. During the poll time, the adapter will fetch all the events marked “New”. In the case where there is no EIS Event store, the event is pushed to the adapter by the backend EIS.

## Assured Event Delivery – Step 2

- Step 2: Event in EIS is marked in-progress

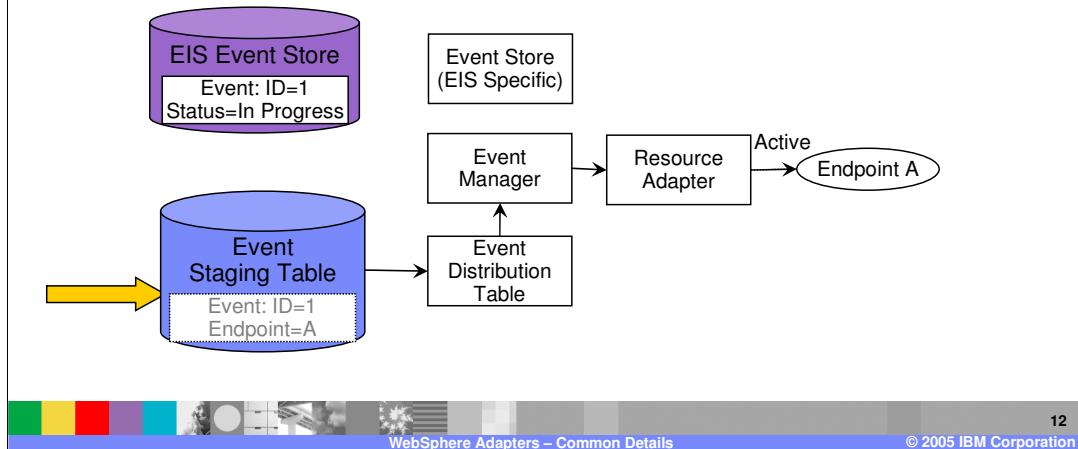


In step 2, the event is saved in the Event staging table within the adapter along with the event ID, the endpoint that is the registered listener of the event.

In addition, the event is marked as “In progress” in the backend EIS Event store.

## Assured Event Delivery – Step 3

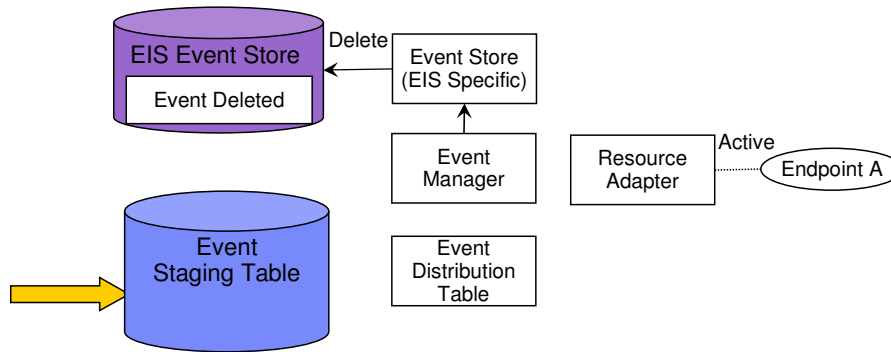
- Step 3: Then the event is published to the active endpoint and simultaneously deleted from the staging table as part of an individual XA transaction (encompassing the endpoint and its specific record in the staging table)



In step 3, the event is published to the active endpoint and deleted from the staging table. The two actions might be part of an XA transaction if the endpoint supports transaction. The delivery of the event to the endpoint indicates the end of the transaction.

## Assured Event Delivery – Step 4

- Step 4: Once the endpoint has received the event (as determine by lack of any remaining references in the staging table), the original event is deleted from EIS



In the last step 4, once the endpoint has received the event, the original event in the EIS Event store is marked for deletion. The event is now considered processed end to end.

