



IBM Software Group

WebSphere Business Process Management Suite V6.2

WebSphere Adapters for JDBC V6.2



@business on demand.

© 2009 IBM Corporation
Updated April 29, 2009

This presentation provides a general overview of the IBM WebSphere® Adapters V6.2

Agenda

- Overview
- Business object model
 - ▶ Inbound operations
 - ▶ Outbound operations
- Event management
- Summary and references



This presentation covers details of the IBM WebSphere Adapters. It starts with reviewing the new enhancements in the development and deployment process of JDBC Adapter. It also introduces new outbound operation for V6.2. The last section provides information about event management in inbound processing. For more information about the new and improved Enterprise Metadata Discovery, refer to the JDBC demonstrations.

Section

Overview



This section provides an overview of the new enhancements in deployment of the WebSphere Adapter for JDBC.

Overview: background

- IBM WebSphere adapter for JDBC implements the Java™ EE Connector Architecture (JCA), version 1.5 specification
- Enables bi-directional connectivity for integration with database applications providing Inbound and Outbound support
 - ▶ Requires database with JDBC driver that supports the JDBC 2.0 or higher specification



The IBM WebSphere Adapter for JDBC implements the JCA 1.5 specification and enables bi-directional connectivity, both inbound and outbound, with those Enterprise Information System business applications that communicate with database applications. The JDBC adapter supports databases that have drivers with implementations of the JDBC 2.0 or higher specification. For inbound services, a set of sample scripts is provided for setting up event stores in DB2®, Oracle, and SQL.

Development in WebSphere Integration Developer

- Package JDBC adapter resource archive file (CWYBC_JDBC.rar) with WebSphere Integration Developer
 - ▶ Automatically import the adapter and create a Java EE connector project in the workspace
- Run Enterprise Metadata Discovery wizard
- Complete Service Component Architecture (SCA) application assembly
- Test application using the Test Environment in WID
- Export as Enterprise Archive file (EAR) to be installed to WebSphere Process Server



You can import the JDBC Adapter RAR file, which is available in the Resource Adapters directory, into WebSphere Integration Developer. Importing this file creates a Java EE connector project in your workspace. Using the enterprise metadata service process, you then enter the adapter configuration information and select the service type, either inbound or outbound. Next, you select the particular operations, the logging and tracing file names, inbound and archive subdirectory names, and other adapter specific properties. Upon completion of the enterprise metadata discovery wizard, you are returned to the business integration perspective, along with a module containing the necessary adapter artifacts for integration and assembly with other Service Component Architecture, or SCA, components. When you complete your application assembly, the application can be tested and debugged within the WebSphere test environment, exported as an Enterprise Archive, or EAR file, and ultimately installed into a stand-alone WebSphere Process Server runtime.

Deployment in WebSphere Process Server

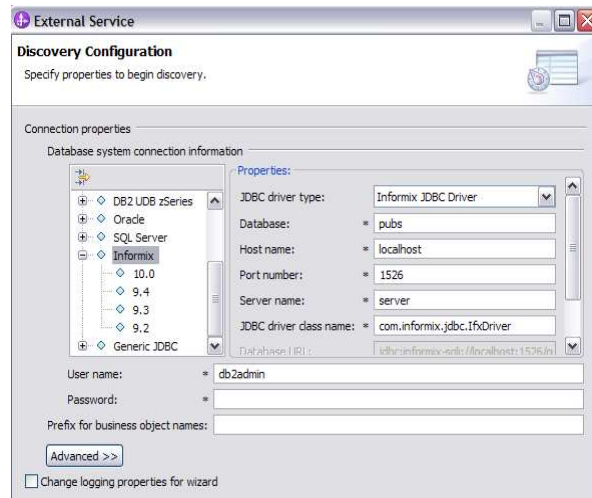
- Use administrative console or wsadmin command line tool
- Create J2C authentication alias specifying user ID and password to database
- Create Data Source JNDI name
- Install new enterprise application (the EAR just exported from WebSphere Integration Developer)
 - ▶ Option to deploy RAR separately
 - ▶ For the most part, accept defaults
 - Will need to specify authentication alias in the “Map resource references to resources” panel
 - ▶ Continue through installation, Finish, Save to Master configuration file

Installation of the application containing the WebSphere Adapter for JDBC to the WebSphere Process Server, is very similar to installing any other enterprise application. However, in the case of outbound communications with the JDBC adapter, you can use the J2C authentication alias which contains the user ID and password that are used to connect to the application database. After creating the Data Source JNDI name, you install the EAR file that you just exported from WebSphere Integration Developer, into the server. You can choose to deploy the RAR file separately, or include it with the application in the EAR file. In general, you can go with the installation defaults, but on the panel that maps resource references to resources, you will have to specify the J2C authentication alias.

