



IBM Software Group

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

IBM WebSphere® Adapter for JDBC™ V6.0



@business on demand.

© 2005 IBM Corporation
Updated November 18, 2005

This presentation will cover the WebSphere Adapter for JDBC V6.0.

Agenda

- **Overview**
- Architecture
- Enterprise Service Discovery
- Object Model
- Event Manager
- Summary

This section will provide an overview of the WebSphere Adapter for JDBC.

Overview: WebSphere Adapter for JDBC

- IBM WebSphere Adapter for JDBC implements the Java™2 Enterprise Edition (J2EE) Connector Architecture (JCA), version 1.5 specification
- Enables bi-directional connectivity for integration with database applications
 - ▶ Requires database with JDBC driver that supports the JDBC 2.0 or higher specification
- Uses standard JDBC APIs to retrieve and update information in the database
 - ▶ Executes SQL statements or Stored Procedures
 - ▶ Specified in the Business Object
- Supports both Local Transactions and Global (XA) as supported by the database
 - ▶ XA support is currently for outbound connections with DB2® or Oracle



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 driver supports the JDBC 2.0 or higher specification and the adapter uses JDBC APIs to retrieve and update information in the database by running SQL statements or stored procedures, depending on what you specify. This is stored and specified in the business object. A stored procedure is a group of SQL statements that form a logical unit and perform a particular task. A stored procedure encapsulates a set of operations or queries for the adapter to run on an object in a database server. The JDBC Adapter supports both local transactions and global XA transactions. This version of the adapter provides XA support for outbound connections with DB2 or Oracle databases.

Overview: Installation Preparation Steps

- WebSphere Adapter for JDBC Installer is used to install adapter to file system
- JDBC 2.0 Driver dependent jar file for Database
 - ▶ Dependent jars need to be in the classpath of the WebSphere Integration Developer project after importing the Resource Adapter Archive (RAR) file
- Event Store table – need to create
 - ▶ Required for inbound processing
 - ▶ Sample scripts provided for DB2, Oracle, SQL Server
 - ▶ When data changes in application tables, appropriate events are inserted into the event store
 - Use triggers defined on respective tables, or other means to capture changed data, such as Oracle Change Data Capture for Oracle databases

In preparation to install and use the WebSphere Adapter for JDBC within your business integration application, you will first use the installer to install the adapter to the file system. JCA adapters are packaged as resource adapter archive (RAR) files, and the installer copies the WebSphere Adapter for JDBC RAR file to the location you specify during installation. JDBC 2.0 driver specific to the database to which you will be integrating with must be added to the class path. Other information necessary in preparation for use of the adapter are the databases and tables used by the adapter. The event table, also referred to as the event store, is required by the adapter for inbound processing. You must create this table and specify the database, table, and driver used to connect. As data changes in the particular user application table, appropriate events are inserted into the event store table through the use of triggers you have created and defined on the application tables or by whatever other means your database supports to capture changed data. For example, with Oracle, the Oracle Change Data Capture can be used.

Overview: Installation Preparation Steps

- Event Distribution Table (Staging Table)
 - ▶ EDT database created manually
 - ▶ EDT table created automatically via use of *autocreateEDT* checkbox during Enterprise Service Discovery
 - ▶ If EDT database and table exist
 - Once and only once event delivery
 - Recovery in case of server failure

Optionally, the event distribution table (EDT), also referred to as the staging table, can be used by the adapter. The event distribution table and event distribution table database are required for guaranteed once and only once inbound delivery of events and also for recovery in the case of a server failure. You must manually create the EDT database, specify the EDT database name, and check the *autocreateEDT* checkbox during the enterprise service discovery phase of adapter configuration in WebSphere Integration Developer. If you do not choose to create a persistent EDT database, inbound processing can still occur. However, there is no guarantee of delivery and no recovery of events. This can be useful for initial application proof of concept testing.

Overview: Deployment

- **WebSphere Integration Developer**
 - ▶ Import JDBC Adapter Resource Archive file (CWYBC_JDBC.rar) into WebSphere Integration Developer
 - Creates a J2EE connector project in the workspace
 - Add any external dependencies to the project
 - ▶ Run Enterprise Service Discovery
 - Enter information necessary to configure the adapter for the first time (database connection information, service description, enter properties used in Activation Specification or Managed Connection Factory)
 - Complete object selection
 - Output saved in a Business Integration module
 - ▶ Complete Service Component Architecture (SCA) application assembly
 - ▶ Test in WebSphere Test Environment and Export as Enterprise Archive file (EAR) to be installed to WebSphere Process Server



Once you have installed the adapter to the file system, you will proceed with WebSphere Integration Developer and import the JDBC Adapter RAR file, CWYBC_JDBC.rar, into Integration Developer. This creates a J2EE connector project in your workspace. Add any external dependencies to the project. For example, the database device driver used to connect to the user application database, as well as the database device driver if using a persistent EDT database and table for inbound processing. Continue with the enterprise discovery service process to enter adapter configuration information, service type of inbound or outbound, selected operations and other adapter specific properties, such as logging and tracing file names, inbound and archive subdirectory names, and other customizable properties. Upon completion of the enterprise service discovery, you will be presented with the business integration perspective and a module containing the necessary adapter artifacts for integration and assembly with other Service Component Architecture (SCA) components. When application assembly is complete, you can test your application within the WebSphere Test Environment and ultimately export the enterprise archive file (EAR) to be installed to the WebSphere Process Server runtime.