## Event Status

- Event Store logic compares the status of the event in the EIS event store with a record of the event in the staging table as described below.
- This approach ensures that event processing logic always proceeds in the same sequence and, if there is an unexpected failure, the exact point in the sequence can still be identified during recovery so no event is delivered more than once or not at all

Event State		Interpretation	Action by Event Manager
In EIS Event Store	In Adapter Event Staging Table		
Event marked as 'new'	No record of event	This event is a new event being detected for the first time	Perform steps 1-4 as described earlier
Event marked as 'new'	one or more records corresponding to this event exist	The event manager was previously interrupted after it had detected and copied the event to the staging table but before it could update the event as in-progress	Perform steps 2-4
Event marked as 'in progress'	one or more records corresponding to this event exist	The event manager was in the process of delivering the event when it was interrupted	Perform steps 3-4
Event marked as 'in progress'	no record of event	The event manager successfully delivered the event but was interrupted before it could delete the event from the event store	Perform step 4

This table shows how “once and only once” event delivery is assured through the steps indicated in the last few pages. When failures occur where the event manager was interrupted, on a restart of the event manager, it looks at the event status in the EIS event store and in the staging table to see what logical steps 1 to 4 need to be performed on the event.

Table entry 1 shows an event in the EIS Event store marked as “New” and there is no record in the staging table. This would indicate that a new event is being detected for the 1<sup>st</sup> time. All the steps 1 to 4 will be performed.

Table entry 2 shows an event in the EIS Event store marked as “New” and there is a record in the staging table. This would indicate that the event manager saved the event in the staging table and before it could update the EIS event store to mark the event as “In process”, the event manager was interrupted. Steps 2 to 4 will be performed.

Table entry 3 shows an event in the EIS Event store marked as “In progress” and there is a record in the staging table. This would indicate that the event manager was interrupted between steps 2 and 3, where the event was in the process of being delivered to the endpoint. Steps 3 and 4 will be performed.

Table entry 4 shows an event in the EIS Event store marked as “In progress” and there is NO record in the staging table. This would indicate that the event manager had send the event to the endpoint successfully and was interrupted before deleting the event in the EIS event store. Step 4 will be performed.

## Event Error Handling

- If the endpoint throws an exception during delivery, the event manager will stop delivering events to that endpoint, and the timer task for polling will stop
  
- Error at the Endpoint
  - ▶ For ORDERED delivery type, the remaining events polled in that cycle will not be delivered until the event with the error is processed
  - ▶ For UNORDERED delivery type, the event manager will attempt to deliver the remaining events in the current poll cycle
  - ▶ When the endpoint is taken offline and re-activated, the event(s) in which the error occurred will be re-delivered, and normal delivery will continue
  - ▶ If the endpoint is transactional, and the transaction rolls back, it will be treated the same as if the endpoint threw an exception
  
- Error in the Event Store
  - ▶ If the Event Store throws an exception during the getObjectForEvent call (for retrieval of the full event), the event manager will mark the status of that event in the Event table to ERROR\_PROCESSING\_EVENT
  - ▶ For "ORDERED" delivery, the polling task will stop until the endpoint is re-activated.
  - ▶ For "UNORDERED" delivery, the polling task will continue.

When an endpoint throws an exception or the endpoint is stopped, event manager stops delivering the events to that endpoint and the polling for new events will stop, until the endpoint restarts without exception.

The error handling is different for ORDERED and UNORDERED events.

## Section

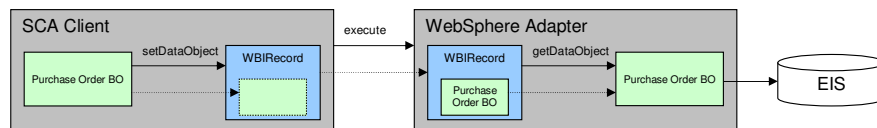
# ***Business Objects***

This section covers the Business Objects and how they are used as the core component in the WebSphere Adapter architecture.



## Business Objects (BO)

- For outbound operation, BOs are used to describe the structure and content of arguments to functions on the EIS client interface or the EIS business objects
- However, CCI client and adapter exchange data via the Record model, as part of JCA specification
  - ▶ *Boolean execute(InteractionSpec ispec, Record inRecord, Record outRecord)*
  - ▶ *Record execute(InteractionSpec ispec, Record inRecord)*
- BO model is not compatible with Record model
- Solution: Adapter Foundation classes provides a Record implementation that wraps the BO to allow it to be passed between the client and the resource adapter



17

WebSphere Adapters – Common Details

© 2005 IBM Corporation

The SCA client uses a Business Object to pass data between the client and the WebSphere Adapter. For an outbound request, the Business Object describe the function or the business service to call in the backend EIS. The Business Object also contains metadata representing the arguments to be passed to the EIS function.