Informix database



- Identity columns in Informix® database
 - ▶ Support auto-generation for serial columns
 - ▶ Retrieve auto-generated values for the serial columns
 - ▶ New connection property is added for Informix database



In a databases table in Informix, if users have a column which is of type Serial then Informix database supports auto-generation of the values for these columns. Informix allows users to define columns in a table as "serial" types, which means when a row is inserted in a table, database automatically generates the value for the serial columns. Adapter runtime automatically discovers these auto-generated values from the database.

In Enterprise Metadata Discovery (EMD), a new connection property is added to include Informix database in discovery configuration and part of database vendor lists in deployment configuration. The adapter automatically discovers that a particular column is of type Serial or Serial8 so in the generated artifacts, Unique Identifier (UID) ASI is automatically set to serial or serial8.

Section

Business object model



This section provides an overview of the business object models.

Business object: models

- Supports generating business objects from different types of objects
 - ▶ Tables and views
 - ▶ Stored procedures and stored functions, and synonyms or nicknames
 - ▶ Query business object
 - User specified statement



The JDBC Adapter's EMD processing allows you to generate business objects from the objects in a database. Currently it supports generating business objects from five types of objects. The first type of object includes Tables and views, the second type includes stored procedures, stored functions, and synonyms or nicknames. The third type of object is the query business object, used for user specified select statements.

For the first and second types, the JDBC EMD step will allow you to discover the objects by first generating a list of all the schemas in the database. Within each schema there are lists of Tables, Views, Stored Procedures and Synonyms or Nicknames that you are allowed to select from, and the corresponding business object is generated. The JDBC EMD processing will analyze the metadata of the selected objects and generate required attributes in the business object. The attributes are generated based on the columns of those selected database objects.

For the third type, the JDBC EMD wizard will allow you to specify one select statement. In addition to this, you will also need to specify the business object name, the types, and dummy values of all parameters in the "where" clause. The JDBC EMD wizard will process the select statement, and analyze the metadata of the returned ResultSet to get the list of columns and types. After putting together this information, the JDBC EMD processing will generate one Query business object for this select statement.

Business object: models

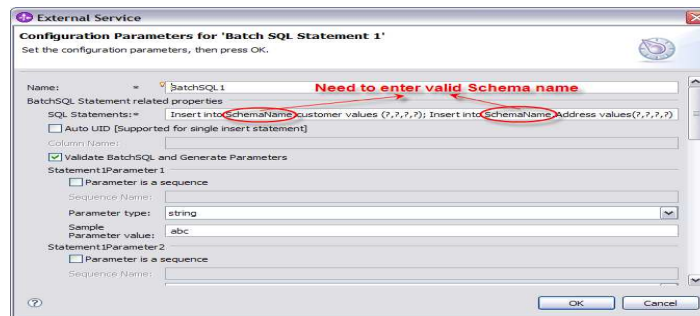
- Wrapper business object
 - ▶ Generate using input provided by user
 - ▶ Add multiple table business objects inside a wrapper business object

10

The fourth type of supported object is the Wrapper Business Object, which wraps together multiple Table Business objects. The JDBC EMD wizard asks you to select multiple table business objects. It uses them to generate a top level wrapper business object which has as it's own attributes, all the key attributes of the selected table business objects. The attributes of this Wrapper business object will not have any ColumnName ASI, since it is really a dummy business object and does not actually have an EIS representation. In the Business Object ASI, it will have Wrapper ASI set to true, but not have any TableName ASI. The Wrapper business object contains the primary keys of the selected table business objects, and structures all the selected table business objects, as N cardinality "child" business objects.

