IronJacamar 1.0 User's Guide

Connecting your Enterprise Information Systems

To all Java EE Connector Architecture users, and especially the IronJacamar community

Preface

1. About IronJacamar

The goal of the IronJacamar project is to provide an implementation of the Java Connector Architecture 1.6 specification.

The specification can be found here: *http://www.jcp.org/en/jsr/detail?id=322*.

The IronJacamar project is licensed under the GNU LESSER GENERAL PUBLIC LICENSE 2.1 (LGPL 2.1) license.

2. Why IronJacamar ?

The Java EE Connector Architecture container can be viewed as a foundation inside an application server as it provides connectivity to the other containers such that they can communicate with EISes. Iron is often used as foundation in building houses too.

The Jacamar bird family which lives in Central and South America are glossy elegant birds with long bills and tails. Why we picked the Jacamar family is left as an exercise for the reader :)

3. Versions

This section contains the highlights of the IronJacamar releases. A full description of each release can be found through our issue tracking system at *http://issues.jboss.org/browse/JBJCA*.

3.1. IronJacamar 1.0

Highlights as compared to previous Java EE Connector Architecture containers inside JBoss Application Server:

- Java EE Connector Architecture 1.6 certified (standalone / Java EE6)
- POJO container environment
- New configuration schemas which focuses on usability
- Fast XML and annotation parsing for quick deployment
- Reauthentication support
- Prefill support for security backed domains
- Support for pool flushing strategies
- Embedded environment for ease of development with Arquillian and ShrinkWrap integration
- New management and statistics integration for components

- Code generator for resource adapters
- Validator tool for resource adapters

4. The team

Jesper Pedersen acts as the lead for the IronJacamar project. He can be reached at jesper (dot) pedersen (at) jboss (dot) org.

Jeff Zhang is a core developer on the IronJacamar project. He can be reached at jizhang (at) redhat (dot) com.

Stefano Maestri is a core developer on the IronJacamar project. He can be reached at stefano.maestri (at) redhat (dot) com.

Lin Gao is a core developer on the IronJacamar project. He can be reached at lgao (at) redhat (dot) com.

Vladimir Rastseluev is a core developer on the IronJacamar project. He can be reached at vrastsel (at) redhat (dot) com.

5. Thanks to

Dimitris Andreadis, Adrian Brock, Carlo de Wolf, Gurkan Erdogdu, Bruno Georges, Paul Gier, Jason Green, Stefan Guilhen, Jonathan Halliday, Søren Hilmer, Ales Justin, Vicky Kak, Aslak Knutsen, Sacha Labourey, Johnaton Lee, Mark Little, Alexey Loubyansky, Patrick MacDonald, Scott Marlow, Shelly McGowan, Andrig Miller, Marcus Moyses, Weston Price, Andrew Lee Rubinger, Heiko Rupp, Anil Saldhana, Scott Stark, Clebert Suconic, Andy Taylor, Jeremy Whiting, Tyronne Wickramarathne and Yang Yong.

6. License

1

Introduction

The Java Connector Architecture (JCA) defines a standard architecture for connecting the Java EE platform to heterogeneous Enterprise Information Systems (EIS). Examples of EISs include Enterprise Resource Planning (ERP), mainframe transaction processing (TP), databases and messaging systems.

The connector architecture defines a set of scalable, secure, and transactional mechanisms that enable the integration of EISs with application servers and enterprise applications.

The connector architecture also defines a Common Client Interface (CCI) for EIS access. The CCI defines a client API for interacting with heterogeneous EISs.

The connector architecture enables an EIS vendor to provide a standard resource adapter for its EIS. A resource adapter is a system-level software driver that is used by a Java application to connect to an EIS. The resource adapter plugs into an application server and provides connectivity between the EIS, the application server, and the enterprise application. The resource adapter serves as a protocol adapter that allows any arbitrary EIS communication protocol to be used for connectivity. An application server vendor extends its system once to support the connector architecture and is then assured of seamless connectivity to multiple EISs. Likewise, an EIS vendor provides one standard resource adapter which has the capability to plug in to any application server that supports the connector architecture.

1.1. What's New

The Java Connector Architecture 1.6 specification adds the following major areas:

- Ease of Development: The use of annotations reduces or completely eliminates the need to deal with a deployment descriptor in many cases. The use of annotations also reduces the need to keep the deployment descriptor synchronized with changes to source code.
- Generic work context contract: A generic contract that enables a resource adapter to control the execution context of a Work instance that it has submitted to the application server for execution.
- Security work context: A standard contract that enables a resource adapter to establish security information while submitting a Work instance for execution to a WorkManager and while delivering messages to message endpoints residing in the application server.
- Standalone Container Environment: A defined set of services that makes up a standalone execution environment for resource adapters.

1.2. Overview

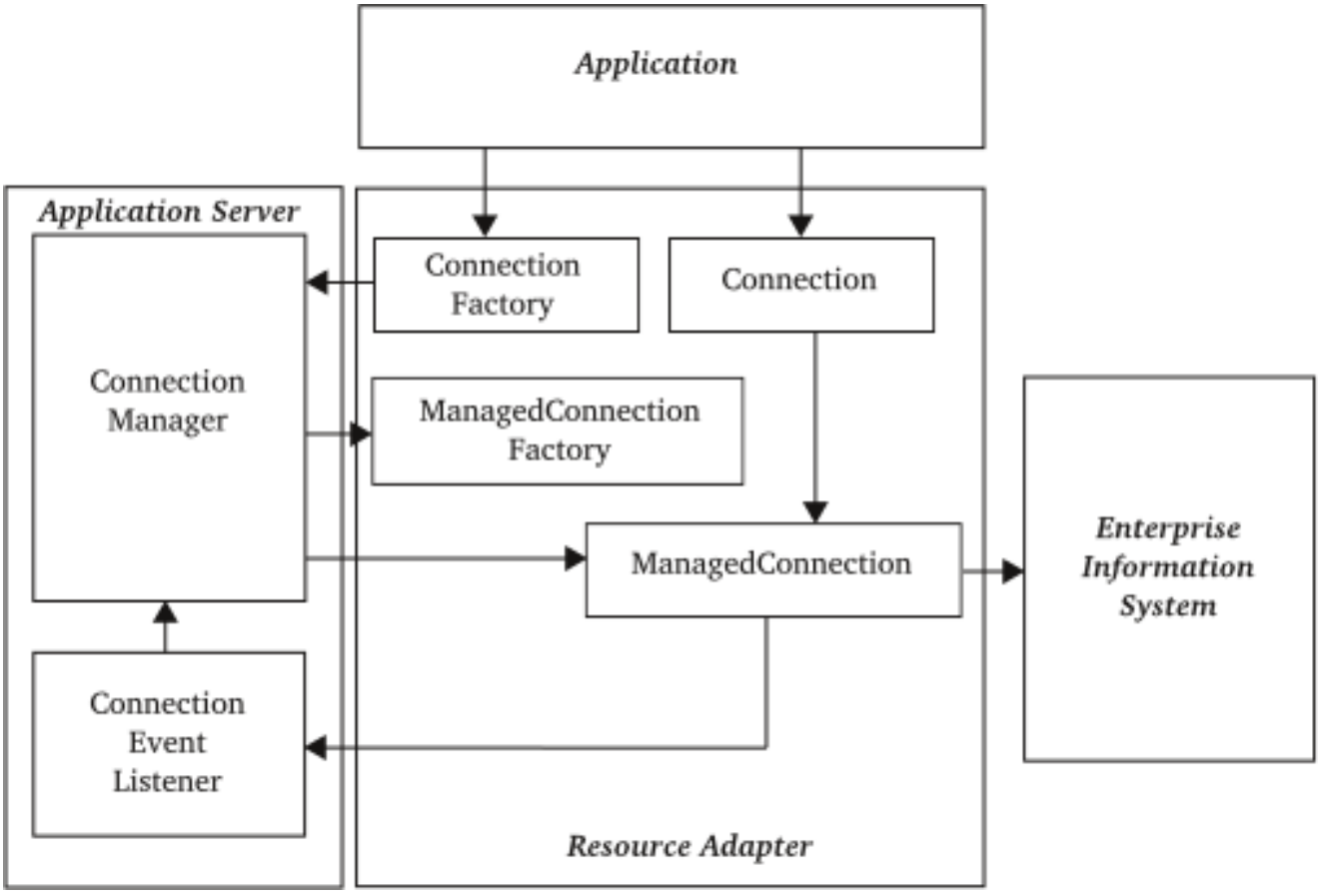
The Java EE Connector Architecture features three different types of resource adapters

- Outbound: The resource adapter allows the application to communicate to the Enterprise Information System (EIS).
- Inbound: The resource adapter allows messages to flow from the Enterprise Information System (EIS) to the application.
- Bi-directional: The resource adapter features both an outbound and an inbound part.