Overview: Deployment

- **WebSphere Process Server**
 - ▶ Use Administrative Console or command line tool wsadmin
 - ▶ Create J2C authentication alias specifying userid / password to database
 - This needs to be created before testing or installing EARs for outbound requests
 - ▶ Install new Enterprise Application (the EAR just exported from WebSphere Integration Developer)
 - For the most part can simply accept defaults, however on the “Map resource references to resources” panel, need to select the authentication alias and apply
 - 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 similar to installing any other enterprise application. For outbound type communications with the JDBC adapter, there is the J2C authentication alias that can be used, containing a userid and password used to connect to the application database.

Overview: Deployment

- Before starting the application
 - ▶ Optionally, edit production time values for properties
 - Managed Connection Factory
 - Activation Specification
 - Adapter Custom
 - Deployment Descriptor (ra.xml)
 - ▶ From Enterprise Application panel, select
 - Your application -> Connector Modules -> RAR (CWYBC_JDBC.rar)

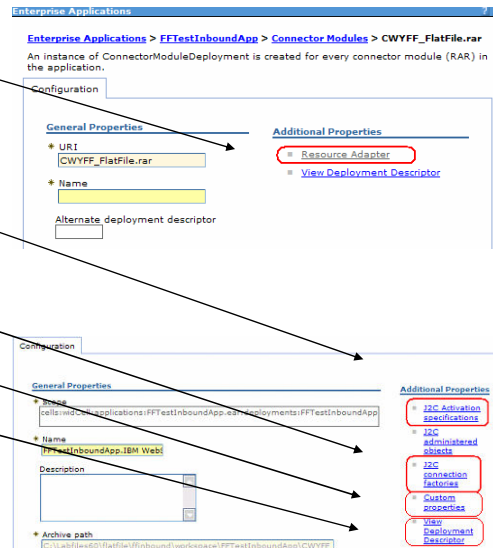
The screenshot displays the configuration page for the Enterprise Application 'FFTestInboundApp'. The page is divided into several sections:

- General Properties:** Name: FFTestInboundApp.
- Binary Management:** Application binaries: C:\Labfiles60\flatfile\ffinbou. Includes checkboxes for 'Use metadata from binaries' (checked) and 'Enable distribution' (unchecked).
- Validation:** warn.
- Class Loading and File Update Detection:**
 - Class loader mode: Parent First.
 - WAR class loader policy: Module.
 - Enable class reloading: checked.
 - Reloading interval: 5.
- Startup Options:** Starting weight: 1.
- Additional Properties:** A list of links including Session management, Application profiles, Libraries, Target mappings, Last participant support extension, View Deployment Descriptor, Provide JMS and EJB endpoint URL information, Publish WSDL files, Provide HTTP endpoint URL information, Provide JNDI Names for Beans, Provide listener bindings for message-driven beans, Map virtual hosts for Web modules, and Map modules to servers.
- Related Items:** Web modules, EJB Modules, and Connector Modules (highlighted with a red box).

Once the application is installed, the administrator can optionally choose to edit some of the property values. This can involve editing managed connection factory, activation specification, or adapter custom properties, or viewing the deployment descriptor properties. Editing can be done using the WebSphere Process Server administrative console, by selecting the enterprise application, the Connector Modules link, and then the RAR file, CWYBC_JDBC.rar.

Overview: Deployment

- ▶ Resource Adapter
 - Custom Properties to set
 - ▶ J2C Activation Specification
 - Select message listener, custom properties and edit values as appropriate
 - ▶ J2C Connection Factories
 - Select factory instance for your application, custom properties and edit values as appropriate
 - ▶ Custom Properties
 - ▶ View Deployment Descriptor
- Save the updates
 - Start the Enterprise Application



From the configuration tab of the CWYBC_JDBC.rar, you can select links for Resource Adapter, J2C Activation Specification, J2C Connection Factories, Custom Properties, or View Deployment Descriptor. Make any necessary updates, save the updates, and then start the enterprise application to make it available for use.

Configuration Properties

- Configuration Properties used by the adapter to communicate with the specific database application

- Types of Properties
 - ▶ Application sign-on
 - Required properties
 - ▶ Activation Specification
 - ▶ Connection Specification
 - ▶ Connection Factory
 - ▶ Custom



Configuration properties are used to communicate with the specific database application. There are various types of properties used such as application sign on, activation specification, connection specification, connection factory, and custom properties.

Application Sign-On (Required)

Property	Description
UserName	▪ UserName to connect to the database
Password	▪ Associated password for UserName
DatabaseURL	▪ DatabaseURL used to connect to the database
AutoCommit	▪ Autocommit to be set on the connection
JDBCDriverClass	▪ JDBC Driver class for the driver used to connect to the database
PingQuery	▪ SQL Query to test valid connection to the database
EventTableName	▪ Table in database that contains events generated for inbound processing

Shown here is the application sign-on type properties, which are defined in the adapter user guide.

Activation Specification Properties

Property	Description
Username	<ul style="list-style-type: none">Username to connect to the database for inbound events
Password	<ul style="list-style-type: none">Password for username to retrieve events
JDBCDriverClass	<ul style="list-style-type: none">Driver class used to connect
DatabaseURL	<ul style="list-style-type: none">Driver-specific URL for creating a connection
EventOrderBy	<ul style="list-style-type: none">Order in which events will be retrieved and processed. Expected values are comma separated names of the event table.

Shown here are highlights of some of the properties used for inbound processing that are stored in the activation specification.