However, the J2C specification does not understand Business Object. The data between a JCA client and JCA Adapter is passed using the common client Interface Record object so the Business Object model is not compatible with the JCA Record object. The problem is solved by wrapping the Business Object inside the Record object and passing the Record containing the wrapped Business Object to the adapter. The diagram shows the interaction between the SCA client and the Adapter. The Adapter extracts the Business Object from the Record object and determines the EIS function to invoke passing the parameters that are in the Business Object as metadata.

## Advantages of using Business Objects (BO)

- BO model is based on the open-standard service data object (SDO) model, and provides extensions and additional functionality to make this model more robust
- BO model aligns well with the larger WebSphere Service-Oriented Architecture (SOA) strategy
- BO model provides support for tracking changes at both object- and property-level to better enable delta update functionality



The Business Object model and the Service component Architecture are the core component of the Business process application following the Service Oriented Architecture principles.

## Section

# ***Common Configuration for All Adapters***

This section covers the common configuration that applies to all the WebSphere Adapters.

## Resource Adapter Configuration

- JCA related configuration artifacts are created as a result of service discovery process in WebSphere Integration Developer
- Properties may be modified after the application has been deployed to the WebSphere Process Server via Administrative Console or wsadmin
  - ▶ ResourceAdapter
    - global configuration properties
  - ▶ ManagedConnectionFactory
    - properties for a target EIS instance (outbound)
    - examples: hostname, port
  - ▶ ConnectionSpec
    - connection instance properties (outbound)
    - examples: user name, password, language
  - ▶ InteractionSpec
    - per request configuration properties (outbound)
    - examples: operation to be invoked, return values required
  - ▶ ActivationSpec
    - properties configurable by a specific endpoint (inbound)
    - examples: durability, business objects of interest

When the developer uses the Enterprise Service Discovery wizards to create the outbound or inbound request to and from backend EIS, the JCA properties for the Resource Adapter, Managed Connection Factory for outbound request and ActivationSpec for inbound request are already created based on the input to the service discovery wizard.

Once the application containing the RAR is deployed, the administrator can modify the properties using the Administrative console or command line “wsadmin” tool.

## Connection Properties

### Inbound Connection Properties

<i>Name</i>	<i>Description</i>
Resource Adapter	Resource Adapter and its custom properties
ActivationSpec	Adapter Activation Spec.

### Outbound Connection Properties

<i>Name</i>	<i>Description</i>
Resource Adapter	Resource Adapter and its custom properties
Managed Connection Factory	Adapter Managed Connection Factory

Each Adapter may have their own additional properties

As indicated in the previous page, the properties for the Inbound request are in the Resource Adapter and the ActivationSpec, whereas the properties for the Outbound request are in the Resource Adapter and the Managed Connection Factory.

## Common Adapter Configuration Properties for both Inbound and Outbound

<i>Name</i>	<i>Description</i>
Hostname	Name of the machine hosting the EIS.
Port	Port on which the EIS instance is listening.
Username	User name for logging into the EIS for inbound events
Password	User password for authorizing the user to retrieve events from the EIS
Timeout	Amount of time in milliseconds to wait for a response from the EIS
EISEncoding	Used if adapter needs to convert Unicode to/from native encoding like in case of file handling etc.. This should specify the native encoding to use.
AdapterID	The id of this deployment of the adapter

This table lists the common properties for outbound and inbound requests needed for the Enterprise Service Discovery tool to connect to the EIS. These are also used as the attributes on the Resource Adapter.

## Common Adapter Logging and Tracing Properties

<i>Name</i>	<i>Required</i>	<i>Description</i>
LogFilename	No	The full path of the log file
LogNumberOfFiles	No	The number of log files to use. When a log file reaches its maximum size it will start using another log file. Default = 1
LogFileMaxSize	No	Size of the log files in kilobytes. Default = No maximum size
TraceFilename	No	The full path of the trace file
TraceNumberOfFiles	No	The number of trace files to use. When a trace file reaches its maximum size it will start using another trace file. Default = 1
TraceFileMaxSize	No	Size of the trace files in kilobytes. Default = No maximum size

The logging and tracing properties for all the WebSphere Adapters are listed here. The file name, maximum file size and the number of history log files are the main options for files for logging and tracing.