For more information about Java EE Connector Architecture see the specification.

1.2.1. Outbound resource adapter

The Java Connector Architecture specification consists of a number of outbound components:



The application uses the

- ConnectionFactory: The connection factory is looked up in Java Naming and Directory Interface (JNDI) and is used to create a connection.

- Connection: The connection contains the Enterprise Information System (EIS) specific operations.

The resource adapter contains

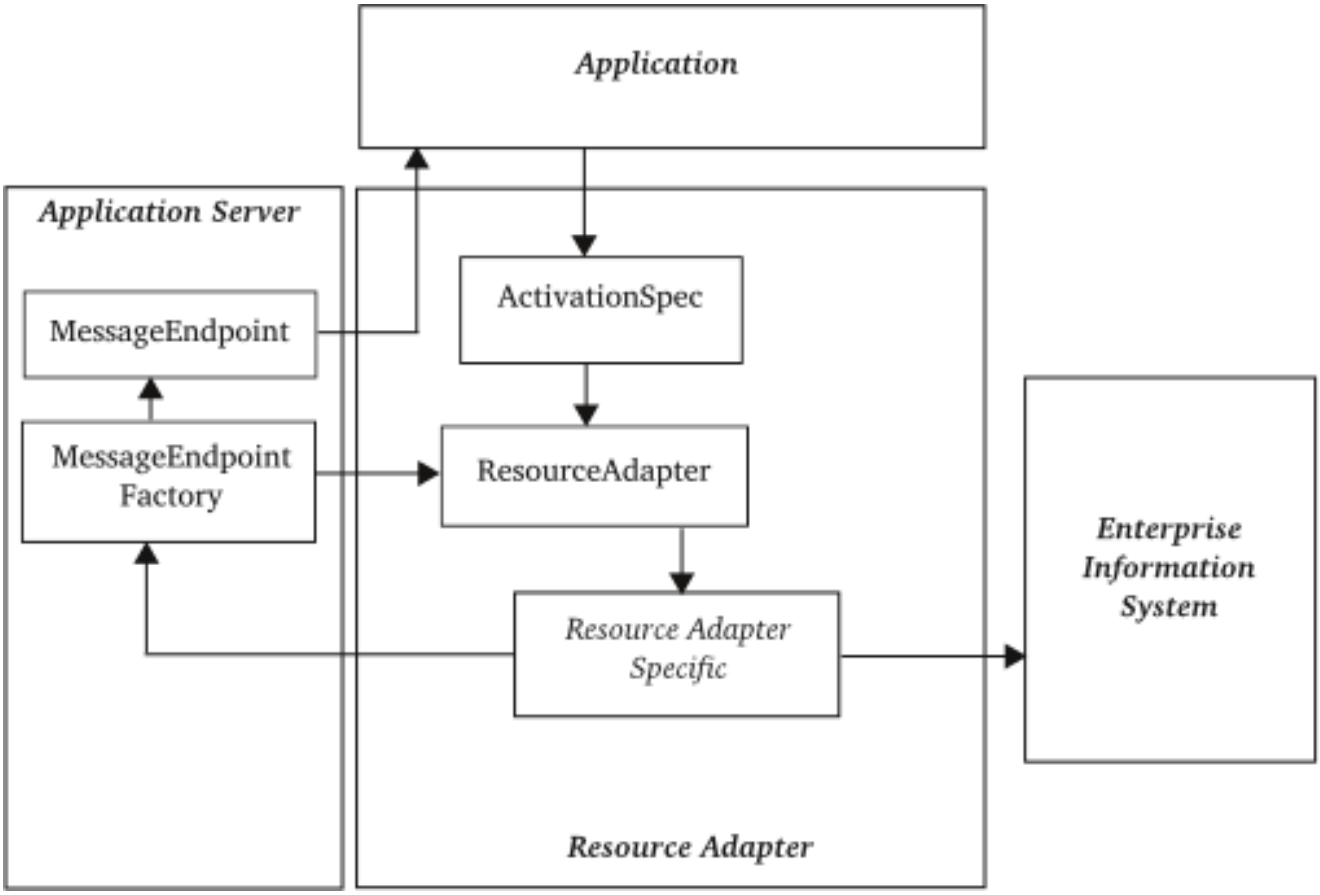
- ManagedConnectionFactory: The managed connection factory creates managed connections.
- ManagedConnection: The managed connection represents a physical connection to the target Enterprise Information System (EIS). The managed connection notifies the application server of events such as connection closed and connection error.

IronJacamar - the application server - contains

- ConnectionManager: The connection manager handles all managed connections in regards to pooling, transaction and security.
- ConnectionEventListener: The connection event listener allows the connection manager to know the status of each managed connection.

1.2.2. Inbound resource adapter

The Java Connector Architecture specification consists of a number of inbound components:



The application uses the

- ActivationSpec: The activation specification specifies the different properties that the application is looking for from the resource adapter and hence the Enterprise Information System (EIS). This specification can be hidden from the user by a facade provided by the application server.

The resource adapter contains

- ResourceAdapter: The resource adapter provides the activation point for inbound communication.
- Resource adapter specific: The resource adapter specific code handles communication with the Enterprise Information System (EIS) and deliver messages through the MessageEndpointFactory.

IronJacamar - the application server - contains

- MessageEndpointFactory: The MessageEndpointFactory is registered with the ResourceAdapter instance and creates the MessageEndpoint instances.
- MessageEndpoint: The MessageEndpoint contains the actual message from the Enterprise Information System (EIS) which the application uses. This could for example be a message driven Enterprise JavaBean (EJB/MDB).

2

Download

The official IronJacamar project page is *http://www.jboss.org/ironjacamar/* where you can download the software.

2.1. Download

The download location is: *http://www.jboss.org/ironjacamar/downloads/*

Each release is labelled with a version number and an identifier.

```
ironjacamar-<major>.<minor>.<patch>.<identifier>
```

where

- Major: The major version number. Signifies major changes in the implementation.
- Minor: The minor version number. Signifies functional changes to a major version.
- Patch: The patch version number. Signifies a binary compatible change to a minor version.
- Identifier: The identifier. Identifies the level of the quality of the release.
  - Final: Stable release
  - CR: Candidate for Release quality. The implementation is functional complete.
  - Beta: Beta quality. The implementation is almost functional complete.
  - Alpha: Alpha quality. The implementation is a snapshot of the development.

An example

```
ironjacamar-1.0.0.Final.tar.gz
```

which is the first stable release of the project.

2.2. Maven repository

The IronJacamar distribution is deployed to the JBoss Nexus repository.

Repository: `http://repository.jboss.org/nexus/content/groups/public/`

Group id: `org.jboss.ironjacamar`

Table 2.1. Maven artifacts

Artifact	Description
`ironjacamar-as`	JBoss Application Server integration tools
`ironjacamar-codegenerator`	The code generator
`ironjacamar-common-api`	The API for the common module
`ironjacamar-common-impl`	The implementation for the common module
`ironjacamar-common-impl-papaki`	The Papaki extension for the common module
`ironjacamar-common-spi`	The SPI for the common module
`ironjacamar-core-api`	The API / SPI for the core module
`ironjacamar-core-impl`	The implementation for the core module
`ironjacamar-depchain`	The dependency chain for the IronJacamar container
`ironjacamar-deployers-common`	The common classes for the deployer chains
`ironjacamar-deployers-fungal`	The deployers for the Fungal kernel based setup
`ironjacamar-embedded`	The embedded module
`ironjacamar-embedded-arquillian`	The Arquillian extension for the embedded module
`ironjacamar-jdbc`	The core library for the JDBC resource adapters
`ironjacamar-spec-api`	The Java EE Connector Architecture 1.6 API
`ironjacamar-validator`	The validator module
`ironjacamar-validator-ant`	The Apache Ant tasks for the validator module
`ironjacamar-validator-cli`	The command line interface for the validator module
`jdbc-local`	A JDBC resource adapter backing standard datasources
`jdbc-xa`	A JDBC resource adapter backing XA datasources
`mail`	An inflow mail resource adapter

2.3. SVN Access

If you want to experiment with the latest developments you may checkout the latest code from SVN trunk. Be aware that the information provided in this manual might then not be accurate.