Business object: models

- The batchSQL business object
 - ▶ User specified insert, update, or delete SQL statements
 - ▶ Single or multiple semi-colon separated statements
 - ▶ Dummy values of all parameters
 - ▶ Parameter value can use a sequence generated by the database
 - ▶ Specify identity column
- Generate one batchSQL for all SQL statements



Last but not least, this last type supports for an additional type of object that gives you more capability to specify other SQL statements, beyond the single Select statement supported by the Query object. This fifth supported Object is the BatchSQL business object which allows you to supply your own user-specified Insert, Update, and Delete SQL Statements. More advanced users who have a good knowledge of databases and who understand SQL statements will benefit from this new feature. It allows you to directly perform these operations on database entities without having to use table business objects.

For this object type, the JDBC EMD wizard allows you to specify single or multiple, semi-colon separated Insert, Update, Delete SQL statements, and the types and dummy values of all parameters are included in the specified SQL statements. You can also specify whether there is an Identity column, which is auto-generated, in the SQL Statement and whether any parameter value should use the sequence generated by the database. The JDBC EMD processing will generate a single BatchSQL business object for all the specified SQL statements, including the parameters that correspond to each SQL statement. At runtime, this BatchSQL business object will process each of the specified Insert, Update and Delete SQL Statements.

If you want to generate a BatchSQL business object from a set of specified Select statements, you need to select the "BatchSQL business objects" check box in the Query Filter Properties window. Then you can set the "BatchSQL business object count" to the maximum number of BatchSQL business objects that are generated when you click "Execute Query". When you look in the navigation tree, you find one top level node labeled "BatchSQL Statements". And by expanding this node, you find child nodes labeled "BatchSQL Statement 1", "BatchSQL Statement 2" and so on. These statements have a dependency on the value for "BatchSQL business object count". The child nodes are all selectable, and you can select any of them to generate a BatchSQL business object.

There are three types of attributes associated with BatchSQL business objects. One type contains all the parameters in all the user-specified SQL Statements. The second attribute type is the name for the Auto UID column, if you specified one. The last attribute type corresponds to the status returned on execution of the SQL Statements.

Business object: inbound operations

- Operations
 - ▶ A business object is passed to the adapter as a request that is processed according to the operation specified in the business object
- Inbound operations
 - ▶ After image support only
 - Create
 - Update
 - Delete



A business object's content is related to the amount and purpose of the information conveyed by the business object. There are two categories of business object content, "After image" and "Delta". After image content represents the state of the business object after all changes have been made. This type of content is indicated by the presence of a top-level verb in the business graph of the business object.

Delta content represents the changes that have happened to the business object content. This type of content is indicated by the absence of a top-level verb.

The operations specified in the business object indicate what type of processing is being requested. Business objects are passed to the adapter and processed according to the operations they specify. In the case of inbound operations, only after-image support is available for the operations, which include Create, Update, and Delete.

Business objects: outbound operations

- After Image
 - ▶ Create
 - ▶ Update
 - ▶ Delete
 - Supports physical and logical deletes
- Delta support
 - ▶ ApplyChanges only handles ChangeSummary



In the case of outbound operations, after-image support is available for the Create, Update, and Delete operations. Delete operations can include both physical and logical deletes.

For Delta support, outbound operations also include the ApplyChanges operation.

Business object: outbound operations



- Other supported operations
 - ▶ Retrieve
 - ▶ RetrieveAll
 - Allows an array of business objects to be retrieved from the database
 - ResultSetLimit property determines number of records to return
 - ▶ Execute
 - Stored Procedures
 - User defined SQL statements
 - ▶ Exists – new operation
 - Supports by Tables/Views/Synonyms/Nicknames



Outbound supports three other operations as well, retrieve, retrieveAll and Execute. The retrieve operation, when passed a hierarchical business object, starts processing by first making a copy of the top level business object without any of its children. It then recursively retrieves the child business objects starting with the parent of the incoming business object.

RetrieveAll is an operation that will direct the JDBC adapter to retrieve an array of business objects from the database. The value of the ResultSetLimit property determines the number of records to return. This value can be set during the EMD processing or by specifying the value for MaxRecords on the Configure Objects panel for the outbound ServiceType.

The Execute operation is for both Stored Procedures and for user defined SQL statements. In support of Stored procedures, the Execute operation generates the required stored procedure that corresponds to the stored procedure definition in the database. For user defined SQL statements, this operation generates the required BatchSQL business object based on user defined SQL statement or statements.