## Common ActivationSpec Attributes for Inbound Events

Name	Required	Description
PollPeriod	Yes	The rate (in milliseconds) at which to poll the EIS event store for new inbound events. Default: 500
PollQuantity	Yes	# of events per poll cycle for each endpoint - Default: 1
EDTDriverName EDTDatabaseName EDTTableName EDTUserName EDTUserPassword EDTSchemaName EDTURL EDTServerName EDTPortName	No	Event Distribution Table (EDT) is used for Event staging information by the adapter For configuration of the EDT, a particular XA provider may use any combination of the EDT properties in addition to the EDTDriverName. If additional property needed for a Datasource and not present in this list, add them to a file called "EDTExtendedProperties.properties". Default Cloudscape database is used
DeliveryType	No	Determines the order in which the events will be published. One at a time (ORDERED) or all at once (UNORDERED) – Default: ORDERED
AutoCreateEDT	No	Flag that indicates whether the adapter should create the EDT table automatically if it doesn't already exists – Default: TRUE

For adapters that poll the EIS system for new events, the poll period and # of events per poll cycle can be customized.

For adapters that require persistent event distribution or staging table within the adapters, all the entries starting with EDT need to be provided.

If there are additional properties required by the event distribution table than what is specified in the wizard, those properties can be specified in the "EDTExtendedProperties.properties" and added to the adapter install directory. The property file uses a standard Java properties file format.

EDTUserName, EDTUserPassword, and EDTTableName are always required, so they are not listed below. EDTSchema is only required for a non-default schema. Refer to the specific Adapter Information Center for the values of the different EDT properties, such as the EDTDriverName and EDTURL.

The inbound event DeliveryType attribute can be ORDERED or UNORDERED, based on whether the events can be processed in a sequence as they come in (ORDERED) or could be in any order (UNORDERED).

Setting the attribute AutoCreateEDT to true, the Adapter will create the table on the database specified in the EDT entries.

\*IBM Confidential\*

WPSWIDV6\_Adapter\_details.ppt

Page 24 of 41



## Section

# ***Locale and Globalization Support***

This section describes the Locale and Globalization support in WebSphere Adapters

## Locale and Globalization Support

- Adapter supports single and double-byte character sets and can deliver message text in the specified language
  - ▶ Adapter supports the processing of bidirectional script data for Arabic and Hebrew languages.
- The bidirectional properties must be configured to use the bidirectional capacity
- If the database bidirectional format differs from the Windows standard format, all properties with bidirectional support are transformed from the Windows standard format to the target database bidirectional format
- Note that the logging of error and information messages uses the locale of the system on which WPS is running



The adapters support BiDi support of single and double-byte character sets and can deliver text in the specified language. It also supports processing of bidirectional script data for Arabic and Hebrew languages. The BiDi configuration is done in the Enterprise Service Discovery panels within WebSphere Integration Developer.

## Bidirectional Language Configuration Properties

- WPS uses bidirectional language format of ILYNN (implicit, left-to-right, on, off, nominal)
  - ▶ Same as the Windows bidirectional language format
  - ▶ All other bidirectional language formats must be converted prior to being used by WPS
- Attributes to configure bidirectional language format:

Letter Position	Purpose	Values	Description	Default Setting
1	Order Schema	I or V	Implicit (Logical) or Visual	I
2	Orientation	L or R	Left-to-Right or Right-to-Left	L
3	Symmetric Swapping	Y or N	Symmetric Swapping is on or off	Y
4	Shaping	Y or N	Text is shaped or not shaped	N
5	Numeric Shaping	H, C, N	Hindi, Contextual, Nominal	N

27

WebSphere Adapters – Common Details

© 2005 IBM Corporation

WebSphere Process Server uses the ILYNN BiDi language format. The position of letters in ILYNN represent (1) Implicit (Logical) or Visual, with default being Implicit, (2) Orientation with default of left to right (3) Symmetric Swapping with default being yes (4) Text Shaping with default being No and (5) Numeric Shaping with default being No.

The ILYNN format is same as Windows bidi language format. All other bidirectional language formats must be converted prior to being used by WPS.