The anonymous SVN repository is located under:

```
svn co http://anonsvn.jboss.org/repos/jbossas/projects/jboss-jca/trunk/ ironjacamar-trunk
```

You can find additional information about this in the developer guide.

3

Installation

Once you have downloaded the distribution you need to install it in a location of your choice.

3.1. Compressed Tape Archive (.tar.gz)

Extract the distribution using

```
tar xzf ironjacamar-1.0.0.Final.tar.gz
```

The distribution will be located in a directory named

```
ironjacamar-1.0.0.Final
```

3.2. Zip Archive (.zip)

Extract the distribution using

```
unzip ironjacamar-1.0.0.Final.zip
```

or any program capable of handling Zip archives such as WinZip and WinRar.

The distribution will be located in a directory named

```
ironjacamar-1.0.0.Final
```

3.3. Directory structure

The IronJacamar container has the following directory structure:

- bin: Contains the scripts that starts the container.
- config: Contains the configuration of the container.
- deploy: Contains user deployments.
- doc: Contains the documentation.
- lib: Contains all the libraries needed by the container.
- log: Contains the log files for the container.
- system: Contains system deployments.
- tmp: Contains temporary files.

3.4. JBoss Application Server

The IronJacamar provides the Java EE Connector Architecture (JCA) container for JBoss Application Server 7 and future versions.

The container can be updated in the JBoss Application Server by using the `as-upgrader.sh` script in the `doc/as` directory. This will allow an easy installation of IronJacamar patch releases to fix bugs in the application server environment.

The script can be used, like:

```
./as-upgrader.sh 1.0.0.Final /path/to/as7/installation
```

where `1.0.0.Final` is the version identifier of the IronJacamar container and the path points to the top-level directory of the JBoss Application Server installation.

You can get an overview of all IronJacamar releases by searching our *Nexus* [https://repository.jboss.org/nexus/] repository.

Warning

Make sure that you understand the version policies specified in the developer guide before upgrading

4

Configuration

The configuration for the IronJacamar container is located in the config/ directory.

4.1. Logging service

The IronJacamar container uses JBoss Logging framework as the implementation.

The configuration is done in the

```
config/logging.properties
```

file.

Consult the *JBoss Logging documentation* [http://www.jboss.org/community/wiki/JBossBootLogging] on how the service can be configured.

4.2. Transaction service

The IronJacamar container uses the JBoss Transaction Manager as its transaction implementation.

The configuration is done in the

```
config/transaction.xml
```

file.

Consult the JBoss Transaction documentation on how the service can be configured.

4.3. JCA

4.3.1. Deployer

The IronJacamar deployer is configured in the

```
config/bootstrap/jca.xml
```

file.

4.3.1.1. Configuration

The configuration of the resource adapter deployer chain is handled by a `org.jboss.jca.deployers.fungal.RAConfiguration` bean.

```
<bean name="RAConfiguration"
      class="org.jboss.jca.deployers.fungal.RAConfiguration">
  <property name="ArchiveValidation">true</property>
  <property name="ArchiveValidationFailOnWarn">false</property>
  <property name="ArchiveValidationFailOnError">true</property>
  <property name="BeanValidation">true</property>
  <property name="PrintStream">
    <inject bean="JBossStdioContext" property="Out"/>
  </property>
  <property name="DefaultBootstrapContext">
    <inject bean="DefaultBootstrapContext"/>
  </property>
  <property name="JndiStrategy"><inject bean="JndiStrategy"/></property>
  <property name="TransactionManager">
    <inject bean="RealTransactionManager"/>
  </property>
  <property name="MetadataRepository"><inject bean="MDR"/></property>
</bean>
```

Table 4.1. Resource adapter deployer configuration

Property	Type	Description
`ArchiveValidation`	`boolean`	Toggle archive validation for the deployment units. Default: `true`
`ArchiveValidation FailOnWarn`	`boolean`	Should an archive validation warning report fail the deployment.

Property	Type	Description
		Default: `false`
`ArchiveValidation FailOnError`	`boolean`	Should an archive validation error report fail the deployment. Default: `true`
`BeanValidation`	`boolean`	Toggle bean validation (JSR-303) for the deployment units. Default: `true`
`DefaultBootstrap Context`	`org.jboss.jca. core.api.bootstrap. CloneableBootstrap Context`	Specifies the default bootstrap context for resource adapters
`BootstrapContexts`	`Map<String, org.jboss.jca. core.api.bootstrap. CloneableBootstrap Context>`	Bootstrap context map (unique name to a cloneable bootstrap context) which allows developers to bind (through `ironjacamar.xml`) their resource adapter to a specific bootstrap context instance.
`PrintStream`	`java.io.PrintStream`	Specifies which print stream that should be used to handle the `LogWriterS`
`MetadataRepository`	`org.jboss.jca. core.spi.mdr. MetadataRepository`	The metadata repository
`ResourceAdapterRepository`	`org.jboss.jca. core.spi.rar. ResourceAdapterRepository`	The resource adapter repository
`ScopeDeployment`	`boolean`	Should each deployment be scoped (isolated) from the container. This feature allows deployment of libraries of a different version than used in the container environment. Default: `false`
`JndiStrategy`	`org.jboss.jca. core.spi.naming. JndiStrategy`	Specifies the JNDI strategy policy for binding the connection factories into the naming environment The JNDI strategies are located in the `org.jboss.jca.core.naming` package

Property	Type	Description
		• `NoopJndiStrategy`: A no operation JNDI strategy which doesn't bind/unbind any objects • `SimpleJndiStrategy`: A simple JNDI strategy which can bind/unbind a single connection factory • `ExplicitJndiStrategy`: A JNDI strategy which can requires explicit JNDI names to bind/unbind a connection factory

4.3.1.2. Resource adapter deployer

The initial deployer for resource adapter archives is handled by a `org.jboss.jca.deployers.fungal.RADeployer` bean.

```
<bean name="RADeployer"
      interface="com.github.fungal.spi.deployers.Deployer"
      class="org.jboss.jca.deployers.fungal.RADeployer">
  <property name="Configuration"><inject bean="RAConfiguration"/></property>
  <depends>BeanValidation</depends>
  <depends>JBossStdioContextSelector</depends>
</bean>
```

This deployer will register the resource adapters with the metadata repository in the system.

Table 4.2. Resource adapter deployer

Property	Type	Description
`Configuration`	`org.jboss.jca.deployers.fungal.RAConfiguration`	The configuration for the deployer

4.3.1.3. Resource adapter metadata deployer

The deployer for deploying our `-ra.xml` deployment descriptor is handled by a `org.jboss.jca.deployers.fungal.RaXmlDeployer` bean.

The deployment descriptor is defined by the `resource-adapters-1_0.xsd` schema.

```
<bean name="RaXmlDeployer"
      interface="com.github.fungal.spi.deployers.Deployer"
      class="org.jboss.jca.deployers.fungal.RaXmlDeployer">
  <property name="Configuration"><inject bean="DeployerConfiguration"/></property>
  <property name="Kernel"><inject bean="Kernel"/></property>
  <depends>BeanValidation</depends>
  <depends>JBossStdioContextSelector</depends>
</bean>
```

This deployer will activate resource adapters based on the deployment information.

Table 4.3. Resource adapter metadata deployer

Property	Type	Description
`Configuration`	`org.jboss.jca.deployers.fungal.RAConfiguration`	The configuration for the deployer

4.3.1.4. Resource adapter activator

The deployer chain features an activator for resource adapter archives is handled by the `org.jboss.jca.deployers.fungal.RAActivator` bean.

```
<bean name="RAActivator"
      class="org.jboss.jca.deployers.fungal.RAActivator">
  <property name="Configuration"><inject bean="RAConfiguration"/></property>
  <property name="Kernel"><inject bean="Kernel"/></property>
  <property name="ExcludeArchives">
    <set elementClass="java.lang.String">
      <value>jdbc-local.rar</value>
      <value>jdbc-xa.rar</value>
    </set>
  </property>
  <depends>BeanValidation</depends>
  <depends>JBossStdioContextSelector</depends>
</bean>
```

This activator will activate any resource adapters which hasn't been activated yet unless they are in the excluded list.