Activation Specification Properties for Event Distribution Table for Inbound Events

Property	Description
EDTDriverName	▪ XA Database driver to use to connect to the Event Distribution Table
EDTDatabaseName	▪ Database name for Event Distribution Table
EDTTableName	▪ Table Name
EDTUserName	▪ User Name to connect to the EDT
EDTUserPassword	▪ User Password for the User name
EDTSchemaName	▪ Schema Name
EDTURL	▪ Database URL
EDTServerName	▪ Server Name
EDTPortNumber	▪ Port Number

Here are the event distribution database and table properties used for inbound processing that are stored in the activation specification. The important thing to note here is that the database driver used for the EDT table must be capable of supporting XA transactions.

Managed Connection Factory Properties

Property	Description
UserName	▪ UserName to connect to the database
Password	▪ Associated password for UserName
DatabaseURL	▪ DatabaseURL used to connect to the database
AutoCommit	▪ Autocommit to be set on the connection
JDBCDriverClass	▪ JDBC Driver class for the driver used to connect to the database
PingQuery	▪ SQL Query to test valid connection to the database
XADataSourceName	▪ XA Datasource name to be used to establish an XA connection to the database
XADatabaseName	▪ Database name to be used for the XA connection

Here are highlights of some of the properties used for outbound processing that are stored in the managed connection factory.

Deployment Descriptor Custom Properties

- Many properties in the deployment descriptor (ra.xml)
- These are just the custom properties

Property	Description
DatabaseVendor	<ul style="list-style-type: none">▪ Specifies which RDBMS the adapter uses for special processing (DB2, Oracle, or SQL Server, Other for Cloudscape)
QueryTimeOut	<ul style="list-style-type: none">▪ Sets the QueryTimeOut for all SQL statements to the number of seconds specified. SQL exception is captured for timeout
ReturnDummyBOForSP	<ul style="list-style-type: none">▪ Used to return output parameters even when the result set is empty (Dummy business object with values from output/input parameters will be returned)

Many properties are stored in the deployment descriptor. Here are highlights of some of those custom properties specific to the WebSphere Adapter for JDBC.

Overview: Advanced Features of JDBC Adapter

- UID Support (auto generated column)
 - ▶ Sequences and Identity Columns
- Order By
- Fixed Length Strings
- Stored Procedures
- Query Timeout



The adapter supports a set of advanced features listed here. An automatically generated column (UID) can be tied to a sequence or be defined as an identity column (numeric value). Sequences can be defined for DB2 and Oracle only and identity columns can be defined for SQL Server and DB2 databases. Ascending or descending order can be specified using the Order By feature. The adapter also supports processing with fixed length strings. If the attribute is of type fixed length when columns are CHAR, the adapter will pad the field with blanks. Support for stored procedures and query timeout complete the list of advanced features.

Agenda

- Overview
- **Architecture**
- Enterprise Service Discovery
- Object Model
- Event Manager
- Summary

This section will provide an overview of the architecture of the adapter.

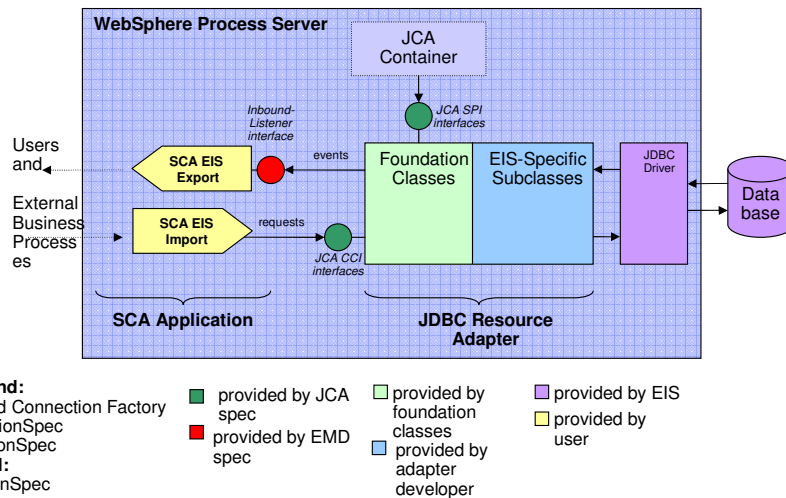
Overview: WebSphere Adapter for JDBC

- Adapter supports integration by
 - ▶ processing requests received from Service Component Architecture (SCA) applications (outbound)
 - requesting data from or making updates to database applications
 - ▶ processing events generated as a result of updates to the database application (inbound)
 - Via event notifications posted in an Event Store table
 - Events transmitted to predefined endpoint in the WebSphere Process Server



The WebSphere Adapter for JDBC supports integration by processing requests received from SCA applications outbound to Enterprise Information Systems requesting data from or making updates to the database application, returning responses as necessary. It also processes events inbound from Enterprise Information Systems as a result of any updates or changes to the database application done through event notifications posted to the event store table, and delivering those events to a predefined endpoint in WebSphere Process Server.

WebSphere Adapter for JDBC Architecture (JCA)



