JCA Adaptor Configuration

An excerpt from the JBoss Administration and Development Book, Third Edition (3.2.x Series)

Scott Stark and

The JBoss Group

Configuring JCA Adaptors

Configuration of the JCA resource adaptors may be done by configuring the JBoss JCA services along with the JCA resource adaptor as shown in the previous section. JBoss 3.2+ provides an alternate simplified schema that avoids having to specify so much redundant configuration information.

Configuring JDBC DataSources

The syntax for configuring JCA JDBC connection factories has been simplified in 3.2. Rather than configuring the connection manager factory related MBeans discussed in the previous section via a mbean services deployment descriptor, an abbreviated datasource centric descriptor is used. This is transformed into the standard jboss-service.xml mbean services deployment descriptor using a XSL transform applied by the `org.jboss.deployment.XSLSubDeployer` included in the jboss-jca.sar deployment. The simplified configuration descriptor is deployed the same as other deployable components. The descriptor must be named using a "*-ds.xml" pattern in order to be recognized by the `XSLSubDeployer`.

The schema for the top-level datasource elements of the "*-ds.xml" configuration deployment file is shown in Figure 7-6.

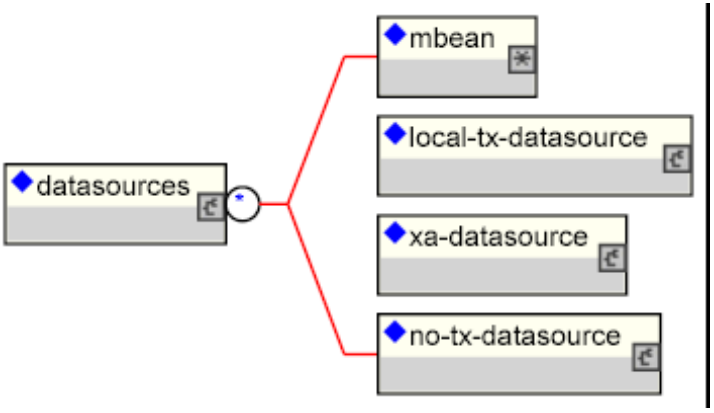

FIGURE 7-6. The simplified JCA DataSource configuration descriptor top-level schema elements

Multiple datasource configurations may be specified in a configuration deployment file. The child elements of the datasources root are:

- **mbean**: Any number mbean elements may be specified to define MBean services that should be included in the jboss-service.xml descriptor that results from the transformation. This may be used to configure services used by the datasources.
- **no-tx-datasource**: this element is used to specify the `org.jboss.resource.connectionmanager.NoTxConnectionManager` service configuration. `NoTxConnectionManager` is a JCA connection manager with no transaction support. The no-tx-datasource child element schema is given in Figure 7-7.
- **local-tx-datasource**: this element is used to specify the `org.jboss.resource.connectionmanager.LocalTxConnectionManager` service configuration. `LocalTxConnectionManager` implements a `ConnectionEventListener` that implements `XAResource` to manage transactions through the transaction manager. To ensure that all work in a local transaction occurs over the same `ManagedConnection`, it includes a xid to `ManagedConnection` map. When a `Connection` is requested or a transaction started with a connection handle in use, it checks to see if a `ManagedConnection` already exists enrolled in the global transaction and uses it if found. Otherwise, a free `ManagedConnection` has its `LocalTransaction` started and is used. The local-tx-datasource child element schema is given in Figure 7-8.
- **xa-datasource**: this element is used to specify the `org.jboss.resource.connectionmanager.XATxConnectionManager` service configuration. `XATxConnectionManager` implements a `ConnectionEventListener` that obtains the `XAResource` to manage transactions through the transaction manager from the adaptor `ManagedConnection`. To ensure that all work in a local transaction occurs over the same `ManagedConnection`, it includes a xid to `ManagedConnection` map. When a `Connection` is requested or a transaction started with a connection handle in use, it checks to see if a `ManagedConnection` already exists enrolled in the global transaction and uses it if found. Otherwise, a free `ManagedConnection` has its `LocalTransaction` started and is used. The xa-datasource child element schema is given in Figure 7-9.