Table 4.4. Resource adapter activator

Property	Type	Description
`Configuration`	`org.jboss.jca.deployers.fungal.RAConfiguration`	The configuration for the deployer
`Enabled`	`boolean`	Should the activator be enabled. Default is `true`
`Kernel`	`com.github.fungal.api.Kernel`	The kernel instance
`ExcludeArchives`	`java.util.Set`	A set of resource adapter archives which should be excluded from activation

4.3.2. Security

The Java EE Connector Architecture 1.6 specification allows units of `javax.resource.spi.Work` to be executed in a specific security context.

This is done through the use of Java Authentication Service Provider Interface for Containers (JSR-196) call backs using the `javax.security.auth.callback.Callback` interface.

The support is activated by letting the work instance implement the

```
javax.resource.spi.work.WorkContextProvider
```

interface and returning an instance of `javax.resource.spi.work.SecurityContext`.

There is currently support for injecting a callback setup based on the file

```
config/callback.properties
```

The format of the `callback.properties` file is described in the file.

The callback setup can be configured through the `Callback` bean in the `config/bootstrap/jca.xml` file.

```
<!-- Callback -->
```

```
<bean name="Callback"
      interface="org.jboss.jca.core.spi.security.Callback"
      class="org.jboss.jca.core.security.DefaultCallback">
  <property name="File">${iron.jacamar.home}/config/callback.properties</property>
</bean>
```

There is support for creating a basic security domain which can provide a `javax.security.auth.Subject` instance to deployments that are using `<security-domain>` or `<security-domain-and-application>` in their setup.

A security domain can be configured through

```
<!-- SubjectFactory -->
<bean name="DefaultSecurityDomain"
      interface="org.jboss.security.SubjectFactory"
      class="org.jboss.jca.core.security.DefaultSubjectFactory">
  <property name="SecurityDomain">DefaultSecurityDomain</property>
  <property name="UserName">user</property>
  <property name="Password">password</property>
</bean>
```

beans.

4.4. Datasources

The IronJacamar project can deploy datasources using the `datasources-1_0.xsd` or `datasources-1_1.xsd` schemas.

The configuration is done in the

```
config/bootstrap/ds.xml
```

file.

Table 4.5. DsXmlDeployer

Property	Type	Description
`JDBCLocal`	`String`	The name of the `jdbc-local.rar` deployment

Property	Type	Description
`JDBCXA`	`String`	The name of the `jdbc-xa.rar` deployment
`TransactionManager`	`javax.transaction.TransactionManager`	The transaction manager
`MetadataRepository`	`org.jboss.jca.core.spi.mdr.MetadataRepository`	The metadata repository
`Kernel`	`com.github.fungal.api.Kernel`	The kernel

The datasource deployer can be removed from the environment by removing the `ds.xml` file in

```
config/bootstrap/
```

as well as the reference in `config/bootstrap/bootstrap.xml` to the file.

Furthermore all `jdbc-*.rar` files in the `system/` directory should be removed too.

4.5. Web server

The IronJacamar project features a web server which is used to serve web archive deployments. More information about Jetty can be found at the *homepage* [http://www.eclipse.org/jetty/].

The configuration is done in the

```
system/web.xml
```

file.

```
<bean name="WebServer" class="org.jboss.jca.web.WebServer">
  <property name="Host">${iron.jacamar.bindaddress:localhost}</property>
  <property name="Port">8080</property>
 <property name="ExecutorService"><inject bean="Kernel" property="ExecutorService"/></property>
</bean>
```

Table 4.6. Web server

Property	Type	Description
`Host`	`String`	Set the bind address for the web server Default: `localhost`
`Port`	`int`	Set the port for the web server Default: `8080`
`AcceptQueueSize`	`int`	Set the accept queue size for the Jetty connector Default: `64`
`ExecutorService`	`java.util.concurrent.ExecutorService`	The thread pool for the web server Default: The kernel thread pool

The web server can be removed from the environment by removing the `web.xml` file in

```
system/
```

Furthermore all `.war` files in the same directory should be removed too.

All the Jetty libraries can be removed by deleting the

```
lib/jetty
```

directory.

5

Deployment

The IronJacamar distribution contains a deploy/ directory where all deployments should be deployed to.

5.1. Packaging requirements

A resource adapter archive is a structured Java Archive (JAR) file, which bundles all Java classes in JAR files, and optionally contains metadata, resources and native libraries.

A resource adapter archive name ends in the `.rar` extension.

An example of a resource adapter archive could look like

```
[jpederse@localhost]$ jar tf ra.rar
META-INF/ra.xml
readme.html
ra.jar
images/icon.jpg
win.dll
linux.so
```

See the Java EE Connector Architecture 1.6 specification chapter 20 for further requirements.

5.2. Deploying resource adapters

Resource adapters (.rar) are deployed by copying the resource adapter into the `deploy/` directory

```
cp example.rar ironjacamar-1.0.0.Final/deploy
```

on a Un*x based system or

```
copy example.rar ironjacamar-1.0.0.Final\deploy
```

on Windows.

The resource adapter can be configured and activated through a `META-INF/ironjacamar.xml` file in the archive. The format of the XML document is defined by the `ironjacamar_1_0.xsd` schema.

A resource adapter can also be configured and activated through deployment of a `-ra.xml` file in the `deploy/` directory - f.ex. `deploy/example-ra.xml`. The format of the XML document is defined by the `resource-adapters_1_0.xsd` schema - f.ex

```
<resource-adapters>
  <resource-adapter>
    <archive>example.rar</archive>
    <connection-definitions>
      <connection-definition jndi-name="java:/eis/example" class-name="com.example.ra.MCF"/>
    </connection-definitions>
  </resource-adapter>
</resource-adapters>
```

to bind the connection factory from `com.example.ra.MCF` under `java:/eis/example`.

See the schema appendix for additional details about the format.

Alternative the resource adapter deployments will be picked up by the `RAActivator` bean which bind a single connection factory under

```
java:/eis/<deploymentName>
```

- f.ex. `java:/eis/example` and a single admin object under

```
java:/eis/ao/<deploymentName>
```

- f.ex. `java:/eis/ao/example`.

5.2.1. Resource adapter descriptor

A resource adapter can be configured using two different ways

- `META-INF/ironjacamar.xml` for internal configuration
- `-ra.xml` for external configuration

to the resource adapter archive. Both formats share the same layout to ease configuration - only the top-level elements differ.

Table 5.1. Main elements

Element	Desciption
`bean-validation-groups`	Specifies bean validation group that should be used
`bootstrap-context`	Specifies the unique name of the bootstrap context that should be used
`config-property`	The config-property specifies resource adapter configuration properties.
`transaction-support`	Define the type of transaction supported by this resource adapter. Valid values are: NoTransaction, LocalTransaction, XATransaction
`connection-definitions`	Specifies the connection definitions
`admin-objects`	Specifies the administration objects

Table 5.2. Bean validation groups elements

Element	Desciption
`bean-validation-group`	Specifies the fully qualified class name for a bean validation group that should be used for validation

Table 5.3. Connection definition / admin object attributes

Attribute	Desciption
`class-name`	Specifies the the fully qualified class name of a managed connection factory or admin object
`jndi-name`	Specifies the JNDI name
`enabled`	Should the object in question be activated
`use-java-context`	Specifies if a java:/ JNDI context should be used
`pool-name`	Specifies the pool name for the object
`use-ccm`	Enable the cache connection manager

Table 5.4. Connection definition elements

Element	Desciption
`config-property`	The config-property specifies managed connection factory configuration properties.
`pool`	Specifies pooling settings
`xa-pool`	Specifies XA pooling settings
`security`	Specifies security settings
`timeout`	Specifies time out settings
`validation`	Specifies validation settings
`recovery`	Specifies the XA recovery settings

Table 5.5. Pool elements

Element	Desciption
`min-pool-size`	The min-pool-size element indicates the minimum number of connections a pool should hold. These are not created until a Subject is known from a request for a connection. This default to 0
`max-pool-size`	The max-pool-size element indicates the maximum number of connections for a pool. No more than max-pool-size connections will be created in each sub-pool. This defaults to 20.
`prefill`	Whether to attempt to prefill the connection pool. Default is false
`use-strict-min`	Specifies if the min-pool-size should be considered strictly. Default false
`flush-strategy`	Specifies how the pool should be flush in case of an error. Valid values are: FailingConnectionOnly (default), IdleConnections, EntirePool

Table 5.6. XA pool elements

Element	Desciption
`min-pool-size`	The min-pool-size element indicates the minimum number of connections a pool should hold. These are not created until a Subject is known from a request for a connection. This default to 0
`max-pool-size`	The max-pool-size element indicates the maximum number of connections for a pool. No more than max-pool-size connections will be created in each sub-pool. This defaults to 20.