19

IBM WebSphere Adapter for JDBC

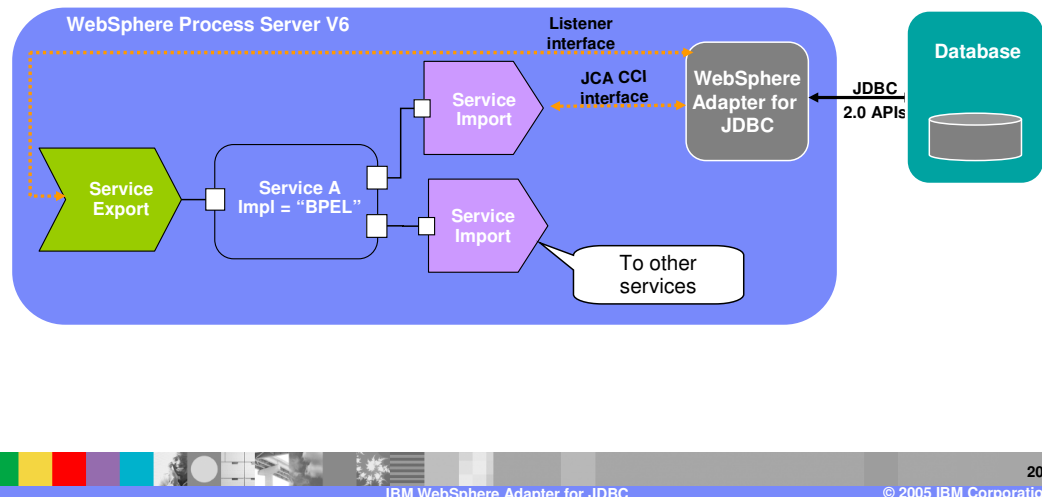
© 2005 IBM Corporation

The two main interfaces to a JCA adapter are the Service Provider Interface (SPI) and the Common Client Interface (CCI). The SPI is the application server view of the adapter. It contains the contracts necessary to work well with an application server, for example, connection creation, matching and management, security, transactions, and work management. The CCI is designed to provide a common view of data and interaction with the adapter. The CCI defines the data model and provides a common mechanism to interact with the adapter. The JDBC Resource Adapter updates the database tables using SQL queries or Stored Procedures as specified in the Business Objects.

This graphic shows the lower level architecture of various components that play a role in the end-to-end invocation of the outbound or inbound request. Using the Enterprise Service Discovery wizard in WebSphere Integration Developer, the Service Component Architecture (SCA) artifacts and the associated business objects are created. For the outbound request, an SCA Enterprise Information System (EIS) export is created, and for the inbound request, an SCA EIS import is created. The SCA clients interact with the SCA EIS export and import components to drive an outbound request or receive an inbound request, as shown in the diagram. The adapter contains the implementation of the JCA specification and has extensions provided by the adapter foundation classes. The SCA export component passes a business object wrapped in a JCA Common Client Interface (CCI) Record object. The adapter extracts the business object from the Record object and determines the operation to call along with its arguments. The adapter uses the JDBC device driver to communication with the application database.

WebSphere Adapter for JDBC Architecture (SCA)

- EIS Import and EIS Export provide Service Component Architecture (SCA) components with the uniform view of the services external to the module
- This allows components to communicate with the variety of external EIS systems using consistent SCA programming model



This diagram shows the higher level programming model supported by WebSphere Integration Developer and WebSphere Process Server. WebSphere Adapters integrate with other business integration applications using similar artifacts such as other SCA components. To support integration, the resource adapter processes requests received from SCA applications outbound to database applications. It also processes events generated as a result of updates to data in the database applications, inbound to the adapter. The adapter transmits these events to various predefined endpoints in the server. *Endpoints* are SCA applications or other client consumers of the event. Here you see a business integration application, Service A, implemented as a BPEL process, making its interface available to other services using the EIS Service Export and WebSphere Adapter for JDBC. Service A also references, or invokes other services using the EIS Service Import and WebSphere Adapter for JDBC, or other SCA modules using a Service Import with the appropriate binding, such as SCA, JMS, Web Service, or a Stateless Session Bean.

Agenda

- Overview
- Architecture
- **Enterprise Service Discovery**
- Object Model
- Event Manager
- Summary

This section will provide an overview of enterprise service discovery.

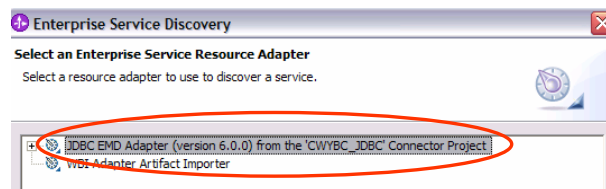
Enterprise Service Discovery

- Discovery of objects in a database
 - ▶ Objects can be created from tables, views, stored procedures, synonyms/nicknames
 - ▶ Generate a business object corresponding to a database object
 - ▶ Generate properties in the business object corresponding to properties in the database object
 - ▶ Set Application Specific Information on the business object
 - ▶ Provide Service Descriptions (inbound and outbound) used to generate the SCA-related files
 - Imports, Exports, and WSDL files

The Enterprise Service Discovery support in the JDBC adapter allows for the discovery of objects in a database, generation of Business Objects from the selected objects, and generates the service constructs that enable the adapter to run as an SCA component. Objects from which business objects can be created include tables, views, stored procedures, and synonyms/nicknames.

Enterprise Service Discovery

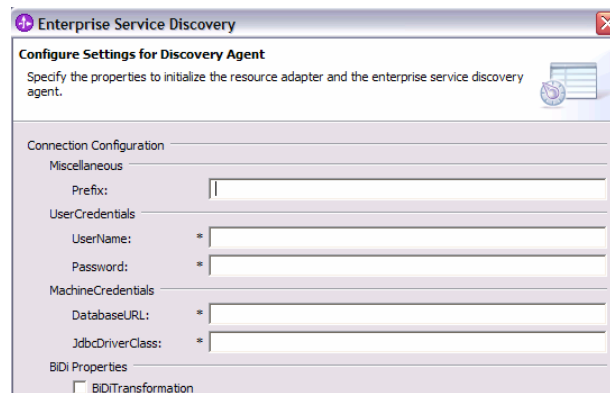
- File > New > Enterprise Service Discovery
- Select the Adapter



To start navigating through the enterprise service discovery wizard in WebSphere Integration Developer you will select File > New > Enterprise Service Discovery from the top level menu and then select the WebSphere Adapter for JDBC.

