



IBM Software Group

WebSphere Business Integration V6.1

WebSphere Adapter for JDBC™ V6.1



@business on demand.

© 2008 IBM Corporation
Converted to video June 23, 2015

This presentation covers the WebSphere Adapter for JDBC™ V6.1

Agenda

- Overview
- Business object model
- Event management
- Configuration properties
- 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 some of new business object models for V6.1. The last section provides tables of configuration properties for reference. 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 2 Enterprise Edition 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.

Section

Business object model

This section provides an overview of the business object model.

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 Enterprise Metadata Discovery, or 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 objects are 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 and 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
 - ▶ Generated using input provided by user
 - ▶ Add multiple table business objects inside a wrapper business object

The screenshot shows a dialog box titled 'External Service' with a sub-tab 'Configure Objects'. The main heading is 'Specify the properties for the objects that will be imported by the discovery agent.' Below this, there are several sections:

- 'Operations for selected business objects': A list of operations (Create, Update, Delete, Retrieve, RetrieveAll, ApplyChanges) with 'Add...' and 'Remove' buttons.
- 'Create and configure user-defined wrapper objects': A section for 'Wrapper object names' with an 'Add...' button.
- 'Table, view, synonym, or nickname child business objects for the selected wrapper object': A section for 'ADDRESS:' with a 'CUSTOMER' entry and 'Add...' and 'Remove' buttons.
- 'Service functions for selected wrapper object': A section with 'Create', 'Delete', 'Retrieve', and 'Update' options and 'Add...' and 'Remove' buttons.
- 'Maximum records for RetrieveAll operation': A field set to '100'.
- 'Business object namespaces': A field containing the URL 'http://www.ibm.com/xmlns/prod/ibmsphere32ca/jdbc'.
- 'Specify the relative folder for generated business objects': An empty field.
- 'Folder': An empty field.
- A checkbox labeled 'Generate a business graph for each business object' which is checked.

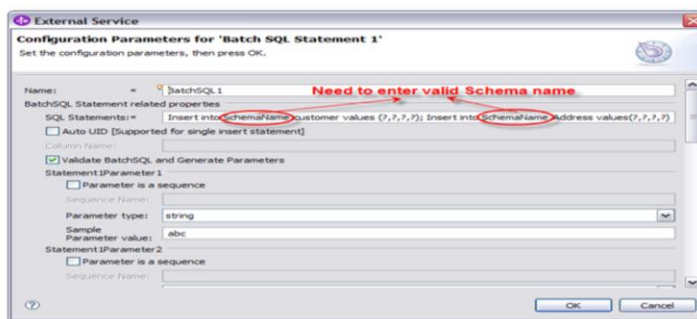
At the bottom, there are navigation buttons: '< Back', 'Next >', 'Finish', and 'Cancel'.

The fourth type of supported object is the Wrapper Business Object, which wraps together multiple Table Business objects. The JDBC EMD wizard will ask you to select multiple table business objects, and use them to generate a top level wrapper business object which has as its 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



10

WebSphere Adapter for JDBC V6.1

© 2008 IBM Corporation

New in this release is support 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 and Delete SQL statements, and the types and dummy values of all parameters 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 run 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 will find one top level node labeled "BatchSQL Statements", and if you expand this node you will 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: 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: Operations

- 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. Outbound operations also include Delta support for the ApplyChanges operation.

Business object: Operations

- **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**



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 now available for both Stored Procedures and for user defined SQL statements. In support of Stored procedures, the start operation will generate the required stored procedure that corresponds to the stored procedure definition in the database. For user defined SQL statements, this operation will generate the required BatchSQL business object based on the user defined SQL statement or statements.