Lastly, a new operation Exists is also introduced to the list of supported operations. The Exists operation determines whether the database table contains records that match the attributes set in a business object. The Exists operation is supported by the table business object, the views business object, the synonyms, and nicknames business object. More details is covered in the next few slides.

Business objects: outbound operations



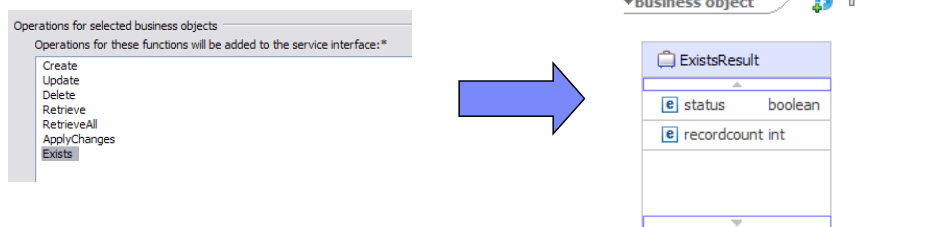
Exists operation

Generates output business object – ExistsResult

- Return in Boolean if the record is found
- Return number of records for specified query

Query records based on any attribute value

- Not just on primary key attributes
- For example
 - Customer { custid primary key, firstName, lastName, Organization }
 - Query the existence of a customer based on any of the attributes in Customer business object

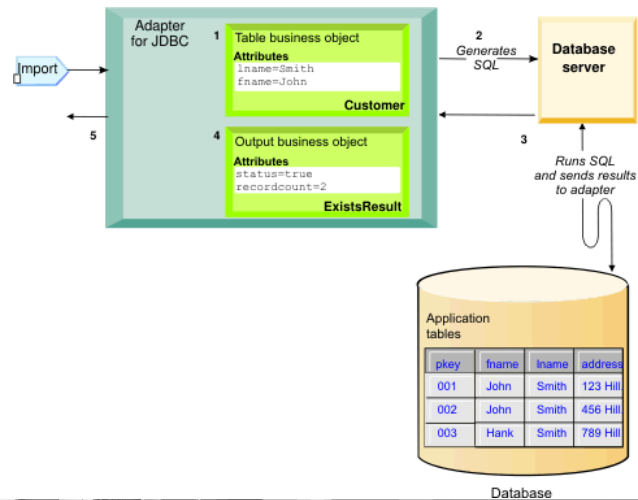


Continued from previous slide, the results of the Exists operation is returned in a different business object than the input business object. It actually generates a new business object called ExistsResults, which is used as the output result based on query criteria set on the attributes of input business object. The business object ExistsResults has two attributes. Status attribute returns with Boolean value if the record is found in the database and recordcount attribute returns the number of records found in the databases for the specified query. The default value for record count attribute is 0. The screen capture illustrates the structure of the ExistsResults business object returned from an Exists operation.

The Exists operation also supports both key and non-key attributes in the selection criteria. It sends that object as an input to query a records in the database. For example, if user has generated a BO for a Customer table which has a primary key and non-primary key attributes, then user can query the existence of a customer based on any of the attributes in Customer business object. And not just query based on primary key attributes only.

