Date	Version	Author	Remarks
Oct-13-2006	1	Jagadish Ramu/Kshitiz Saxena	Created
Dec-10-2006	1.01	Jagadish Ramu	Changed default values for 2 attributes of jdbc-connection-pool : match-connections - "false", statement-timeout-in-seconds - "-1"

<!-- jdbc-resource
JDBC javax.sql.(XA)DataSource resource definition

Used in:
resources
-->
<!ELEMENT jdbc-resource (description?, property*)>

<!ATTLIST jdbc-resource
jndi-name CDATA #REQUIRED
pool-name CDATA #REQUIRED
object-type %object-type; "user"
enabled %boolean; "true">

<!-- jdbc-connection-pool
jdbc-connection-pool defines configuration used to create and
manage a pool physical database connections. Pool definition is
named, and can be referred to by multiple jdbc-resource elements
(See <jdbc-resource>).

Each named pool definition results in a pool instantiated at server
start-up. Pool is populated when accessed for the first time. If two or
more jdbc-resource elements point to the same jdbc-connection-pool
element, they are using the same pool of connections, at run time.

children
property
Most JDBC 2.0 drivers permit use of standard property lists,
to specify User, Password and other resource configuration.
While these are optional properties, according to the
specification, several of these properties may be necessary
for most databases. See Section 5.3 of JDBC 2.0 Standard
Extension API.

The following are the names and corresponding values for these
properties

databaseName

Name of the Database

serverName

Database Server name.

port

Port where a Database server is listening for requests.

networkProtocol

Communication Protocol used.

user

default name of the database user with which connections will be stablished. Programmatic database authentication or default-resource-principal specified in vendor specific web and ejb deployment descriptors will take precedence, over this default. The details and caveats are described in detail in the Administrator's guide.

password

password for default database user

roleName

The initial SQL role name.

datasourceName

used to name an underlying XADataSource, or ConnectionPoolDataSource when pooling of connections is done

description

Textual Description

When one or more of these properties are specified, they are passed as is using set<Name>(<Value>) methods to the vendors Datasource class (specified in datasource-classname). User and Password properties are used as default principal, if Container Managed authentication is specified and a default-resource-principal is not found in application deployment descriptors.

attributes

allow-non-component-callers

A pool with this property set to true, can be used by non-J2EE components (i.e components other than EJBs or Servlets). The returned connection is enlisted automatically with the transaction context obtained from the transaction manager. This property is to enable the pool to be used by non-component callers such as ServletFilters, Lifecycle modules, and 3rd party persistence managers. Standard J2EE components can continue to use such pools. Connections obtained by non-component callers are not automatically

cleaned at the end of a transaction by the container. They need to be explicitly closed by the the caller.

connection-validation-method

specifies the type of validation to be performed when is-connection-validation-required is true. The following types of validation are supported:

auto-commit

using connection.autoCommit()

meta-data

using connection.getMetaData()

table

performing a query on a user specified table (see validation-table-name).

datasource-classname

Name of the vendor supplied JDBC datasource resource manager. An XA or global transactions capable datasource class will implement javax.sql.XADatasource interface. Non XA or Local transactions only datasources will implement javax.sql.Datasource interface.

fail-all-connections

indicates if all connections in the pool must be closed should a single validation check fail. The default is false. One attempt will be made to re-establish failed connections.

idle-timeout-in-seconds

maximum time in seconds, that a connection can remain idle in the pool. After this time, the pool implementation can close this connection. Note that this does not control connection timeouts enforced at the database server side. Adminsitrators are advised to keep this timeout shorter than the database server side timeout (if such timeouts are configured on the specific vendor's database), to prevent accumulation of unusable connection in Application Server.

is-connection-validation-required

if true, connections are validated (checked to find out if they are usable) before giving out to the application. The default is false.