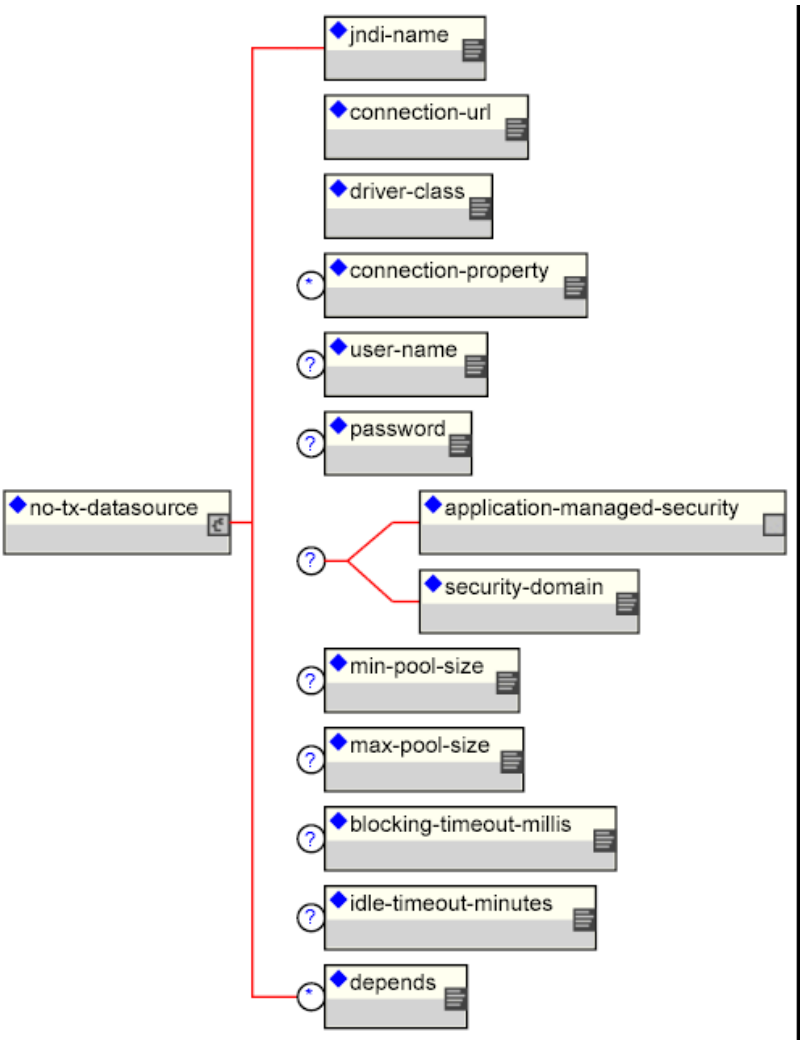

FIGURE 7-7. **The non-transactional DataSource configuration schema**

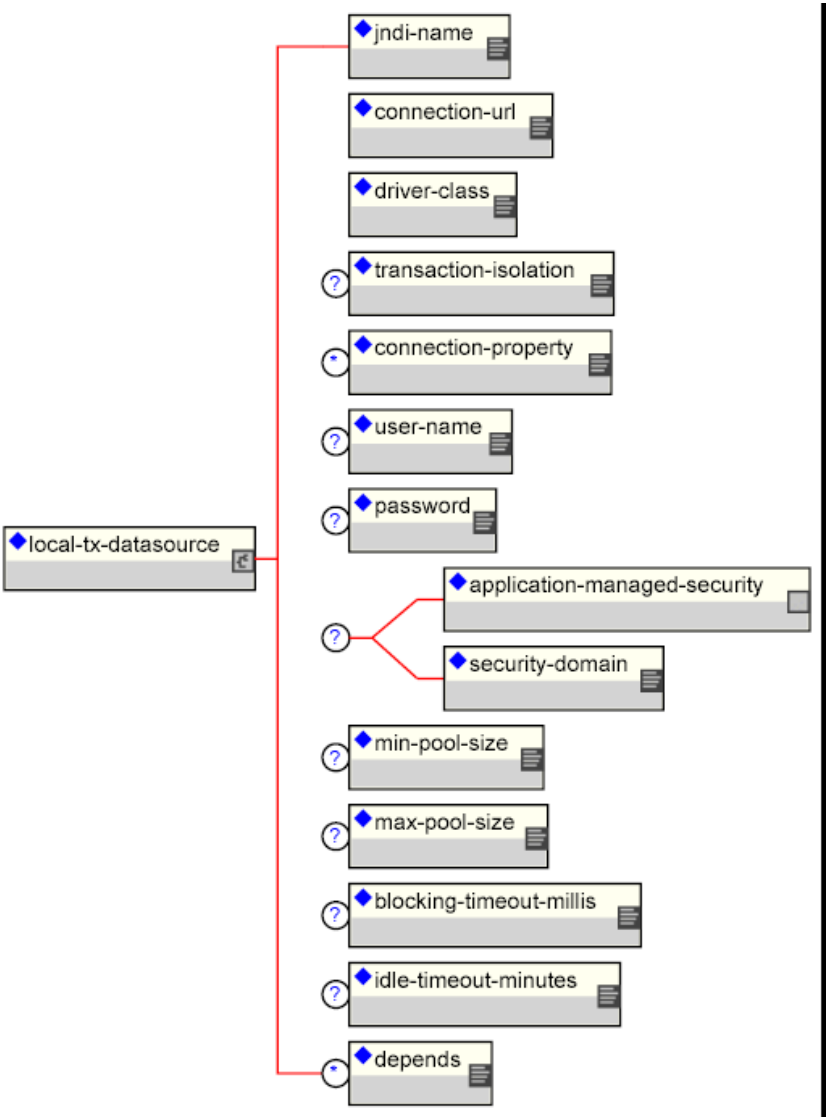