Business object: Verbs

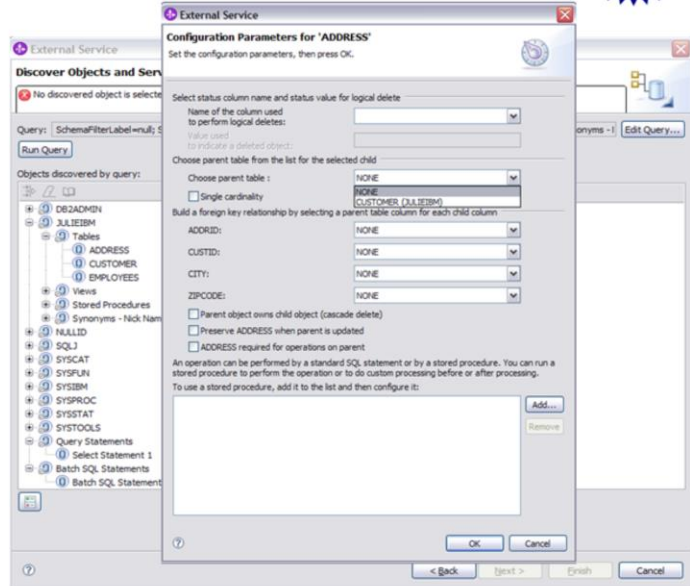
- Verbs - deprecated
 - ▶ Function name will determine the action
 - For example, emitCreateAfterImageCustomerBG will have action CREATE
 - ▶ For compatibility with earlier versions, UpdateWithDelete is still supported by the adapter, however, it is not supported by Enterprise Metadata Discovery.

Support for verbs has been deprecated in the V6.1 release of the WebSphere Adapter for JDBC. Verbs are no longer being used, and if encountered they are ignored. Now the Function name, emitCreateAfterImageCustomerBG for example, will determine the action, Create in this case. For compatibility with earlier versions, UpdateWithDelete is still being supported by base classes, but it is no longer supported by the EMD process.

Business object: Hierarchy

Enhanced
V6.0.2.2

- Now allows you to build a hierarchy of Parent-Child business objects
- A new mechanism for building using Enterprise Metadata Discovery



15

WebSphere Adapter for JDBC V6.1

© 2008 IBM Corporation

The earlier releases of the adapter did not allow you to build a hierarchy of Parent-Child business objects using the EMD process. You had to use the business object editor to build this hierarchy. This version of the adapter provides a mechanism to do the same thing by using the EMD process. This new mechanism can be invoked at a separate screen during the 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 business objects and Wrapper business objects are always top level business objects, so they do not show up in this drop down list.

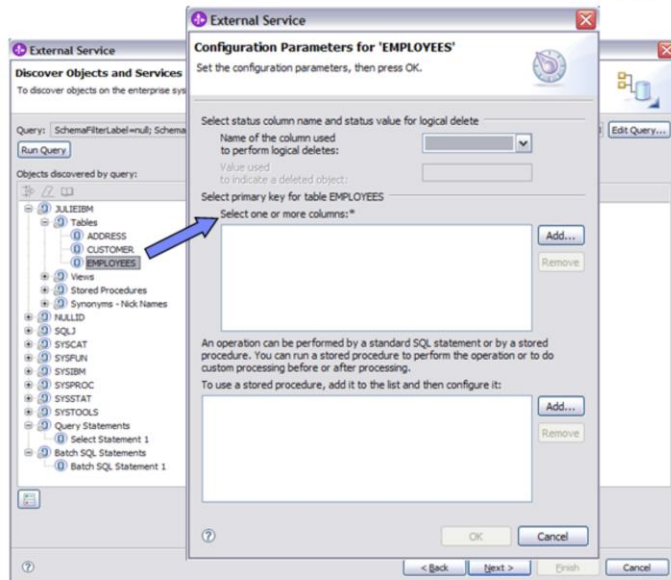
Click “Set Parent” and select the parent object, “Customer” in this example, from the drop down list that contains all possible parent business objects. Now you must check to see if your selected parent business object, “Customer”, is a child, grand child or great grand child of the selected business 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. Click on the Single cardinality check box, if you want to make “Address” a single cardinality child of “Customer”, otherwise it will be 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. Click on the 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

New
V6.1

- 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.



WebSphere Adapter for JDBC V6.1

© 2008 IBM Corporation

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 is shown; in this example, it is “Configure Parameters for EMPLOYEES.” 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 are not shown in a new window. Instead, the properties are 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.