Business objects: outbound operations

- Exists operation
 - ▶ Supports flat/hierarchical business objects

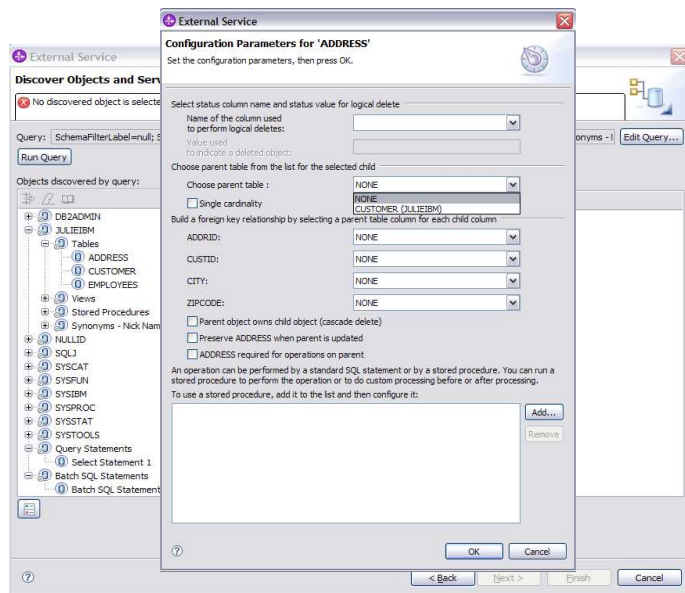


The above illustration shows how the adapter processes a table business object with an Exists operation. In this scenario, the adapter receives a Customer table business object from the import. This input business object can be flat (simple with no child business objects) or hierarchical (complex, containing one or more child business objects). If the business object is hierarchical, it is only for the top-level business object (the individual business object at the top of a hierarchical business object) for which the adapter builds the query.

It then uses this input table business object to generate an SQL SELECT statement and sends to the database server. In this case, the SQL statement specifies *non-primary key* attributes *fname* with the assigned values of John and *lname* with the assigned values of Smith. The adapter includes the attribute information from the table business object in the *where* clause of the SQL query. The database server runs the SQL query and sends the results back to the adapter. Then it obtains the results of the SQL query from the database server and sets the *recordcount* and *status* attributes on the *ExistsResults* business object. Finally, the adapter returns the *ExistsResult* business object to the caller.

Business object: hierarchy

- Allows to build a hierarchy of Parent-Child business objects
- A mechanism for building the relationship using Enterprise Metadata Discovery
- Supports both inbound and outbound



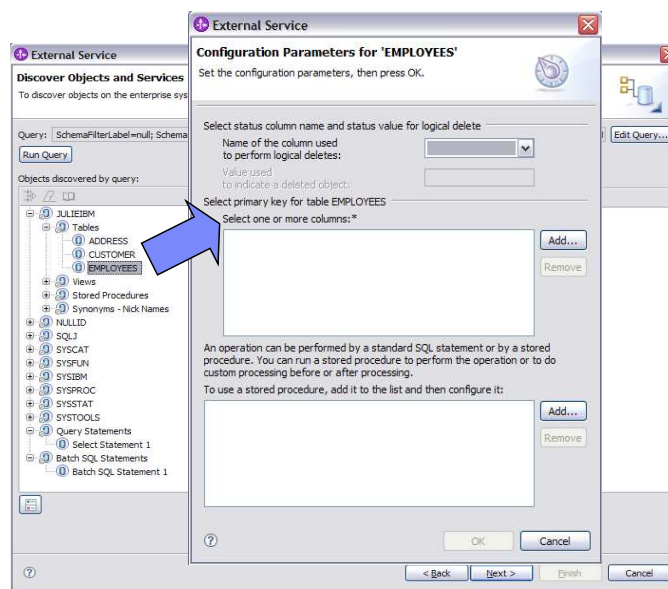
The adapter can now allow you to build a hierarchy of Parent-Child business objects using the EMD process. This mechanism can be invoked at a separate screen during EMD wizard. For any Business Object, for example “Customer”, if you want to associate a child Object, like “Address”, you first select the Address object from a drop down list of Business Objects. The Query business objects, Stored Procedure and Wrapper business objects are always top level business objects, so they will not show up in this drop down list.

Click the “Set Parent” button and select the parent Object, “Customer” in this example, from the drop down list that contains all possible parent business objects. Now you must check to see if your selected parent business object, “Customer”, is a child, grand child or great grand child of the selected business object, “Address”. This is done by walking the business object hierarchy and determining if the parent business object, “Customer” is in the sub-hierarchy under “Address”. The screen now displays the list of attributes of each business object, “Customer and Address” in this example, in separate drop down lists.

Select the Single cardinality Check box, if you want to make “Address” a Single Cardinality child of “Customer”, otherwise it is a Multiple cardinality child of “Customer”. Select appropriate attributes in the child and parent objects, one at a time, that you want to map as Foreign Keys, and build up a list of these attribute pairs. Select Ownership, KeepRelationship and Required check boxes to set the appropriate ASI on the child business object element, that is to be created in the parent business object. When you click Next, the appropriate child element is created in the parent business object and foreign key references are set in the child or parent as required.