is-isolation-level-guaranteed

Applicable only when a particular isolation level is specified for transaction-isolation-level. The default value is true. This assures that every time a connection is obtained from the pool, it is guaranteed to have the isolation set to the desired value. This could have some performance impact on some JDBC drivers. Can be set to false by that administrator when they are certain that the application does not change the isolation level before

returning the connection.

max-pool-size

maximum number of conections that can be created

max-wait-time-in-millis

amount of time the caller will wait before getting a connection timeout. The default is 60 seconds. A value of 0 will force caller to wait indefinitely.

name

unique name of the pool definition.

non-transactional-connections

A pool with this property set to true returns non-transactional connections. This connection does not get automatically enlisted with the transaction manager.

pool-resize-quantity

number of connections to be removed when idle-timeout-in-seconds timer expires. Connections that have idled for longer than the timeout are candidates for removal. When the pool size reaches steady-pool-size, the connection removal stops.

res-type

DataSource implementation class could implement one of of javax.sql.DataSource, javax.sql.XADataSource or javax.sql.ConnectionPoolDataSource interfaces. This optional attribute must be specified to disambiguate when a Datasource class implements two or more of these interfaces. An error is produced when this attribute has a legal value and the indicated interface is not implemented by the datasource class. This attribute has no default value.

steady-pool-size

minimum and initial number of connections maintained in the pool.

transaction-isolation-level

Specifies the Transaction Isolation Level on the pooled database connections. Optional. Has no default. If left unspecified the pool operates with default isolation level provided by the JDBC Driver. A desired isolation level can be set using one of the standard transaction isolation levels, which see.

Applications that change the Isolation level on a pooled connection programmatically, risk polluting the pool and this could lead to program errors. Also see: is-isolation-level-guaranteed

validation-table-name

specifies the table name to be used to perform a query to validate a connection. This parameter is mandatory, if

connection-validation-type set to table. Verification by
accessing a user specified table may become necessary for
connection validation, particularly if database driver caches
calls to setAutoCommit() and getMetaData().

connection-leak-timeout-in-seconds

To aid user in detecting potential connection leaks by the application.
When a connection is not returned back to the pool by the application
within the specified period, it is assumed to be a potential leak and
stack trace of the caller will be logged. Default is 0 seconds, which
implies there is no leak detection, by default. A non-zero value turns
on leak tracing.

connection-leak-reclaim

If enabled, connection will be re-usable (put back to pool) after
connection-leak-timeout-in-seconds occurs. Default value is false.

connection-creation-retry-attempts

The number of attempts to create a new connection. Default is 0, which
implies no retries.

connection-creation-retry-interval-in-seconds

The time interval between retries while attempting to create a connection.
Default is 10 seconds. Effective when connection-creation-retry-attempts
is greater than 0.

validate-atmost-once-period-in-seconds

Used to set the time-interval within which a connection is validated
atmost once. Default is 0 seconds, not enabled.

statement-timeout-in-seconds

Sets the timeout property of a connection to enable termination of abnormally
long running queries. Default value is -1, not enabled.

lazy-connection-enlistment

Enlist a resource to the transaction only when it is actually used in
a method, which avoids enlistment of connections, that are not used,
in a transaction. This also prevents unnecessary enlistment of connections
cached in the calling components. Default value is false.

lazy-connection-association

Connections are lazily associated when an operation is performed on
them. Also they are disassociated when the transaction is completed
and a component method ends, which helps to reuse the physical
connections. Default value is false.

associate-with-thread

Associate a connection with the thread such that when the
same thread is in need of a connection, it can reuse the connection
already associated with that thread, thereby not incurring the overhead
of getting a connection from the pool. Default value is false.