Element	Desciption
`prefill`	Whether to attempt to prefill the connection pool. Default is false
`use-strict-min`	Specifies if the min-pool-size should be considered strictly. Default false
`flush-strategy`	Specifies how the pool should be flush in case of an error. Valid values are: FailingConnectionOnly (default), IdleConnections, EntirePool
`is-same-rm-override`	The is-same-rm-override element allows one to unconditionally set whether the javax.transaction.xa.XAResource.isSameRM(XAResource) returns true or false
`interleaving`	An element to enable interleaving for XA connection factories
`no-tx-separate-pools`	Oracle does not like XA connections getting used both inside and outside a JTA transaction. To workaround the problem you can create separate sub-pools for the different contexts
`pad-xid`	Should the Xid be padded
`wrap-xa-resource`	Should the XAResource instances be wrapped in a org.jboss.tm.XAResourceWrapper instance

Table 5.7. Security elements

Element	Desciption
`application`	Indicates that application supplied parameters (such as from getConnection(user, pw)) are used to distinguish connections in the pool.
`security-domain`	Indicates Subject (from security domain) are used 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.
`security-domain-and-application`	Indicates that either application supplied parameters (such as from getConnection(user, pw)) or Subject (from security domain) are used 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.

Table 5.8. Time out elements

Element	Desciption
`blocking-timeout-millis`	The blocking-timeout-millis element indicates 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 30000 (30 seconds).
`idle-timeout-minutes`	The idle-timeout-minutes elements indicates 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.
`allocation-retry`	The allocation retry element indicates the number of times that allocating a connection should be tried before throwing an exception. The default is 0.
`allocation-retry-wait-millis`	The allocation retry wait millis element indicates the time in milliseconds to wait between retrying to allocate a connection. The default is 5000 (5 seconds).
`xa-resource-timeout`	Passed to XAResource.setTransactionTimeout(). Default is zero which does not invoke the setter. Specified in seconds

Table 5.9. Validation elements

Element	Desciption
`background-validation`	An element to specify that connections should be validated on a background thread versus being validated prior to use
`background-validation-minutes`	The background-validation-minutes element specifies the amount of time, in minutes, that background validation will run.
`use-fast-fail`	Whether fail a connection allocation on the first connection if it is invalid (true) or keep trying until the pool is exhausted of all potential connections (false). Default is false

Table 5.10. Admin object elements

Element	Desciption
`config-property`	Specifies an administration object configuration property.

Table 5.11. Recovery elements

Element	Desciption
`recover-credential`	Specifies the user name / password pair or security domain that should be used for recovery.
`recover-plugin`	Specifies an implementation of the org.jboss.jca.core.spi.recovery.RecoveryPlugin class.

The deployment schemas are defined in `doc/ironjacamar_1_0.xsd` and `doc/resource-adapters_1_0.xsd`.

5.2.2. Resource adapter extensions

A resource adapter can make use of a couple of Java EE Connector Architecture extensions in the IronJacamar container in order to improve the integration.

The extensions include

- `org.jboss.jca.core.spi.recovery.RecoveryPlugin`: Plugin to provide feedback to the recovery module inside IronJacamar.
- `org.jboss.jca.core.spi.statistics.Statistics`: Plugin to identify a resource adapter component (`ResourceAdapter`, `ManagedConnectionFactory` and admin object) that provides statistics.

The following sections will describe these extensions points.

5.2.2.1. Recovery extension

The IronJacamar recovery extension allows the resource adapter deployment to give feedback to the container if a `ManagedConnection` can be used for recovery. This extension is used as part of XA recovery in the environment, and should therefore be implemented by all resource adapters capable of working in an `XATransaction` semantics.

The interface `org.jboss.core.spi.recovery.RecoveryPlugin` located in the `ironjacamar-core-api` artifact makes up the SPI for the extension.

The interface contains two methods that should be implemented in a resource adapter specific manner.

The method

```
public boolean isValid(Object c) throws ResourceException;
```

will return `true` if the connection can be used for recovery.

The method

```
public void close(Object c) throws ResourceException;
```

will close a connection that was used for recovery.

The recovery extension is activated by adding a recovery element to the deployment

```
<recovery>
   <recovery-plugin>com.mycompany.myproject.RecoveryPluginImpl</recovery-plugin>
</recovery>
```

The following recovery plugins are provided by IronJacamar

- `org.jboss.jca.core.recovery.DefaultRecoveryPlugin`: Default recovery plugin that tries to call a `close()` method on the underlying object
- `org.jboss.jca.core.recovery.ConfigurableRecoveryPlugin`: A recovery plugin where the results of the `isValid` and `close` can be specified
- `org.jboss.jca.core.recovery.ValidatingManagedConnectionFactoryRecoveryPlugin`: A recovery plugin that uses the `javax.resource.spi.ValidatingManagedConnectionFactory` interface to verify the connection

Note

The IronJacamar container will use a default implementation of the recovery SPI if an implementation isn't specified by the deployment.

5.2.2.2. Statistics extension

The IronJacamar statistics extension allows a resource adapter to expose statistics to the container and hence to the environment where IronJacamar is running. Statistics can be enabled for `ResourceAdapter`, `ManagedConnectionFactory` and admin object instances.

The extension include two interfaces `org.jboss.core.spi.statistics.Statistics` and `org.jboss.core.spi.statistics.StatisticsPlugin`. Both these interfaces are located in the `ironjacamar-core-api` artifact.

The `Statistics` interface will mark a resource adapter component as statistics capable and return the statistics plugin implementation instance.

The `StatisticsPlugin` interface contains methods to expose and describe each statistic that the plugin makes available. This information will then be made available to the environment where the IronJacamar container is running using the environment's prefered mechanism.

Note

The IronJacamar container will only expose core statistics for a deployment if no implementation of this extension is available.

5.3. Deploying datasources

Datasources (-ds.xml) are deployed by copying the definition into the `deploy/` directory

```
cp postgres-xa-ds.xml ironjacamar-1.0.0.Final/deploy
```

on a Un*x based system or

```
copy postgres-xa-ds.xml ironjacamar-1.0.0.Final\deploy
```

on Windows.

You will need to install the database JDBC driver into the `lib/` directory.

You can find examples of datasource definitions in the `doc/datasources` directory and the schemas: `doc/datasources_1_0.xsd` and `doc/datasources_1_1.xsd`.

5.3.1. Datasource descriptor

Datasource descriptors are divided into

- `<datasource>` for a standard datasource
- `<xa-datasource>` for an XA capable datasource

definitions.

A datasource descriptor supports the following parameters.