Business object: primary key

- Check for primary key on the selected database table
 - ▶ If the primary key is defined in the database table, the adapter will set the primary key ASI. And it will not ask user primary key selection.
 - ▶ If the primary key is not defined in the database table, the adapter will prompt user to select primary key column.



Before generating the business objects for objects of type Table or Views, the EMD process will check to see if the primary keys are defined on the selected database table. If the primary key is already defined on the database table, then EMD processing will automatically set the PrimaryKey ASI in the generated Table business object. However, if the primary key is not defined on the table, then you need to provide that information during the EMD process. Once you select the node of type Table and click “Add to import”, a new window, “Configure Parameters for EMPLOYEES” in this example, is shown.

This is where you can add one or more columns to be the primary keys for that business object. If you have checked the “Add business object ASI” check box from the Edit Query screen, then the primary key properties does not show a new window. Instead, the properties is part of the same screen that is used for building the hierarchy of business objects, associating stored procedures with tables or views and for defining the status column for logical deletes.

Section

Event management



This section provides an overview of the event management of the WebSphere Adapter for JDBC V6.2

Event management: standard event process

- Standard event processing using Event store (event table)
 - ▶ Triggers on user tables to record events
 - ▶ If “AssuredOnceDelivery” is set to true, an XID value is set and is updated in the event table
 - ▶ XID column ensures that the events is reprocessed in case of failure
 - ▶ Filter events to be processed by business objects type



Asynchronous event delivery is supported either by having an event table (event store) or by using custom queries. For standard event processing, the event table in the EIS is populated by the application for any changes that occur in your tables. It can be updated by placing triggers on your tables that record events in the event table, that correspond to updates to your table. If the ‘AssuredOnceDelivery’ property is set to true, an XID value is set for each event in the event store. After each event is picked up for processing, the XID value for that event is updated in the event table. Each event is then delivered to its corresponding endpoint and subsequently deleted from the event table. Before the event is delivered to the endpoint, the database connection might be lost or the application might be stopped, resulting in the event not being processed completely. The XID column ensures that the event is re-processed and sent to the endpoint. Once the database connection is re-established or the adapter starts up again, it first checks for events in the event table that have a value in the XID column. It then processes these events first and then polls the other events during the poll cycles.

The adapter can also filter the events to be processed by business object type, or timestamp. The filter is set in the EventFilterType property specified in the adapter foundation classes. This property has a comma-delimited list of business object types and only the types specified in this property is picked for processing. If no value is specified for the property, no filter is applied and all the events are picked up for processing. If the FilterFutureEvents property in the adapter foundation classes is set to true, the adapter will filter events based on their timestamp. The adapter will retrieve the system-time in each poll cycle and compare it to the timestamp on each event. If the event is set to occur in the future, it is not picked up for processing until that future time.

Event management: event filtering

- Adapter Instance
 - ▶ Configure in Activation Specification
 - ▶ String value for event filtering
 - ▶ Ignore if adapter instance for event filtering is not configured

Event delivery configuration

Type of delivery: ORDERED

Ensure assured-once event delivery (may reduce performance)

Do not process events that have a timestamp in the future

Event types to process:

Adapter Instance for event filtering:

Retry limit for failed events: 5

Number of connections for event delivery

Minimum: 1

Maximum: 1

With V6.2, the adapter can also filter the events to be processed by connector id. The purpose of this new feature is to enable seamless migration for WebSphere Business Integration Adapters (WBIA) to JCA where users are currently taking advantage of the connector ID filtering. It allows large users that are currently dependent on this in WBIA to more easily migrate. This is also a benefit to users to load balance when they have large number of events of the same type.

The JDBC adapter supports this filter through Activation Specification property and provides the filtering mechanism when a connector ID is configured. The adapter only retrieves events with that particular connector ID. It also ensures compatible with earlier versions when connector ID field is not specified. That allows the event table structure without the connector ID field to be supported. In addition, there is minimal impact for users who migrate from WBIA since the connector ID filtering has similar mechanism.