Enterprise Service Discovery

- Configure Settings for Discovery Agent
 - ▶ Properties: UserName, Password, Database URL, JDBCDriverClass, Prefix



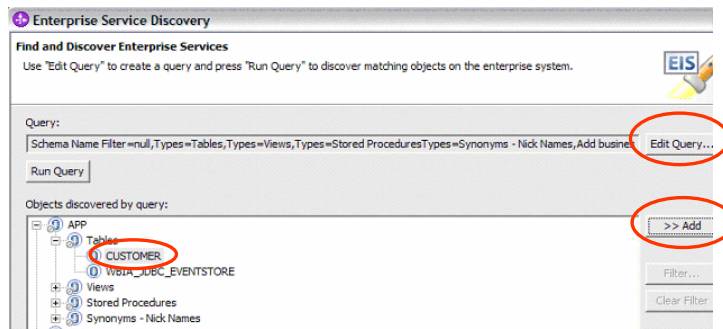
The screenshot shows a window titled "Enterprise Service Discovery" with a close button in the top right corner. Below the title bar is a section titled "Configure Settings for Discovery Agent" with a subtitle: "Specify the properties to initialize the resource adapter and the enterprise service discovery agent." The main area is divided into several sections: "Miscellaneous" with a "Prefix:" label and an empty text box; "UserCredentials" with "UserName:" and "Password:" labels, each followed by an asterisk and an empty text box; "MachineCredentials" with "DatabaseURL:" and "JdbcDriverClass:" labels, each followed by an asterisk and an empty text box; and "BIDI Properties" with a "BIDITransformation" checkbox that is currently unchecked.

Configure settings for the discovery agent. Metadata Discovery Agent Configuration Setting required properties are UserName, Password, Database URL, and JDBCDriverClass.

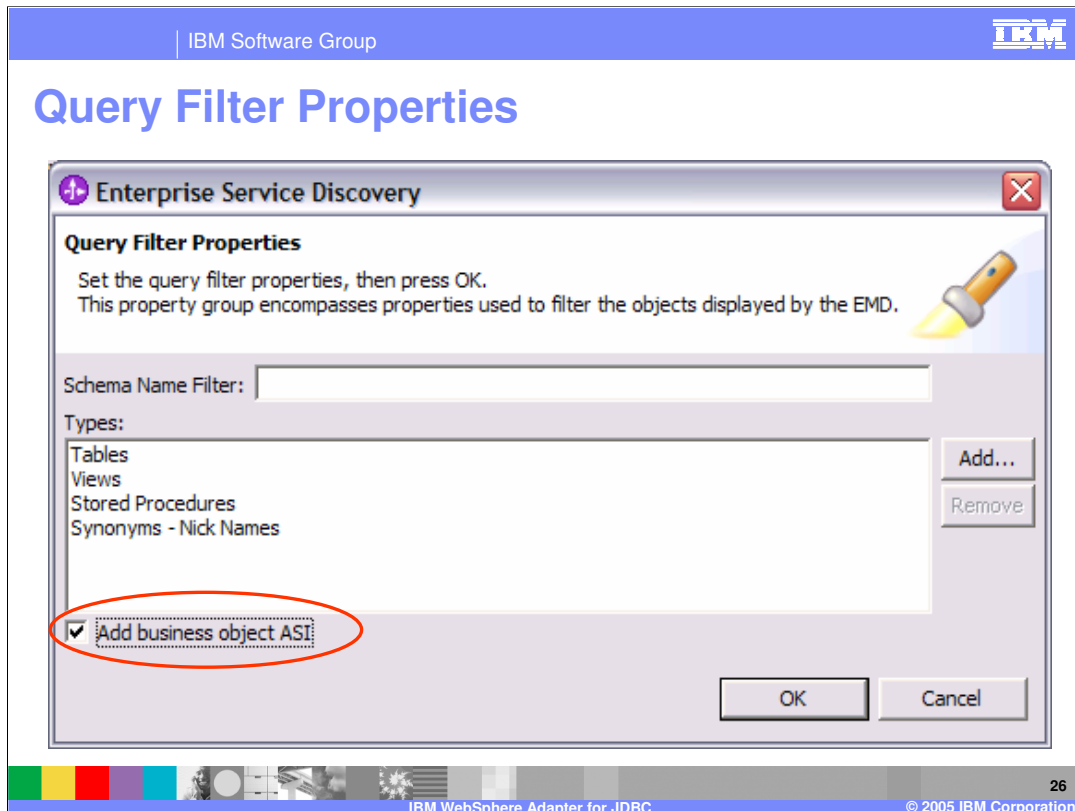
Metadata Discovery System Capabilities Schemas will be displayed as top-level nodes in a tree. Nodes, labeled Tables, Views, Stored Procedures, and Synonyms/Nicknames for that schema are found under each schema. These nodes are selectable. Before displaying the tree, the SchemaNameFilter property will be used to filter the list of schemas displayed. If the SchemaNameFilter property is not set, all schemas will be displayed. The Types property will then be used to determine which type nodes to add under the schemas listed. Upon expanding a node, the ObjectNameFilter property will be used to determine which database objects to display.

Enterprise Service Discovery

- Metadata Discovery System Capabilities
 - ▶ Analyze database to identify schemas
 - ▶ Set filter properties to limit number of objects displayed and whether to set ASI values for the business objects



Edit query can be selected to set query filters based on schema name, type, and to check the “add Business object ASI” box to select stored procedures.



From the query filter properties panel you can filter on schema name or types and optionally select the add business object ASI checkbox to specify stored procedures.