## Bidirectional Language Configuration

- The bidirectional properties can be set at the following different levels:
  - ▶ Resource adapter
  - ▶ J2C connection factory
  - ▶ J2C activation specification
- Once defined, they can be modified in WID and WPS
  - ▶ WID: Via the Assembly Editor on the EIS Import or Export SCA component
  - ▶ WPS: Via the Resource Adapter properties
- Bidirectional properties for BO, BO attributes, and application-specific metadata attributes: Use annotations in the Business Object Designer within WID

The bidi configuration can be set at the Resource adapter, J2C Connection factory and J2C Activation spec levels. With WebSphere Integration Developer, they are defined when using the service discovery tool and can later be modified using the Attribute panel on the Adapter EIS SCA components.

The BiDi properties for Business Object, or attributes on Business Object can be specified using annotations in the Business Object Designer editor within WebSphere Integration Developer.

## Section

# ***Problem Determination: Common for All WebSphere Adapters***

This section covers the common Problem determination for all WebSphere Adapters.

## Problem Determination – Log and Trace Files

- WebSphere Process Server
  - ▶ <WPS\_Home>\profiles\<profile>\logs\<server>\SystemOut.log
  - ▶ <WPS\_Home>\profiles\<profile>\logs\<server>\SystemErr.log
- Adapter Logs and Traces Files
  - ▶ Messages and Exceptions logged in these files
  - ▶ Log and Trace file location and levels specified during EMD
  - ▶ Can change log and trace file setting through the RAR custom properties settings in the WebSphere Process Server
- EMD log, specified during EMD 
  - ▶ Default: <WID\_workspace>\.metadata\<Adapter>MetadataDiscovery.log
- WebSphere Integration Developer
  - ▶ <WID\_workspace>\.metadata\.log



The location of the log and trace files are specified here.

The WebSphere Process Server log files are the System out and System error log files in the profile logs directory of the server.

The adapter trace file locations are specified using the log and trace file attributes on the adapter. In addition, the trace files location and trace strings are specified in the Administrative console of the Process Server for the server's change log and trace file option.

While running the Enterprise Metadata or Service Discovery tool, the log file is within the WID workspace in the directory specified.

The overall log file for WebSphere Integration Developer is in the workspace metadata log file.

## Problem Determination: Logging

- User messages and Language translated

Logging Level	Content
Fatal	Task cannot continue. Component cannot function.
Severe	Task cannot continue. Component can still function. This also includes conditions that indicate an impending fatal error, i.e., reporting on situations that strongly suggest that resources are on the verge of being depleted.
Warning	Potential error or impending error. This also includes conditions that indicate a progressive failure, i.e., the potential leaking of resources.
Audit	Significant event affecting server state or resources
Info	General Information outlining overall task progress
Config	Configuration change or status
Detail	General Information detailing subtask progress

The different logging levels are specified on this page. These messages are translated and they appear in the WebSphere Process Server System out log files.

## Problem Determination: Tracing

- Service level information for IBM Support
  - Not translated
  - Specified at the Adapter level through its custom properties

Tracing Level	Content
Fine	Trace info – general trace, plus method entry / exit / return values
Finer	Trace info – detailed trace
Finest	<ul style="list-style-type: none"><li>▪ Trace info – most detailed trace</li><li>▪ Includes all detail needed to debug problems</li></ul>

The trace strings are intended for use by IBM Support. The different tracing levels are specified on this page. These are specified at the Adapter through its custom properties.



## Section

# *Summary and Reference*

This section will provide a Summary and References.

## Summary

- This presentation covered the details of some of the common functions in all the IBM WebSphere Adapters
  - ▶ Enterprise Metadata or Service Discovery and creation of EIS SCA components
  - ▶ Event Management Framework
  - ▶ Common attributes for J2C adapters, Managed Connection and ActivationSpec
  - ▶ Problem Determination log and trace levels and file locations
- Adapter Foundation classes provide the foundation needed for adapter implementation

The summary page lists the topics discussed in detail in this presentation .

## References

- Java Connector Architecture
  - ▶ <http://java.sun.com/j2ee/connector/index.jsp>
- Enterprise MetaData Discovery Whitepaper
  - ▶ <http://www.ibm.com/developerworks/java/library/j-emd/>

## 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.

© Copyright International Business Machines Corporation 2005. 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.