Business object: application specific information

- CopyAttribute feature
 - ▶ The value of one attribute is copied to value of another attribute
 - ▶ Specified by CopyAttribute ASI
 - ▶ Same business object level or at the immediate parent business object level
- Placeholders feature
 - ▶ Attributes that are not associated with any database entity
 - Examples of placeholders
 - Parameters
 - jdbcWhereClause in a Query business object
- Support for UNVL
 - ▶ Allow to set column value in the database to null
 - ▶ Retrieve rows from database by searching on null values



In outbound operations, the adapter supports attribute copying, where the value of one attribute is copied to the value of another attribute specified by the CopyAttribute ASI. The attribute can be at the same business object level or at the immediate parent business object level. **CopyAttribute** is basically a string you specify which refers to another attribute name from within the same business object or refers to another attribute name in the parent business object. If the value set in the ASI refers to the name of another attribute within the same business object, the adapter uses the value of the other attribute to set the value of this attribute. This is done before it adds the business object to the database during Create operations.

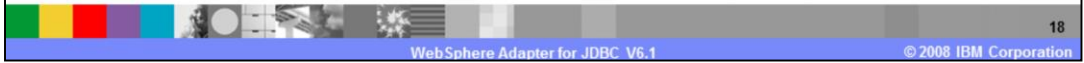
The value set in the ASI cannot reference an attribute in a child business object, but it can reference an attribute in the parent business object if it is preceded by two periods. If you do not include this parameter in the application-specific information, the adapter uses the value of the current attribute without copying the CopyAttribute value from another attribute. This support is also only applicable to simple attributes.

Placeholders are attributes that are not associated with any database entity. Examples of placeholders are parameters and the jdbcWhereClause in a Query business object.

The JDBC adapter now includes support for the NULL value. In the previous WebSphere Business Integration Adapter, there was no way to determine if the value of a particular attribute was set or unset. As a result, that adapter used the UNVL ASI on the attribute to be able to set a null value on that column in the database, or to retrieve a row using the “**is null**” operator in the “where” clause. In the case of SDOs, the isSet function returns whether a value has been set on that attribute or not. The JDBC Adapter now uses this information to be able to update columns with a NULL value. The UNVL ASI is not needed for the JCA JDBC Adapter, but the adapter has been modified to use this information to also generate the appropriate “where” clause, using the “is null” operator. This will allow proper Update and Retrieve NULLI values to be passed to and from the Database.

Section

Event management



This section provides an overview of the event management of the WebSphere Adapter for JDBC V6.1

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.

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)
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

19

WebSphere Adapter for JDBC V6.1

© 2008 IBM Corporation

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 may 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 event table is shown on the right side of this page.

The adapter can also filter the events to be processed by business object type. The filter is set in the EventFilterType property specified in the adapter foundation classes. This property will have a comma-delimited list of business object types and only the types specified in this property will be 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. This feature is implemented in the adapter foundation classes. 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: Customized event process

- Customized event processing
 - ▶ Standard SQL, Stored Procedure, Stored Function
 - ▶ No support for event filtering
- 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

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, 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, in order for xids 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 will not have auto-generated event ID values. Also, when using a custom query, there is no support for event filtering.

Section

Configuration properties

This section provides details of the configuration properties of the WebSphere Adapter for JDBC V6.1.

Required application sign-on for inbound and outbound

Property	Description
UserName	<ul style="list-style-type: none">User Name to connect to the database
Password	<ul style="list-style-type: none">Associated password for User Name
Database URL	<ul style="list-style-type: none">Database URL used to connect to the database
AutoCommit	<ul style="list-style-type: none">Boolean value for AutoCommit to be set on the connection
JDBCClass	<ul style="list-style-type: none">JDBC Driver class for the driver used to connect to the database
PingQuery	<ul style="list-style-type: none">SQL Query to test valid connection to the database
EventTableName	<ul style="list-style-type: none">Table in EIS that contains events generated by EIS for Inbound processing.
DriverConnectionProperties	<ul style="list-style-type: none">Additional properties for connecting to database using JDBC Driver, besides Username and Password. Must be specified as Name-Value pairs separated by semi-colons.Example: <code>loginTimeout:20; readOnly:true; securityMechanism:USER_ONLY_SECURITY</code>

Shown here is a list of connection properties for the Enterprise Metadata Discovery process, which are required for both Inbound and Outbound services. These properties are fully defined in the Adapter User's Guide and the Information Center, and are largely the same as the previous version of the adapter. One additional property for V6.1 is the DriverConnectionProperties property. It provides additional properties for connecting to a database using the JDBC driver, beyond just user name and password. A sample value for this property for a DB2 database is shown here.