Business Object Application Specific Information

- **StatusColumnName**
 - Name of the database column used to perform logical deletes
- **StatusValue**
 - Value that specifies whether a business object is inactive or deleted
- **StoredProcedureName**
 - Associate with the Stored Procedure Type

Enterprise Service Discovery

Configuration Parameters for CUSTOMER (1 of 1)

Set the configuration parameters, then press OK.
This property group encompasses properties that are set for a business object by the EMD.

StatusColumnName: ID

StatusValue: ID

BeforeCreateSP: LNAME, FNAME

StoredProcedure: [Dropdown]

StoredProcedureParameters: [Text Field]

None: [Text Field]

AfterCreateSP: [Dropdown]

StoredProcedure: [Dropdown]

StoredProcedureParameters: [Text Field]

None: [Text Field]

CreateSP: [Dropdown]

StoredProcedure: [Dropdown]

StoredProcedureParameters: [Text Field]

None: [Text Field]

BeforeUpdateSP: [Dropdown]

StoredProcedure: [Dropdown]

StoredProcedureParameters: [Text Field]

You can set configuration properties such as StatusColumnName and StatusValue, which come into play when you want to use logical deletes from the database. This is also where you would associate the stored procedure type.

Enterprise Service Discovery - Inbound

- Activation Specification Properties
- Delivery Mode and Polling Info, Credentials, and Miscellaneous

Properties for Activation Spec

BO Namespace: *

Event Distribution Database Info

EDT DatabaseName:

EDT DriverName:

EDT SchemaName:

EDT TableName:

EDT UserName:

EDT Password:

Auto Create EDT

Delivery Mode and Polling Info

Delivery Type:

Poll Period:

Poll Quantity:

Retry Interval:

UserCredentials

UserName: *

Password: *

MachineCredentials

DatabaseURL: *

JdbcDriverClass: *

Miscellaneous

EventOrderBy: *

28

IBM WebSphere Adapter for JDBC © 2005 IBM Corporation

Once the query has been run and the selection of business objects has been done, you then specify inbound or outbound service type. Here you see an example of the set of properties specified with an inbound service type.

Enterprise Service Discovery - Inbound

- Adapter Properties
- Miscellaneous Properties

ResourceAdapterProperties

Logging and Tracing

Adapter ID: * ResourceAdapter

Log File Size: 0

Log File Name:

Number Of Log Files: 1

Trace File Size: 0

Trace File Name:

Number Of Trace Files: 1

Miscellaneous

AutoCommit

PingQuery:

EventTableName:

DatabaseVendor: *

QueryTimeOut:

ReturnDummyBOForSP: false

29

IBM WebSphere Adapter for JDBC © 2005 IBM Corporation

You will continue through the wizard, setting additional properties that can be specified for inbound service type.

Enterprise Service Discovery - Outbound

- J2C Authentication Data Entry and other Managed Connection Factory Properties
- Adapter Properties

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

Use connection properties specified on server

Use discovered connection properties

J2C Authentication Data Entry:

ManagedConnectionProperties

UserCredentials

UserName:

Password:

MachineCredentials

DatabaseURL: * *

JdbcDriverClass: * *

ResourceAdapterProperties

Logging and Tracing

Adapter ID: * *

Log File Size:

Log File Name:

Number Of Log Files:

Trace File Size:

Trace File Name:

Number Of Trace Files:

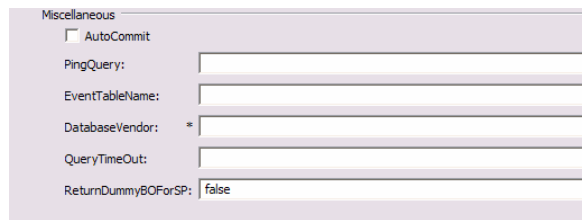
30

IBM WebSphere Adapter for JDBC © 2005 IBM Corporation

Here you see an example of the set of properties specified with an outbound service type.

Enterprise Service Discovery - Outbound

- Miscellaneous Properties



Miscellaneous

AutoCommit

PingQuery:

EventTableName:

DatabaseVendor: *

QueryTimeout:

ReturnDummyBOForSP: false

You will continue through the wizard, setting additional properties that can be specified for outbound service type.

Agenda

- Overview
- Architecture
- Enterprise Service Discovery
- **Object Model**
- Event Manager
- Summary

This section will provide an overview of the object model.

Business Object Structure

- Each business object (BO) corresponds to a database table or view
 - ▶ Each simple attribute (represents a single value) within the object corresponds to a column in that table or view
- Each BO can represent a view that spans multiple database tables
 - ▶ The adapter can use this type of BO when:
 - processing outbound requests with operations Retrieve and RetrieveAll
 - processing inbound events with operations Create, Update, and Delete



Each business object corresponds to a database table or view. Each simple attribute within the object corresponds to a column in that table or view. Each business object can represent a view that spans multiple database tables. The WebSphere Adapter for JDBC can use this type of business object when processing outbound requests with the operations of retrieve and retrieveAll or processing inbound events with the operations of create, update, and delete.

Business Object Structure

- Each BO can be flat or hierarchical
 - ▶ Flat
 - All attributes are simple and represent one row in the table
 - ▶ Hierarchical
 - Attributes represent a child BO, an array of child BOs, or a combination
 - Each child BO can contain a child BO, an array of child BOs or a combination

Each business object can be of a flat or hierarchical structure. A flat structure is where all attributes are simple and represent one row in the table. A hierarchical structure is where attributes represent a child business object, an array of child business objects or a combination.