Table 5.12. Common datasource attributes

Attribute	Desciption
`jndi-name`	Specifies the JNDI name for the datasource
`pool-name`	Specifies the pool name for the datasource used for management
`enabled`	Specifies if the datasource should be enabled
`use-java-context`	Setting this to false will bind the DataSource into global JNDI
`spy`	Enable spy functionality on the JDBC layer - e.g. log all JDBC traffic to the datasource. The logging category `org.jboss.jdbc` must be enabled too.
`use-ccm`	Enable the cached connection manager
`jta`	Enable JTA integration (only `<datasource>`)

Table 5.13. `datasource` elements

Element	Desciption
`connection-url`	The JDBC driver connection URL
`driver-class`	The fully qualifed name of the JDBC driver class
`datasource-class`	The fully qualifed name of the JDBC datasource class
`driver`	An unique name for the JDBC driver specified in the drivers section. Or the name of the .jar file if deployed as standalone deployment This element is mandatory when deploying in JBoss Application Server
`connection-property`	The connection-property element allows you to pass in arbitrary connection properties to the 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
`new-connection-sql`	Specify an SQL statement to execute whenever a connection is added to the connection pool
`transaction-isolation`	Set java.sql.Connection transaction isolation level to use. The constants defined by transaction-isolation-values are the possible transaction isolation levels and include: TRANSACTION_READ_UNCOMMITTED TRANSACTION_READ_COMMITTED

Element	Desciption
	TRANSACTION_REPEATABLE_READ TRANSACTION_SERIALIZABLE TRANSACTION_NONE
`url-delimiter`	Specifies the delimeter for URLs in connection-url for HA datasources
`url-selector-strategy-class-name`	A class that implements org.jboss.jca.adapters.jdbc.URLSelectorStrategy
`pool`	Specifies the pooling settings
`security`	Specifies the security settings
`validation`	Specifies the validation settings
`timeout`	Specifies the time out settings
`statement`	Specifies the statement settings

Table 5.14. `xa-datasource` elements

Element	Desciption
`xa-datasource-property`	Specifies a property 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
`xa-datasource-class`	The fully qualifed name of the javax.sql.XADataSource implementation class
`driver`	An unique name for the JDBC driver specified in the drivers section. Or the name of the .jar file if deployed as standalone deployment. This element is mandatory when deploying in JBoss Application Server
`url-delimiter`	Specifies the delimeter for URLs in the connection url for HA datasources
`url-selector-strategy-class-name`	A class that implements org.jboss.jca.adapters.jdbc.URLSelectorStrategy
`new-connection-sql`	Specifies an SQL statement to execute whenever a connection is added to the connection pool

Element	Desciption
`transaction-isolation`	Set java.sql.Connection transaction isolation level to use. The constants defined by transaction-isolation-values are the possible transaction isolation levels and include: TRANSACTION_READ_UNCOMMITTED TRANSACTION_READ_COMMITTED TRANSACTION_REPEATABLE_READ TRANSACTION_SERIALIZABLE TRANSACTION_NONE
`xa-pool`	Specifies the pooling settings
`security`	Specifies the security settings
`validation`	Specifies the validation settings
`timeout`	Specifies the time out settings
`statement`	Specifies the statement settings
`recovery`	Specifies the recovery settings

Table 5.15. Pool settings

Element	Desciption
`min-pool-size`	The min-pool-size element indicates the minimum number of connections a pool should hold. These are not created until a Subject is known from a request for a connection. This default to 0
`max-pool-size`	The max-pool-size element indicates the maximum number of connections for a pool. No more connections will be created in each sub-pool. This defaults to 20
`prefill`	Whether to attempt to prefill the connection pool. Empty element denotes a true value. Default is false
`use-strict-min`	Define if the min-pool-size should be considered a strictly. Default false
`flush-strategy`	Specifies how the pool should be flush in case of an error. Valid values are: FailingConnectionOnly (default), IdleConnections, EntirePool
`allow-multiple-users`	Specifies if multiple users will access the datasource through the getConnection(user, password) method and hence if the internal pool type should account for that (1.1)

Table 5.16. XA pool settings

Element	Desciption
`min-pool-size`	The min-pool-size element indicates the minimum number of connections a pool should hold. These are not created until a Subject is known from a request for a connection. This default to 0
`max-pool-size`	The max-pool-size element indicates the maximum number of connections for a pool. No more connections will be created in each sub-pool. This defaults to 20
`prefill`	Whether to attempt to prefill the connection pool. Empty element denotes a true value. Default is false
`use-strict-min`	Define if the min-pool-size should be considered a strictly. Default false
`flush-strategy`	Specifies how the pool should be flush in case of an error. Valid values are: FailingConnectionOnly (default), IdleConnections, EntirePool
`allow-multiple-users`	Specifies if multiple users will access the datasource through the getConnection(user, password) method and hence if the internal pool type should account for that (1.1)
`is-same-rm-override`	The is-same-rm-override element allows one to unconditionally set whether the javax.transaction.xa.XAResource.isSameRM(XAResource) returns true or false
`interleaving`	An element to enable interleaving for XA connection factories
`no-tx-separate-pools`	Oracle does not like XA connections getting used both inside and outside a JTA transaction. To workaround the problem you can create separate sub-pools for the different contexts
`pad-xid`	Should the Xid be padded
`wrap-xa-resource`	Should the XAResource instances be wrapped in a org.jboss.tm.XAResourceWrapper instance

Table 5.17. Security settings

Element	Desciption
`user-name`	Specify the username used when creating a new connection.

Element	Desciption
password	Specify the password used when creating a new connection.
security-domain	Indicates Subject (from security domain) are used 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.
reauth-plugin	Defines a reauthentication plugin that can be used for reauthentication of physical connections.

Table 5.18. Validation settings

Element	Desciption
valid-connection-checker	An org.jboss.jca.adapters.jdbc.ValidConnectionChecker that provides a SQLException isValidConnection(Connection e) method to validate is a connection is valid. An exception means the connection is destroyed. This overrides the check-valid-connection-sql when present
check-valid-connection-sql	Specify an SQL statement to check validity of a pool connection. This may be called when managed connection is taken from pool for use.
validate-on-match	The validate-on-match element indicates whether or not connection level validation should be done when a connection factory attempts to match a managed connection for a given set. This is typically exclusive to the use of background validation
background-validation	An element to specify that connections should be validated on a background thread versus being validated prior to use
background-validation-minutes	The background-validation-minutes element specifies the amount of time, in minutes, that background validation will run
use-fast-fail	Whether fail a connection allocation on the first connection if it is invalid (true) or keep trying until the pool is exhausted of all potential connections (false) default false
stale-connection-checker	An org.jboss.jca.adapters.jdbc.StaleConnectionChecker that provides a boolean

Element	Desciption
	isStaleConnection(SQLException e) method which if it it returns true will wrap the exception in an org.jboss.jca.adapters.jdbc.StaleConnectionException which is a subclass of SQLException
`exception-sorter`	An org.jboss.jca.adapters.jdbc.ExceptionSorter that provides a boolean isExceptionFatal(SQLException e) method to validate is an exception should be broadcast to all javax.resource.spi.ConnectionEventListener as a connectionErrorOccurred message

Table 5.19. Time out settings

Element	Desciption
`blocking-timeout-millis`	The blocking-timeout-millis element indicates 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 30000 (30 seconds).
`idle-timeout-minutes`	The idle-timeout-minutes elements indicates 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.
`set-tx-query-timeout`	Whether to set the query timeout based on the time remaining until transaction timeout, any configured query timeout will be used if there is no transaction. The default is false
`query-timeout`	Any configured query timeout in seconds The default is no timeout
`use-try-lock`	Any configured timeout for internal locks on the resource adapter objects in seconds The default is a 60 second timeout
`allocation-retry`	The allocation retry element indicates the number of times that allocating a connection should be tried before throwing an exception. The default is 0.
`allocation-retry-wait-millis`	The allocation retry wait millis element indicates the time in milliseconds to wait between retrying to allocate a connection. The default is 5000 (5 seconds).

Element	Desciption
`xa-resource-timeout`	Passed to XAResource.setTransactionTimeout() Default is zero which does not invoke the setter. In seconds

Table 5.20. Statement settings

Element	Desciption
`track-statements`	Whether to check for unclosed statements when a connection is returned to the pool and result sets are closed when a statement is closed/return to the prepared statement cache. valid values are: `false` - do not track statements and results; `true` - track statements and result sets and warn when they are not closed; `nowarn` - track statements but do no warn about them being unclosed (the default)
`prepared-statement-cache-size`	The number of prepared statements per connection in an LRU cache
`share-prepared-statements`	Whether to share prepare statements, i.e. whether asking for same statement twice without closing uses the same underlying prepared statement. The default is false

Table 5.21. Recovery elements

Element	Desciption
`recover-credential`	Specifies the user name / password pair or security domain that should be used for recovery.
`recover-plugin`	Specifies an implementation of the org.jboss.jca.core.spi.recovery.RecoveryPlugin class.