Activation specification properties (inbound)

Property	Description
Username	<ul style="list-style-type: none"> ▪ User name for logging into the EIS for inbound events
Password	<ul style="list-style-type: none"> ▪ Password for authorizing user to retrieve events from the EIS
JDBCDriverClass	<ul style="list-style-type: none"> ▪ The driver class used to connect to the EIS
DatabaseURL	<ul style="list-style-type: none"> ▪ The driver-specific URL for creating a connection to the EIS
EventOrderBy	<ul style="list-style-type: none"> ▪ The order in which events is retrieved and processed.
EventTableName	<ul style="list-style-type: none"> ▪ The name of the event table from where events are picking up by JDBC Adapter during Inbound processing
SPBeforePoll	<ul style="list-style-type: none"> ▪ This is for any stored procedure that user wants to run before the actual poll query is called. It will take one input parameter for poll quantity.
SPAAfterPoll	<ul style="list-style-type: none"> ▪ This is for any stored procedure that user wants to run after each poll cycle. It will take one input parameter for poll quantity.

Shown here are highlights of some of the properties used for inbound processing that are stored in the activation specification. These properties hold the inbound event processing configuration information for a message endpoint. They can be set through either the EMD wizard or through the WebSphere Process Server or WebSphere Enterprise Service Bus administrative console. For more information on these properties, refer to the Adapter User's Guide and the Information Center.

Activation specification properties (inbound)

Property	Description
EventQueryType	<ul style="list-style-type: none">This property values will decide whether to use standard event store or custom query. The valid values are Standard (for the standard event store) and Dynamic (for custom event processing)
CustomEventQuery	<ul style="list-style-type: none">The SQL query, stored procedure or stored function for custom event processing. This runs during each poll cycle when the EventQueryType is set to Dynamic.
CustomUpdateQuery	<ul style="list-style-type: none">The custom update query that is started after each event is processed so that the same event does not get picked up for processing in the subsequent event cycles.
CustomDeleteQuery	<ul style="list-style-type: none">The custom delete query that is started after each event is processed.
DataSourceJNDIName	<ul style="list-style-type: none">This is the JNDI data source name to be used by the adapter to establish connection to the database.

Additional inbound properties can be added to support user-customizable event processing. These include properties for a custom event query, a custom update query, a custom delete query, a Stored Procedure before and after polling and an event query type property. They determine if a custom query or a standard event store should be used for event processing.

More information on the complete list of activation specification properties can be found in the WebSphere Adapter Information Center. The link to the Information Center is provided at the end of this presentation.

Activation specification properties

Property	Description
DriverConnectionProperties	<ul style="list-style-type: none"> Additional properties for connecting to database using JDBC Driver, besides Username and Password. Have to be specified as Name-Value pairs separated by semi-colons.
DatabaseVendor	<ul style="list-style-type: none"> Specifies which RDBMS the RA uses for special processing.
QueryTimeOut	<ul style="list-style-type: none"> The value of this property is an integer in seconds that sets the QueryTimeOut for all SQL Statements to the number of seconds specified.
Ping Query	<ul style="list-style-type: none"> SQL Query to test valid connection to the database
MimimumConnections	<ul style="list-style-type: none"> Specifies the number of initial connections to the EIS to use for inbound event delivery. Any value less than 1 is treated as 1.
MaximumConnections	<ul style="list-style-type: none"> Specifies the maximum number of connections to the EIS to use for inbound event delivery. Any value less than 1 is treated as 1.