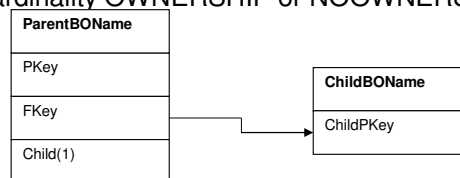
Business Object Structure

- Data descriptions generated from the database components using Enterprise Service Discovery are represented as XML schemas
- Business Object
 - ▶ maps to a complex type definition
 - ▶ Attributes map to element type definition
 - ▶ Application Specific Information is contained in annotations at the complex type
 - ▶ Application Specific Information for each property is contained in annotations for the element types

Data descriptions generated from the database components using enterprise service discovery are represented as XML schemas. Looking at the structure of a business object, the business object maps to a complex type definition, attributes map to element type definitions, application specific information is contained in annotations at the complex type, and application specific information for each property is contained in annotations for the element types.

Hierarchical Business Object Single Cardinality Relationship

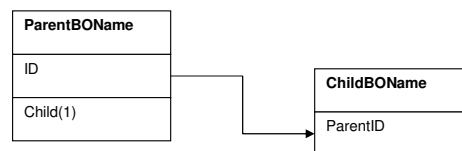
- Parent BO has an attribute that represents one child BO
 - ▶ At least two attributes represent the relationship stored in the parent BO
 - The type of one attribute in the parent BO is of the same type as the child BO
 - A simple attribute that contains the child BO's primary key as a foreign key in the parent BO
 - ▶ Parent BO has as many foreign key attributes as the child BO has primary key attributes
 - ▶ Each parent BO can have only one child BO of a given type
 - ▶ Parent BO can have cardinality OWNERSHIP or NOOWNERSHIP child BO (lookup tables)



The next several slides discuss hierarchical business object cardinality relationships. In the case of a hierarchical business object with a single cardinality relationship, the parent business object has an attribute that represents one child business object. At least two attributes represent the relationship stored in the parent business object. In the parent business object, the type of one attribute is of the same type as the child business object. Also, a simple attribute in the parent business object contains the child business object's primary key as a foreign key. Since the foreign key for the child business object is stored in the parent business object, each parent business object can have only one child business object of a given type. Cardinality can be of the ownership or noownership kind. With ownership, typically each parent business object owns the data within the child business object that it contains. For example, customer business object contains one address business object. When a new customer is created, a new row is inserted into both the customer and address tables; likewise for other operations. The address is unique to the customer. In the case of nowownership, there are cases where multiple hierarchical business objects contain the same data, which none of them owns. For example, address business object contains a state business object that represents the state lookup table. When a new address is created, the state is simply retrieved from the lookup table and not created; likewise for other operations. The state business object is independent of the address business object.

Hierarchical Business Object Single Cardinality Relationship

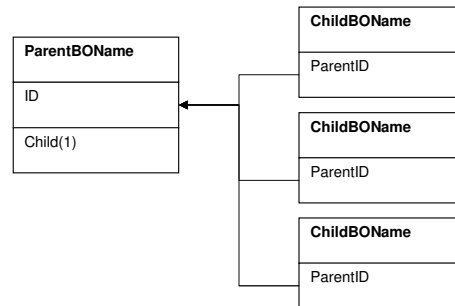
- Relationship is stored in the child BO
 - ▶ Child BO contains a foreign key whose value is identical to the value in the parent BO's primary key
 - ▶ Child BO does not exist independently of its parent BO and can be accessed only through the parent BO
 - ▶ Requires the parent BO and its primary key value to exist before the child BO and its foreign key value can be created



Another case of a hierarchical business object with a single cardinality relationship, is where the relationship is stored in the child business object. In this case, the child business object contains a foreign key whose value is identical to the value of the parent business object's primary key. The child business object does not exist independently of its parent business object and can be accessed only through the parent business object. This type of relationship requires that the parent business object and its primary key value exist before the child business object and its foreign key value can be created.

Hierarchical Business Object Multiple-Cardinality Relationship

- An attribute in the parent BO represents an array of child BOs
 - ▶ Has only one attribute that represents the relationship (primary key)
 - ▶ Foreign key is stored in the child BO
 - ▶ Child BO has as many foreign keys as the parent has primary keys
 - ▶ Each parent BO can have 0 or more child BOs



In the case of a hierarchical business object with a multiple cardinality relationship, an attribute in the parent business object represents an array of child business objects. The parent business object has only one attribute that represents the relationship, the primary key. The foreign key is stored in the child business object. The child business object has as many foreign keys as the parent has primary keys. Each parent business object can have zero or more child business objects.

Application Specific Information

- **Business Object ASI**
 - ▶ **TableName**
 - Database table associated with the business object
 - ▶ **StatusColumnName**
 - Name of the database column used to perform logical delete operations. If defined, the adapter performs a logical delete operation. If not defined, the adapter performs a physical delete operation.
 - ▶ **StatusValue**
 - Signifies whether a business object is inactive or deleted

Application-specific information in business object definitions provides the adapter with application-dependent instructions on how to process business objects. The adapter parses the application-specific information from the attributes or verb of a business object or from the business object itself to generate queries for create, update, retrieve, and delete operations. The adapter stores some of the business-object's application-specific information in cache and uses this information to build queries for all the verbs.

Application Specific Information

▪ Verb ASI

- ▶ Associate Stored Procedures at the verb level
- ▶ Each Stored Procedure definition consists of:
 - StoredProcedureType
 - Determines when the stored procedure is called
 - For example: BeforeCreateSP, AfterCreateSP, CreateSP, ... etc.
 - StoredProcedureName
 - Name of the Stored Procedure associated with the appropriate StoredProcedureType
 - ResultSet
 - Determines whether a resultset is returned
 - Parameters
 - Can be a combination of input only (IP), output only (OP), and input and output (IO)



In the case of verb application specific information, associated stored procedures are specified. Each stored procedure definition consists of the storedproceduretype, storedprocedurename, resultset, and parameters. Oracle stored procedures can return ResultSets as OUTPUT parameters only, whereas other databases return them as return values. This feature for Oracle is handled by defining an output parameter as a ResultSet (OP=RS).