Table 5.22. Driver attributes

Attribute	Desciption
`name`	An unique name for the JDBC driver
`module`	The module definition for the JDBC driver. The format of a module inside JBoss Application Server 7+ is `com.h2database.h2` which will map to the H2 installation under `modules/com/h2database/h2/main`. A ':' can be used to identify the slot - f.ex `com.h2database.h2:1.3.159`. The format for IronJacamar Standalone/Embedded is the name of the .jar file

Attribute	Desciption
`major-version`	The major version of the driver
`minor-version`	The minor version of the driver

Table 5.23. Driver elements

Element	Desciption
`driver-class`	The fully qualified class name of the driver class
`datasource-class`	The fully qualified class name of the datasource class
`xa-datasource-class`	The fully qualified class name of the XA datasource class

The datasource deployment schema is defined in `doc/datasources_1_0.xsd` and `doc/datasources_1_1.xsd`.

5.3.2. Datasource extensions

The datasource deployments can make use of a couple of extensions in the JDBC resource adapter to improve the connection validation and checking if an exception should reestablish the connection in question.

The extensions include

- `org.jboss.jca.adapters.jdbc.spi.ExceptionSorter`: Plugin to check if a `SQLException` is fatal for the connection on which it was thrown.
- `org.jboss.jca.adapters.jdbc.spi.StaleConnection`: Plugin to wrap stale `SQLException`'s in a `org.jboss.jca.adapters.jdbc.StaleConnectionException`.
- `org.jboss.jca.adapters.jdbc.spi.ValidConnection`: Plugin to Check if a connection is valid for use by the application.

Configuration of the extensions are done by using

- The `<exception-sorter>` tag for an `ExceptionSorter`
- The `<stale-connection-checker>` tag for a `StaleConnection`
- The `<valid-connection-checker>` tag for a `ValidConnection`

IronJacamar features implementations of these extensions for a couple of popular databases. Contributions in this area are most welcome either generic solutions or for a specific database.

Informix:

- `org.jboss.jca.adapters.jdbc.extensions.informix.InformixExceptionSorter`

Microsoft SQLServer:

- `org.jboss.jca.adapters.jdbc.extensions.mssql.MSSQLValidConnectionChecker`

PostgreSQL:

- `org.jboss.jca.adapters.jdbc.extensions.postgres.PostgreSQLExceptionSorter`
- `org.jboss.jca.adapters.jdbc.extensions.postgres.PostgreSQLValidConnectionChecker`

MySQL:

- `org.jboss.jca.adapters.jdbc.extensions.mysql.MySQLExceptionSorter`
- `org.jboss.jca.adapters.jdbc.extensions.mysql.MySQLReplicationValidConnectionChecker`
- `org.jboss.jca.adapters.jdbc.extensions.mysql.MySQLValidConnectionChecker`

IBM DB2:

- `org.jboss.jca.adapters.jdbc.extensions.db2.DB2ExceptionSorter`
- `org.jboss.jca.adapters.jdbc.extensions.db2.DB2StaleConnectionChecker`
- `org.jboss.jca.adapters.jdbc.extensions.db2.DB2ValidConnectionChecker`

Generic:

- `org.jboss.jca.adapters.jdbc.extensions.novendor.NullExceptionSorter`
- `org.jboss.jca.adapters.jdbc.extensions.novendor.NullStaleConnectionChecker`
- `org.jboss.jca.adapters.jdbc.extensions.novendor.NullValidConnectionChecker`
- `org.jboss.jca.adapters.jdbc.extensions.novendor.JDBC4ValidConnectionChecker`

Sybase:

- `org.jboss.jca.adapters.jdbc.extensions.sybase.SybaseExceptionSorter`
- `org.jboss.jca.adapters.jdbc.extensions.sybase.SybaseValidConnectionChecker`

Oracle:

- `org.jboss.jca.adapters.jdbc.extensions.oracle.OracleExceptionSorter`
- `org.jboss.jca.adapters.jdbc.extensions.oracle.OracleStaleConnectionChecker`
- `org.jboss.jca.adapters.jdbc.extensions.oracle.OracleValidConnectionChecker`

6

Running

6.1. Starting the container

The IronJacamar container is started by entering the bin/ directory

```
cd ironjacamar-1.0.0.Final/bin
```

and executing

```
./run.sh
```

on a Un*x based system or

```
run.bat
```

on Windows.

The command takes an optional -b argument to define the binding address of the naming server

```
./run.sh -b 192.168.0.199
```

Once the container has started you should see a log entry like

```
13:33:10,999 INFO  [Main] Server started in 941ms
```

```
```

in the console where the command was executed.

After the container has started you can browse to

```
http://localhost:8080
```

to view the project documentation and use the administration console.

6.2. Stopping the container

The IronJacamar container is stopped by pressing the `Ctrl-C` keys.

Once the container has stopped you should see a log entry like

```
13:35:06,752 INFO  [Main] Server stopped in 29ms
```

in the console where the container was running.

Alternative the container can be stopped through the command line interface.

6.3. Command line interface

The IronJacamar container can be controlled by a command line interface.

If you are accessing a remote container you can use the `-h` option to specify the host name.

6.3.1. Deploy

You can deploy a resource adapter archive (.rar) using

```
java -jar fungal-cli.jar deploy <file>
```

where `file` specifies the resource adapter archive.

6.3.2. Undeploy

You can undeploy a resource adapter archive (.rar) using

```
java -jar fungal-cli.jar undeploy <file>
```

where `file` specifies the resource adapter archive.

6.3.3. Shutdown

You can shutdown the IronJacamar environment by

```
java -jar fungal-cli.jar shutdown
```

7

Validator

7.1. Introduction

The IronJacamar container features a validator which checks resource adapter archives against the Java Connector Architecture (JCA) specification.

The validator is doing a static analysis of the resource adapter classes and checks them against the rules defined in the validator.

The validator is used in the deployer chain of the JCA container, and is available as a standalone tool, as an Apache Ant task and as a Apache Maven plugin too.

7.2. Reports

The validator works by scanning the resource adapter in question and output a report which lists which rules have been violated.

An example could be

```
Severity: ERROR
Section: 19.4.2
Description: A ResourceAdapter must implement a "public int hashCode()" method.
Code: com.mycompany.myproject.ResourceAdapterImpl

Severity: ERROR
Section: 19.4.2
Description: A ResourceAdapter must implement a "public boolean equals(Object)" method.
Code: com.mycompany.myproject.ResourceAdapterImpl
```

which means that `com.mycompany.myproject.ResourceAdapterImpl` is missing an `equals` and `hashCode` implementation.

Table 7.1. Validator report

Key	Desciption
`Severity`	Specifies the severity of the rule.

Key	Desciption
	• ERROR: Critical error which must be fixed in order for the resource adapter to operate correctly. • WARN: Error which should be fixed in order for the resource adapter to operate correctly.
`Section`	A reference to a section in the Java Connector Architecture specification where the requirement is defined.
`Descrption`	A short description of the rule.
`Code`	The class which triggered the rule.

7.3. Running the standalone validator

The validator can be run on the command line by

```
cd doc/validator
./validator.sh <file>
```

The reports will be generated into the current directory under the name of `<file>.log`.

7.4. Apache Ant integration

The validator integrates with Apache Ant such that you can generate the reports directly from your build environment before deploying the resoruce adapter into the IronJacamar container.

First you have to define the `taskdef` for the task

```
<taskdef name="validator"
         classname="org.jboss.jca.validator.ant.ValidatorTask"
         classpathref="ironjacamar.lib.path.id"/>
```

See the Apache Ant documentation for additional instructions on installation.

7.4.1. Usage

```
<validator rarFile="${myArchive.rar}" outputDir="${report.dir}"/>
```

Table 7.2. Apache Ant: validator

Key	Value
`rarFile`	The resource adapter file
`outputDir`	The directory where the reports should be generated
`classpath`	A classpath to resolve additional dependencies against

7.5. Apache Maven integration

The validator integrates with Apache Maven such that you can generate the reports directly from your build environment before deploying the resoruce adapter into the IronJacamar container.

To be able to use the validator plugin in your Maven project, you will have to add the following plugin declaration in the pom.xml of your project:

```
<build>
  <plugins>
    <plugin>
      <groupId>org.jboss.ironjacamar</groupId>
      <artifactId>ironjacamar-validator-maven</artifactId>
      <!-- The version of the plugin you want to use -->
      <version>1.0.0.Final</version>
      <executions>
        <execution>
          <goals>
            <goal>validate</goal>
          </goals>
        </execution>
      </executions>
      <configuration>
        <!-- output directory-->
        <outputDir>.</outputDir>

        <!-- rar filename -->
        <rarFile>/path/to/myresourceadapter.rar</rarFile>

        <!--  optional classpath
        <classpath>
          <param>classpath1</param>
          <param>classpath2</param>
        </classpath>
        -->
      </configuration>
    </plugin>
  </plugins>
</build>
```

Note

By default, the validator-maven plugin is attached to the "package" phase of Maven.