FIGURE 7-8. The non-XA DataSource configuration schema

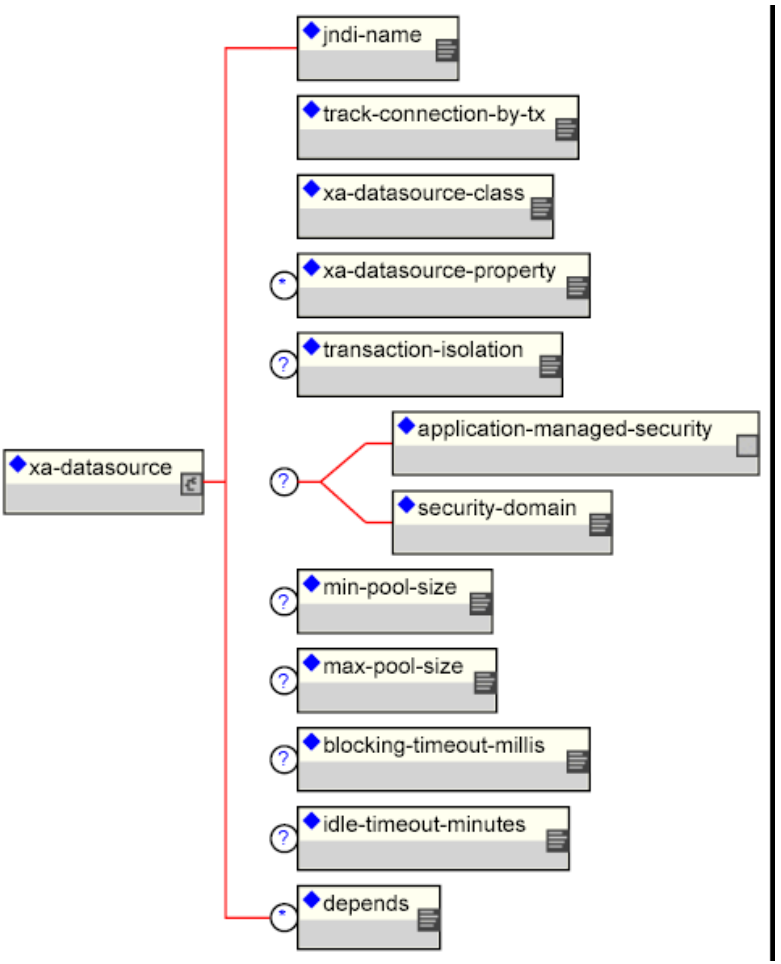

FIGURE 7-9. The XADataSource configuration schema

Elements that are common to all datasources include:

- **jndi-name**: The JNDI name under which the `DataSource` wrapper will be bound. Note that this name is relative to the "java:/" prefix. The full JNDI name of the `DataSource` will be "java:/" + jndi-name. `DataSource` wrappers are bound under the "java:/" prefix since they are not usable outside of the server VM.

application-managed-security: Specifying this element indicates that application code supplied parameters, such as from getConnection(user, pw), are used to distinguish connections in the pool.

- **security-domain**: Specifying this element indicates that either application code supplied parameters, or JAAS `Subject` based information is to distinguish connections in the pool. The content of the security-domain is the name of the JAAS security manager that will handle authentication. This name correlates to the JAAS login-config.xml descriptor application-policy/name attribute.
- **min-pool-size**: This element specifies the minimum number of connections a pool should hold. These pool instances are not created until an initial request for a connection is made. This default to 0.
- **max-pool-size**: This element specifies the maximum number of connections for a pool. No more than the max-pool-size number of connections will be created in a pool. This defaults to 20.
- **blocking-timeout-millis**: This element specifies the maximum time in milliseconds to block while waiting for a connection before throwing an exception. Note that this blocks only while waiting for a permit for a connection, and will never throw an exception if creating a new connection takes an inordinately long time. The default is 5000.
- **idle-timeout-minutes**: This element specifies the maximum time in minutes a connection may be idle before being closed. The actual maximum time depends also on the IdleRemover scan time, which is 1/2 the smallest idle-timeout-minutes of any pool.
- **depends**: The depends element specifies the JMX `ObjectName` string of a service that the connection manager services depend on. The connection manager service will not be started until the dependent services have been started.

Additional common child elements for both no-tx-datasource and local-tx-datasource include:

- **connection-url**: The JDBC driver connection URL string, for example, “jdbc:hsqldb:hsql://localhost:1701”.
- **driver-class**: The fully qualified name of the JDBC driver class, for example, “org.hsqldb.jdbcDriver”.
- **connection-property**: The connection-property element allows you to pass in arbitrary connection properties to the `java.sql.Driver.connect(url, props)` method. Each connection-property specifies a string name/value pair with the property name coming from the name attribute and the value coming from the element content.
- **user-name**: This element specifies the default username used when creating a new connection. The actual username may be overridden by the application code `getConnection` parameters or the connection creation context JAAS `Subject`.
- **password**: This element specifies the default password used when creating a new connection. The actual password may be overridden by the application code `getConnection` parameters or the connection creation context JAAS `Subject`.

The remaining local-tx-datasource is:

- **transaction-isolation**: This element specifies the `java.sql.Connection` transaction isolation level to use. The constants defined in the `Connection` interface are the possible element content values and include:
    - TRANSACTION_READ_UNCOMMITTED
    - TRANSACTION_READ_COMMITTED
    - TRANSACTION_REPEATABLE_READ
    - TRANSACTION_SERIALIZABLE
    - TRANSACTION_NONE

The unique xa-datasource child elements are:

- **track-connection-by-tx**: Specifying a true value for this element makes the connection manager keep an xid to connection map and only put the connection back in the pool when the transaction completes and all the connection handles are closed or disassociated (by the method calls returning). As a side effect, we never suspend and resume the xid on the connection's `XAResource`. This is the same connection tracking behavior used for local transactions.