match-connections

To switch on/off connection matching for the pool. It can be set to false if the
administrator knows that the connections in the pool

```
        will always be homogeneous and hence a connection picked from the pool
        need not be matched by the resource adapter. Default value is false.
    max-connection-usage-count
      When specified, connections will be re-used by the pool for the specified number
      of times after which it will be closed. eg : To avoid statement-leaks.
      Default value is 0, which implies the feature is not enabled.
    wrap-jdbc-objects
      When set to true, application will get wrapped jdbc objects for Statement,
      PreparedStatement, CallableStatement, ResultSet, DatabaseMetaData.

  Used in:
    resources
-->

<!ELEMENT jdbc-connection-pool (description?, property*)>
<!ATTLIST jdbc-connection-pool
    name CDATA #REQUIRED
    datasource-classname CDATA #REQUIRED
    res-type (javax.sql.DataSource | javax.sql.XADataSource | javax.sql.ConnectionPoolDataSource) #IMPLIED
    steady-pool-size CDATA "8"
    max-pool-size CDATA "32"
    max-wait-time-in-millis CDATA "60000"
    pool-resize-quantity CDATA "2"
    idle-timeout-in-seconds CDATA "300"
    transaction-isolation-level %isolation; #IMPLIED
    is-isolation-level-guaranteed %boolean; "true"
    is-connection-validation-required %boolean; "false"
    validate-atmost-once-period-in-seconds CDATA “0”
    connection-validation-method (auto-commit | meta-data |  table) "auto-commit"
    validation-table-name CDATA #IMPLIED
    fail-all-connections %boolean; "false"
    non-transactional-connections %boolean; "false"
    connection-leak-timeout-in-seconds CDATA “0”
    connection-leak-reclaim %boolean; “false”
    connection-creation-retry-attempts  CDATA “0”
    connection-creation-retry-interval-in-seconds CDATA “10”
    statement-timeout-in-seconds CDATA “-1”
    max-connection-usage-count CDATA “0”
    lazy-connection-enlistment %boolean; "false"
    lazy-connection-association %boolean; "false"
    associate-with-thread %boolean; "false"
    match-connections %boolean; "false"
    wrap-jdbc-objects %boolean; “false”
    allow-non-component-callers %boolean; "false">
```

```
<!-- connector-resource
  Used in:
    resources
-->
<!ELEMENT connector-resource (description?, property*)>

<!ATTLIST connector-resource
    jndi-name CDATA #REQUIRED
    pool-name CDATA #REQUIRED
    object-type %object-type; "user"
    enabled %boolean; "true">



<!-- connector-connection-pool
    connector-connection-pool defines configuration used to create
    and manage a pool of connections to a EIS. Pool definition is
    named, and can be referred to by multiple connector-resource
    elements (See connector-resource).

    Each named pool definition results in a pool instantiated at server
    start-up. Pool is populated when accessed for the first time. If two or
    more connector-resource elements point to the same
    connector-connection-pool element, they are using the same pool of
    connections, at run time.

    There can be more than one pool for one connection-definition in one
    resource-adapter.


  children
    property
        Properties are used to override the ManagedConnectionFactory
        javabean configuration settings.

        When one or more of these properties are specified, they are passed as
        is using set<Name>(<Value>) methods to the Resource Adapter's
        ManagedConnectionfactory class (specified in ra.xml).


  attributes
    connection-definition-name
        unique name, identifying one connection-definition in a
        Resource Adapter. Currently this is ConnectionFactory type.
    fail-all-connections
        indicates if all connections in the pool must be closed
```

should a single connection fail validation. The default is false. One attempt will be made to re-establish failed connections.

idle-timeout-in-seconds

maximum time in seconds, that a connection can remain idle in the pool. After this time, the pool implementation can close this connection. Note that this does not control connection timeouts enforced at the database server side. Adminsitrators are advised to keep this timeout shorter than the EIS connection timeout (if such timeouts are configured on the specific EIS), to prevent accumulation of unusable connection in Application Server.

is-connection-validation-required

This attribute specifies if the connection that is about to be returned is to be validated by the container,

max-pool-size

maximum number of conections that can be created

max-wait-time-in-millis

amount of time the caller will wait before getting a connection timeout. The default is 60 seconds. A value of 0 will force caller to wait indefinitely.