Here are highlights of some of the properties used for activation specification, that are specific to the foundation classes. For QueryTimeOut, the Adapter EMD will display the unit of measure in the label for the property wherever applicable. New in the Activation Specification is connection pool information. This connection pooling is necessary to support the new threading model in the event manager.

Managed connection factory properties (outbound)

Property	Description
UserName	<ul style="list-style-type: none"> User name to connect to the EIS
Password	<ul style="list-style-type: none"> Associated password for user name
DatabaseURL	<ul style="list-style-type: none"> Database URL used to connect to the database
DataSourceJNDIName	<ul style="list-style-type: none"> JNDI name of the data source used to get the JDBC connection
AutoCommit	<ul style="list-style-type: none"> Auto commit to be set on the connection
JDBCDriverClass	<ul style="list-style-type: none"> JDBC Driver class for the driver used to connect to the database
XADatasourceName	<ul style="list-style-type: none"> XA data source name to be used to establish an XA connection to the database
XADatabaseName	<ul style="list-style-type: none"> Database name to be used for the XA connection
DriverConnectionProperties	<ul style="list-style-type: none"> Additional properties for connecting to database using JDBC Driver, besides Username and Password. Have to be specified as Name-Value pairs separated by semi-colons.
DatabaseVendor	<ul style="list-style-type: none"> Specifies which RDBMS the RA uses for special processing.
QueryTimeout	<ul style="list-style-type: none"> The value of this property is an integer in seconds that sets the Query Time Out for all SQL Statements to the number of seconds specified.
ReturnDummyBOForSP	<ul style="list-style-type: none"> This property is used to return output parameters even when the result set is empty.
PingQuery	<ul style="list-style-type: none"> SQL Query to test valid connection to the database

26

WebSphere Adapter for JDBC V6.1

© 2008 IBM Corporation

Here are highlights of some of the properties used for outbound processing that are stored in the managed connection factory. The properties used in connection establishment are used in a particular order as follows:

If the DataSourceJNDIName property is set, then it is used to establish the connection to the database.

If the UserName and Password properties are set along with DataSourceJNDIName then they override user name and password set on the data source.

If the DataSourceJNDIName property is not set and the XADatasourceName and XADatabaseName properties are set, then they are used to establish the connection.

If DataSourceJNDIName, XADatasourceName and XADatabaseName are not set then the regular DatabaseURL, DriverClassName, UserName and Password properties are used to establish the connection.

An “advanced properties” column has also been added to the managed connection factory. The DatabaseVendor property value should be set to DB2, Oracle or MSSQLServer, depending on which of these databases you are using. If you are using a different database, select Other. If you are using a non-default database, you must also ensure that the proper driver is present in the class path. A value for this property is required for the adapter to connect and process successfully; no default value is provided. If you happen to specify an incorrect value for this property, then some of the database specific features (for example, Struct and Array for an Oracle database) will likely not work correctly.

For the QueryTimeout property, if the value is not specified, it implies that no timeout is set on the query. If the query takes longer than the number of seconds specified in this property, the database will produce an SQL Exception that is captured, and the associated message is logged in the log file.

Section

Summary and references



This section provides a summary of the WebSphere Adapter for JDBC V6.1

Summary

- This presentation covered an overview of the new and enhanced IBM WebSphere Adapters for JDBC V6.1
 - ▶ Introduced new business object models
 - ▶ Differentiated between standard and customized event processing
 - ▶ Introduced new configuration properties

This presentation covered an overview of the new and enhanced IBM WebSphere Adapter for JDBC V6.1 release. You have been introduced to several different business object models and learned about the new features and enhancements of this release. You also learned about some of the new configuration properties for the JDBC adapter.

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-306.ibm.com/software/integration/wbiadapters/library/infocenter/>
- WebSphere Process Integration Information Center
 - ▶ http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/index.jsp?topic=/com.ibm.websphere.wps.610.doc/welcome_top_wps.htm

Additional reference information can be found at these Web addresses.

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

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

DataStage, DataStage, is a trademark of Ascential Software Corporation 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 NON-INFRINGEMENT. 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.