Application Specific Information

- Attribute ASI for simple attributes
 - ▶ ColumnName
 - ▶ PrimaryKey
 - ▶ ForeignKey
 - ▶ UniqueIdentifier (UID)
 - ▶ OrderBy
 - ▶ FixedChar
 - ▶ ByteArray
 - ▶ KeepRelationship
 - ▶ Ownership



The application-specific information for attributes differs depending on whether the attribute is a simple attribute or an attribute that represents a child or an array of child business objects. The application-specific information for an attribute that represents a child also differs depending on whether the parent-child relationship is stored in the child or in the parent.

Business Object Content

- Relate to the amount and purpose of information conveyed by the business object
- After Image
 - ▶ Presence of a top-level verb indicates After Image
 - ▶ State of business object after all changes have been made
- Delta
 - ▶ Absence of a top-level verb indicates Delta
 - ▶ Operation must be ApplyChanges and no verb specified
 - Adapter will inspect ChangeSummary information to identify the operation for each business object
 - ApplyChanges is an outbound operation

The business object content relates to the amount and purpose of 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 outbound operation, applyChanges, supports the delta content.

Verbs

- Verbs
 - ▶ Reflect the state of the business object
 - ▶ Are definable in the Business Graph for After-Image business objects only
 - ▶ Are used to communicate the type of After-Image event
 - ▶ Verbs supported for After-Image
 - Create
 - Update
 - Delete

Verbs reflect the state of the business object and are definable in the Business Graph for after-image business objects only. The verb is used to communicate the type of after-image event. The three verbs supported for an after-image event are create, update, and delete.

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



Operations specified in the business object indicate what type of processing is requested. Business objects are passed to the adapter and processed according to the operation specified. In the case of inbound operations, after-image only support is available for the create, update, and delete operations.

Operations

- Outbound operations
 - ▶ After Image
 - Create
 - Update
 - Delete
 - Supports physical and logical deletes
 - ▶ Delta support
 - ApplyChanges (no top level verb exists)
 - Adapter will inspect ChangeSummary information to identify the operation for each business object

In the case of outbound operations, after image support is available for create, update, and delete; delta support is available for applyChanges.

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
 - Set MaxRecords value found on the Configure Objects panel for outbound ServiceType during the Enterprise Service Discovery process



Other supported outbound operations include retrieve and retrieveAll. In the case of retrieve, when given a hierarchical business object, the retrieve operation first makes a copy of the top level business object without any children, and then recursively retrieves the child business objects starting with the parent of the incoming business object. In the case of retrieveAll, this operation enables the adapter to retrieve an array of business objects from the database. The ResultSetLimit property determines the number of records to return. This value can be set during the enterprise service discovery process, by specifying the value for MaxRecords on the Configure Objects panel for outbound ServiceType.

Agenda

- Overview
- Architecture
- Enterprise Service Discovery
- Object Model
- **Event Manager**
- Summary



This section will provide information regarding the event manager and its role in inbound processing.

Event Manager

- Framework for delivering Inbound Events
 - ▶ Provides recovery and guaranteed event delivery
- Event Table (Event Store) is required
 - ▶ Database application populates the event table with any changes in the user tables
- Staging Table (Event Distribution Table)
 - ▶ Reference for each retrieved event is written to the Staging Table for each active endpoint
 - ▶ Each event delivered to it's corresponding endpoint is part of a unique XA transaction
- Creation of a persistent staging table is optional
 - ▶ In-memory mechanism will be used if persistent table is not specified, but lose guaranty delivery
 - Useful for testing and Proof of Concept scenarios

The Event Manager is a framework for delivering inbound events. An event table, or event store, must exist before inbound processing can occur. Events are generated to the event store as a result of updates to databases. The adapter polls the event store and retrieves a batch of events. It then processes and transmits these events to various predefined endpoints, with a subscription to the specific business object, in the WebSphere Process Server. *Endpoints* are client consumers of the event. The use of a persistent staging table, or event distribution table, is optional. However, it is through the usage of a persistent staging table that the event manager framework can provide guaranteed event delivery and recovery in the case of a server failure. This is done by using XA transactions between the recording of the event to the staging table and the delivery of the event to the endpoint.

Agenda

- Overview
- Architecture
- Enterprise Service Discovery
- Object Model
- Event Manager
- **Summary**

This section will provide a summary of this presentation.

Summary

- WebSphere Adapter for JDBC enables integration with SCA Applications and JDBC 2.0 Databases
 - ▶ Inbound and Outbound support
- Provides Support for Enterprise Metadata Discovery for discovering services
- Advanced Feature Support
 - ▶ Stored Procedures, UID, OrderBy, Query Timeout, Fixed Length Strings
- Event Manager Framework provides event persistence and guaranteed event delivery

To summarize this presentation, the WebSphere Adapter for JDBC enables integration with SCA business integration applications and Enterprise Information System database applications. The adapter supports both inbound and outbound interaction. Enterprise service discovery is used for discovery of services and creating the service description as well as specifying values for custom adapter properties and discovery of business objects. Advanced feature support includes support for stored procedures, UID, orderby, query timeout, and fixed length strings. The event manager framework is used to provide event persistence and guaranteed event delivery.

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/v6rxmx/index.jsp>

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 2004. 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.