  The XA spec implies that any connection may be enrolled in any transaction using any xid for that transaction at any time from any thread (suspending other transactions if necessary). The original JCA implementation assumed this and aggressively delisted connections and put them back in the pool as soon as control left the ejb they were used in or handles were closed. Since some other transaction could be using the connection the next time work needed to be done on the original transaction, there is no way to get the original connection back. It turns out that most `XADataSource` driver vendors do not support this, and require that all work done under a particular xid go through the same connection.
- **xa-datasource-class**: The fully qualified name of the javax.sql.XADataSource implementation class, for example, “com.informix.jdbcx.IfxXADataSource”.
- **xa-datasource-property**: The xa-datasource-property element allows for specification of the properties to assign to the `XADataSource` implementation class. Each property is identified by the name attribute and the property value is given by the xa-datasource-property element content. The property is mapped onto the `XADataSource` implementation by looking for a JavaBeans style getter method for the property name. If found, the value of the property is set using the JavaBeans setter with the element text translated to the true property type using the `java.beans.PropertyEditor` for the type.
- **transaction-isolation**: This element specifies the `java.sql.Connection` transaction isolation level to use. The constants defined in the `Connection` interface are the possible element content values and include:
    - TRANSACTION_READ_UNCOMMITTED

- TRANSACTION_READ_COMMITTED
- TRANSACTION_REPEATABLE_READ
- TRANSACTION_SERIALIZABLE
- TRANSACTION_NONE

Configuring Generic JCA Adaptors

The `XSLSubDeployer` also supports the deployment of arbitrary non-JDBC JCA resource adaptors using an alternate abbreviated syntax. The schema for the top-level connection factory elements of the "*-ds.xml" configuration deployment file is shown in Figure 7-10.

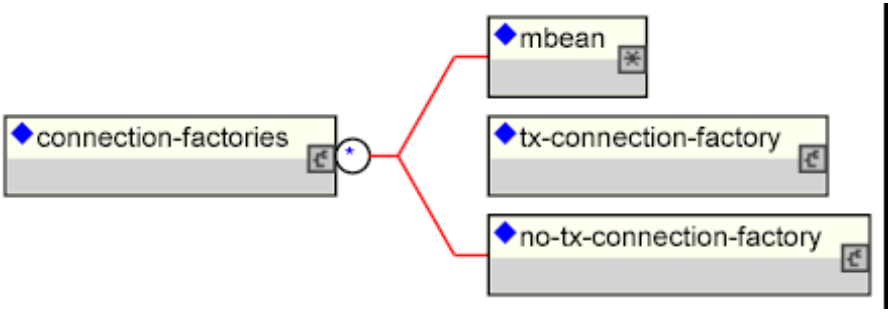

FIGURE 7-10. The simplified JCA adaptor connection factory configuration descriptor top-level schema elements

Multiple connection factory configurations may be specified in a configuration deployment file. The child elements of the connection-factories root are:

- **mbean**: Any number mbean elements may be specified to define MBean services that should be included in the jboss-service.xml descriptor that results from the transformation. This may be used to configure services used by the adaptor.
- **no-tx-connection-factory**: this element is used to specify the `org.jboss.resource.connectionmanager.NoTxConnectionManager` service configuration. `NoTxConnectionManager` is a JCA connection manager with no transaction support. The no-tx-connection-factory child element schema is given in Figure 7-11.

- **tx-connection-factory**: this element is used to specify the `org.jboss.resource.connectionmanager.TxConnectionManager` service configuration. The tx-connection-factory child element schema is given in Figure 7-12.