name

unique name of the pool definition.

pool-resize-quantity

number of connections to be removed when idle-timeout-in-seconds timer expires. Connections that have idled for longer than the timeout are candidates for removal. When the pool size reaches steady-pool-size, the connection removal stops.

resource-adapter-name

This is the name of resource adapter. Name of .rar file is taken as the unique name for the resource adapter.

steady-pool-size

minimum and initial number of connections maintained in the pool.

transaction-support

Indicates the level of transaction support that this pool will have. Possible values are "XATransaction", "LocalTransaction" and "NoTransaction". This attribute will override that transaction support attribute in the Resource Adapter in a downward compatible way, i.e it can support a lower/equal transaction level than specified in the RA, but not a higher level.

connection-leak-timeout-in-seconds

To aid user in detecting potential connection leaks by the application.

When a connection is not returned back to the pool by the application within the specified period, it is assumed to be a potential leak and stack trace of the caller will be logged. Default is 0 seconds, which implies there is no leak detection, by default. A non-zero value turns on leak tracing.

connection-leak-reclaim

If enabled, connection will be re-usable (put back to pool) after connection-leak-timeout-in-seconds occurs. Default value is false.

connection-creation-retry-attempts

The number of attempts to create a new connection. Default is 0, which implies no retries.

connection-creation-retry-interval-in-seconds

The time interval between retries while attempting to create a connection Default is 10 seconds. Effective when connection-creation-retry-attempts is greater than 0.

validate-atmost-once-period-in-seconds

Used to set the time-interval within which a connection is validated atmost once. Default is 0 seconds, not enabled.

lazy-connection-enlistment

Enlist a resource to the transaction only when it is actually used in a method, which avoids enlistment of connections, that are not used, in a transaction. This also prevents unnecessary enlistment of connections cached in the calling components. Default value is false.

lazy-connection-association

Connections are lazily associated when an operation is performed on them. Also they are disassociated when the transaction is completed and a component method ends, which helps to reuse the physical connections. Default value is false.

associate-with-thread

Associate a connection with the thread such that when the same thread is in need of a connection, it can reuse the connection already associated with that thread, thereby not incurring the overhead of getting a connection from the pool. Default value is false.

match-connections

To switch on/off connection matching for the pool. It can be set to false if the administrator knows that the connections in the pool will always be homogeneous and hence a connection picked from the pool need not be matched by the resource adapter. Default value is true.

max-connection-usage-count

When specified, connections will be re-used by the pool for the specified number of times after which it will be closed. eg : To avoid statement-leaks. Default value is 0, which implies the feature is not enabled.

Used in:

```
    resources
-->
<!ELEMENT connector-connection-pool (description?, security-map*, property*)>
<!ATTLIST connector-connection-pool
    name CDATA #REQUIRED
    resource-adapter-name CDATA #REQUIRED
    connection-definition-name CDATA #REQUIRED
    steady-pool-size CDATA "8"
    max-pool-size CDATA "32"
    max-wait-time-in-millis CDATA "60000"
    pool-resize-quantity CDATA "2"
    idle-timeout-in-seconds CDATA "300"
    fail-all-connections %boolean; "false"
    transaction-support (XATransaction | LocalTransaction | NoTransaction) #IMPLIED
    is-connection-validation-required %boolean; "false"
    validate-atmost-once-period-in-seconds CDATA “0”
    connection-leak-timeout-in-seconds CDATA “0”
    connection-leak-reclaim %boolean; “false”
    connection-creation-retry-attempts  CDATA “0”
    connection-creation-retry-interval-in-seconds CDATA “10”
    lazy-connection-enlistment %boolean; "false"
    lazy-connection-association %boolean; "false"
    associate-with-thread %boolean; "false"
    match-connections %boolean; "true"
    max-connection-usage-count CDATA “0”>
```