Event filtering

- Event Store
 - ▶ Additional column for connector ID
 - ▶ Sample script is updated to add ConnectorID field

Name	Description
Xid	▪ Unique XID (transaction id) value for assured once delivery
Event_id	▪ Unique Eventid which is a primary key
Object_key	▪ A string that contains keys of the record in the event store that is retrieved
Object_name	▪ Business graph name for WebSphere Process Server, and business object name for other brokers (WebSphere Application Server, WebSphere Message Broker, WTX, DataStage)
Object_function	▪ Operation corresponding to the event (delete, create, update and so on) (not null)
Connector_ID	▪ Filter by connector_ID value
Event_priority	▪ Valid values are any positive integers
Event_time	▪ Date/Time when event was generated
Event_status	(In Progress, Success, Failure) and so on. Values: 0 – Ready for poll 1 – delivered -1 – delivered
Event_comment	▪ Description

The event store is a persistent cache where event records are saved until the polling adapter can process them. The adapter uses the event store to keep track of inbound requests as they make their way through the system. Each time a database record is created, updated, or deleted, the adapter updates the status of the event in the event store. The status of each event is continually updated by the adapter for recovery purposes until the events are delivered to a configured export on the server.

With this new event filtering feature, it is configured in Activation Specification with any string values and required changes to the event store to add another column for connector ID. The adapter then adds the corresponding column to the event query. The adapter samples and scripts are also updated to add the connectorID field to the event table structure.

Event management: customized event process

- Customized event processing
 - ▶ Standard SQL statements
 - ▶ A stored procedure
 - ▶ A stored function
- Custom Query - Support for assured once delivery
 - ▶ Required standard event store
 - ▶ Store events returned by custom event query
 - ▶ Update the events with XID values and process the events

23

WebSphere Adapters for JDBC V6.2

© 2009 IBM Corporation

For customized event processing, you can either enter a standard SQL statement, a stored procedure or a stored function. All three cases will return a `ResultSet` that will have the data for these columns in this order: `event_id`, `object_key`, `object_name` and `object_function`. The adapter will construct a `Business` object for each event and then deliver it to the end points that have a subscription for this specific business object. The adapter will also support custom `Update` and `Delete` queries for custom event processing. The custom update and delete queries are started after each event is processed, and will use the event ID as an input parameter. The update query is used to ensure that the same record does not get picked up for processing during subsequent poll cycles. The delete query is used in case records need to be deleted after each event is delivered.

If you want to have support for assured-once delivery in the case of a custom query, you will need to have the standard event store created. That allows XID to be able to be stored in that event store. The adapter will store the events returned by the custom event query in the event store, update those events with XID values and process the events as it does in the case of standard event processing. In this type of scenario, the event table does not have auto-generated event ID values. Also, when using a custom query, there is no support for event filtering.

Section

Summary and references



This section provides a summary of the WebSphere Adapter for JDBC V6.2

Summary

- This presentation covered an overview of the new and enhanced IBM WebSphere Adapters for JDBC V6.2
 - ▶ Reviewed steps in development and deployment
 - Supported auto-generation in Informix database
 - ▶ Reviewed different business object models
 - ▶ Introduced new outbound operation
 - ▶ Introduced event filtering for inbound



This presentation covered an overview of the new and enhanced IBM WebSphere Adapter for JDBC V6.2 release. You have reviewed how to develop and deploy JDBC adapter with new enhanced for Informix database. You have also been introduced to many different business object models and learned about new outbound operation and event filtering enhancement for inbound.

A separate demonstration is available that illustrates the features and functions of the newly enhanced Enterprise Metadata Discovery process.

Reference information

- WebSphere adapter for JDBC user guide
- Java Connector Architecture
 - ▶ <http://java.sun.com/j2ee/connector/index.jsp>
- Enterprise Metadata Discovery
 - ▶ <http://www.ibm.com/developerworks/java/library/j-emd/>
- WebSphere Adapter information center
 - ▶ <http://www.ibm.com/software/integration/wbiadapters/library/infocenter/>
- WebSphere Process Integration information center
http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/topic/com.ibm.websphere.wps.620.doc/welcome_wps.html



Additional reference information can be found at these URLs.

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_WBIV62_IEA_AdapterJDBC.ppt

This module is also available in PDF format at: [../WBIV62_IEA_AdapterJDBC.pdf](..WBIV62_IEA_AdapterJDBC.pdf)



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

Trademarks, copyrights, and disclaimers

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

DB2 Informix WebSphere

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

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

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

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements or changes in the products or programs described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (for example, IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.

IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

© Copyright International Business Machines Corporation 2009. 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.