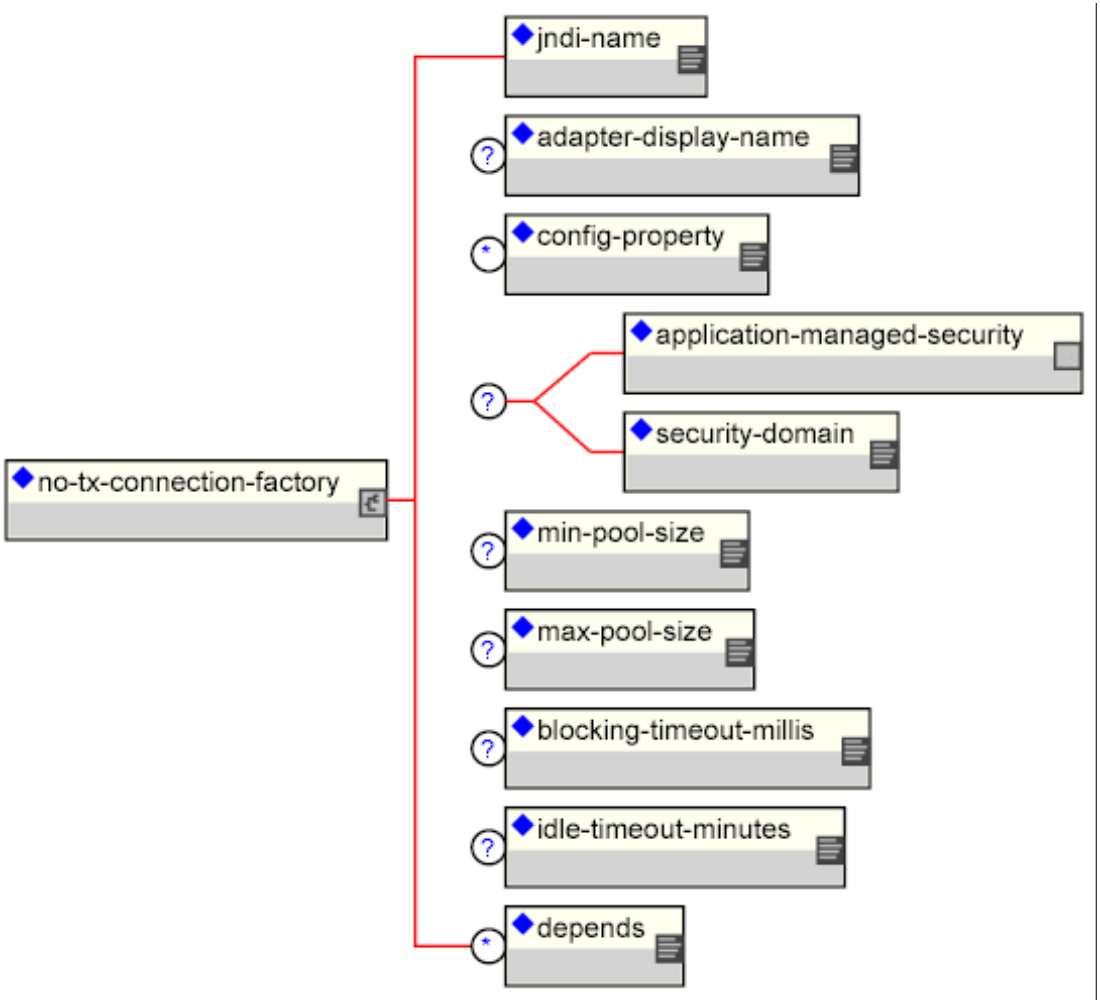

FIGURE 7-11. **The no-tx-connection-factory element schema**

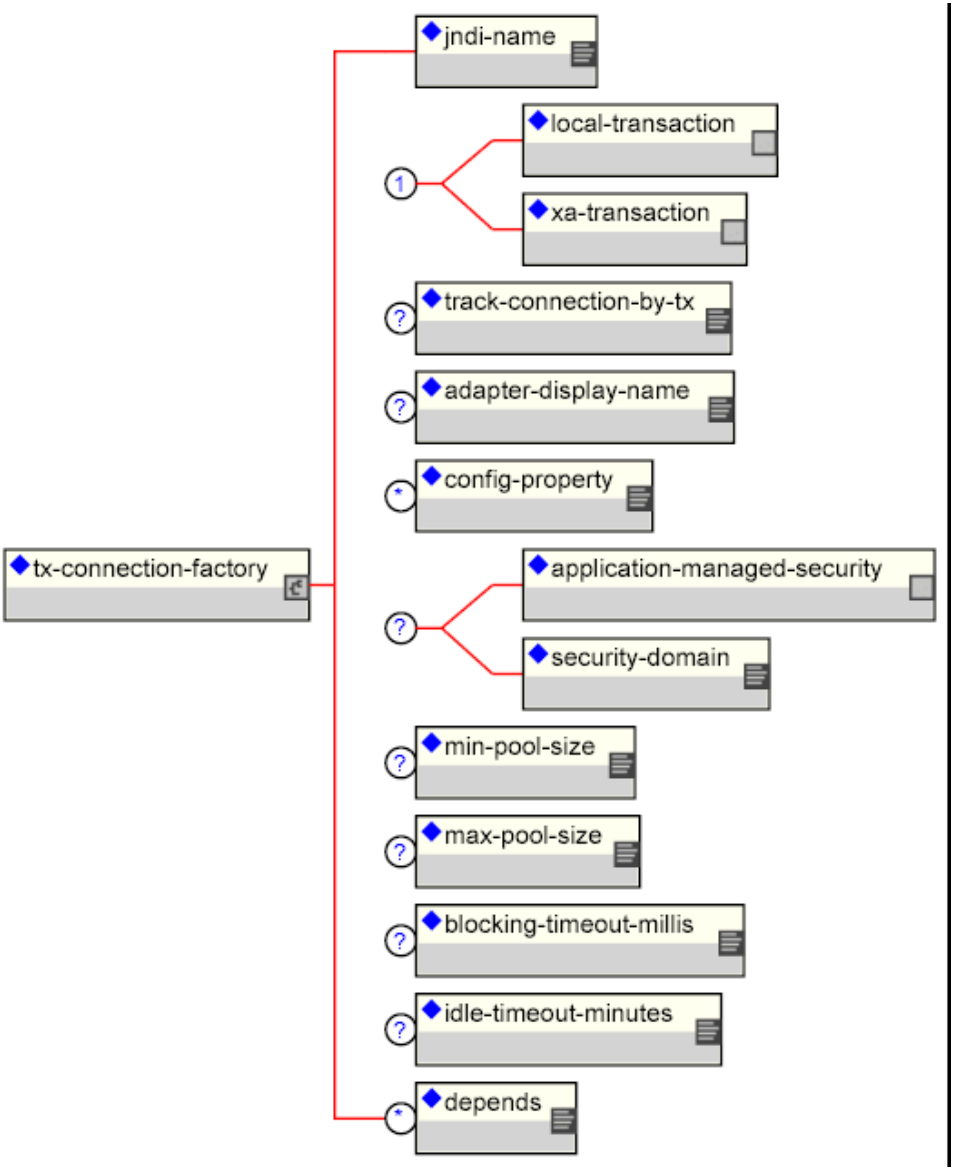

FIGURE 7-12. **The tx-connection-factory element schema**

The majority of the elements are the same as those of the datasources configuration. The element unique to the connection factory configuration include:

- **adaptor-display-name**: A human readable display name to assign to the connection manager MBean.
- **config-property**: Any number of properties to supply to the ManagedConnectionFactory (MCF) MBean service configuration. Each config-property element specifies the value of a MCF property. The config-property element has two required attributes:
  - *name*: The name of the property
  - *type*: The fully qualified type of the property
- The content of the config-property element provides the string representation of the property value. This will be converted to the true property type using the associated type `PropertyEditor`.
- **local-transaction** | **xa-transaction**: These element specify whether the tx-connection-factory supports local transaction or XA transactions.

Sample Configurations

Example configurations of many third-party JDBC drivers is included in the JBOSS_DIST/docs/examples/jca directory. Current example configurations include:

- asapxcess-jb3.2-ds.xml
- db2-ds.xml
- facets-ds.xml
- firebird-ds.xml
- generic-ds.xml
- hsqldb-ds.xml
- informix-ds.xml
- informix-xa-ds.xml
- jdatastore-ds.xml
- jms-ds.xml
- msaccess-ds.xml
- mssql-ds.xml
- mssql-xa-ds.xml
- mysql-ds.xml
- oracle-ds.xml

- oracle-xa-ds.xml
- postgres-ds.xml
- sapdb-ds.xml
- sapr3-ds.xml
- solid-ds.xml
- sybase-ds.xml