See the Apache Maven documentation for additional instructions on installation.

7.5.1. Usage

Once you have configured your project's pom.xml to include the validator-maven plugin, as explained earlier, you can generate the report by running the package goal on your project.

```
mvn clean package
```

Table 7.3. Apache Maven: validator

Key	Value
`rarFile`	The resource adapter file
`outputDir`	The directory where the reports should be generated
`classpath`	A classpath to resolve additional dependencies against

8

Code generator

8.1. Introduction

The IronJacamar project includes a resource adapter code generator which can generate a complete code skeleton that will help developers get started with their development tasks.

8.2. Functionality

The code generator will generate a resource adapter code skeleton based on the user input. The code generator supports

- Resource adapter using JCA 1.6 annotations
- Resource adapter using JCA 1.6 metadata
- Resource adapter using JCA 1.5
- Resource adapter using JCA 1.0
- Apache Ant build environment
- Apache Ant + Ivy build environment
- Apache Maven build environment
- Test suite environment

8.3. Running the tool

The code generator can be run on the command line by

```
./codegenerator.sh
```

from the `doc/codegenerator` directory.

The code generator supports the following arguments

Table 8.1. Code generator arguments

Argument	Desciption
`-o`	Specifies the output directory for the code skeleton.

The developer must then answer various questions regarding the properties of the resource adapter.

8.3.1. Developer Input

This section describes the questions that are asked in order to generate the code.

Table 8.2. Developer input

Question	Spec	Desciption	Type
Profile version (1.6/1.5/1.0)	All	Defines which Java EE Connector Architecture specification that the resource adapter should target	
Type (O/Outbound/I/ Inbound/B/Bidirectional)	JCA 1.5+	Defines if the resource adapter should contain outbound communication., inbound communication or both	
Transaction support (N/NoTransaction/L/ LocalTransaction/X/ XATransaction)	All	The transaction support level	
Package name	All	The package name of the resource adapter	
Use annotations (Y/Yes/ N/No)	JCA 1.6+	Should annotations be used for specifying the structure. If 'No' is selected a `META-INF/ra.xml` is generated	
Include a ResourceAdapter (Y/ Yes/N/No)	JCA 1.5+	Should an instance of a resource adapter class be included in the archive	Outbound
Resource adapter class name	JCA 1.5+	The class name of the resource adapter	Outbound or Bidirectional
Managed connection factory class name	All	The class name of the managed connection factory	Outbound or Bidirectional
Managed connection class name	All	The class name of the managed connection	Outbound or Bidirectional
Connection interface class name	All	The class name of the connection interface	Outbound or Bidirectional

Question	Spec	Desciption	Type
Connection implementation class name	All	The class name of the connection implementation	Outbound or Bidirectional
Connection factory interface class name	All	The class name of the connection factory interface	Outbound or Bidirectional
Connection factory implementation class name	All	The class name of the connection factory implementation	Outbound or Bidirectional
Resource adapter config properties	All	Include a configuration properties in the resource adapter instance	Outbound or Bidirectional
Managed connection factory config properties	All	Include a configuration properties in the managed connection factory instance	Outbound or Bidirectional
Use ResourceAdapterAssociation (Y/Yes/N/No)	All	Associate the managed connection factory instance with the resource adapter instance	Outbound or Bidirectional
Use CCI (Y/Yes/N/No)	All	Use the Common Client Interface for the connection / connection factory in the 'Outbound' part of the resource adapter	Outbound or Bidirectional
MessageListener interface name	JCA 1.5+	The name of the message listener interface for the activation	Inbound or Bidirectional
ActivationSpec class name	JCA 1.5+	The class name of the activation specification instance	Inbound or Bidirectional
ActivationSpec config properties	JCA 1.5+	Include configuration properties in the activation specification instance	Inbound or Bidirectional
Activation class name	JCA 1.5+	The class name of the activation instance	Inbound or Bidirectional
Add methods to connection interface (Y/Yes/N/No) [N]:	All	Use for add methods to connection interface	Outbound or Bidirectional
Build environment [A/Ant/I/Ant+Ivy/M/Maven]	All	Type of build environment	

8.4. Generated code

The generated code will consist of the classes making up the resource adapter and a test suite environment based on the embedded distribution.

8.4.1. Apache Ant build environment

The following targets are supported in the Apache Ant build environment

Table 8.3. Apache Ant build environment

Target	Desciption
`compile`	Compiles all the files
`rar`	Builds the resource adapter archive
`prepare-test`	Prepares the test environment
`test`	Executes the tests
`docs`	Generates the documentation

8.4.2. Apache Ant + Ivy build environment

The following targets are supported in the Apache Ant + Ivy build environment

Table 8.4. Apache Ant + Ivy build environment

Target	Desciption
`compile`	Compiles all the files
`rar`	Builds the resource adapter archive
`prepare-test`	Prepares the test environment
`test`	Executes the tests
`docs`	Generates the documentation

8.4.3. Apache Maven build environment

The following targets are supported in the Apache Maven build environment

Table 8.5. Apache Maven build environment

Target	Desciption
`compile`	Compiles all the files
`test`	Executes the tests

9

Embedded

9.1. Overview

The IronJacamar embedded configuration provides a way of running a JCA container in-VM.

The configuration is useful when you want a

- JCA container within your environment
- JCA container when doing unit testing

Especially the ability to unit test your resource adapter archives before deploying them into a testing or a production environment will benefit developers.

In order to enhance the experience with working with the embedded configuration the container integrates with the *ShrinkWrap* [http://www.jboss.org/community/wiki/ShrinkWrap] and *Arquillian* [http://community.jboss.org/en/arquillian] frameworks.

9.2. Deployment

You will need all the JAR files located in the

```
$IRON_JACAMAR_HOME/bin
$IRON_JACAMAR_HOME/lib
$IRON_JACAMAR_HOME/lib/embedded
```

directories on your application class loader - f.ex.

```
java -classpath allthejarfiles.jar yourapp
```

in order to use the embedded configuration.

If you want integration with the Arquillian framework you need to add the JAR files located in the

```
$IRON_JACAMAR_HOME/lib/embedded/arquillian
```

directory as well.

Furthermore you will need to configure Java Naming and Directory Interface (JNDI) and logging using for example property files.

`jndi.properties` file:

```
java.naming.factory.initial=org.jnp.interfaces.LocalOnlyContextFactory
java.naming.factory.url.pkgs=org.jboss.naming:org.jnp.interfaces
```

`logging.properties` file:

```
# Additional logger names to configure (root logger is always configured)
loggers=org.jboss.jca,org.jboss,org.jnp,com.arjuna

# Root logger level
logger.level=${iron.jacamar.log.level:INFO}
logger.handlers=CONSOLE, FILE

# org.jboss.jca
logger.org.jboss.jca.level=DEBUG

# org.jboss
logger.org.jboss.level=INFO

# org.jnp
logger.org.jnp.level=INFO

# com.arjuna
logger.com.arjuna.level=INFO

# Console handler configuration
handler.CONSOLE=org.jboss.logmanager.handlers.ConsoleHandler
handler.CONSOLE.properties=autoFlush
handler.CONSOLE.level=${iron.jacamar.log.console.level:INFO}
handler.CONSOLE.autoFlush=true
handler.CONSOLE.formatter=PATTERN

# File handler configuration
handler.FILE=org.jboss.logmanager.handlers.FileHandler
handler.FILE.level=${iron.jacamar.log.file.level:DEBUG}
handler.FILE.properties=autoFlush,fileName
handler.FILE.autoFlush=true
handler.FILE.fileName=${test.dir}/embedded/test.log
handler.FILE.formatter=PATTERN
```

```
# Formatter pattern configuration
formatter.PATTERN=org.jboss.logmanager.formatters.PatternFormatter
formatter.PATTERN.properties=pattern
formatter.PATTERN.pattern=%d{HH:mm:ss,SSS} %-5p [%c{1}] %m%n
```

These files needs to be available to the application classloader.

The code generator will generate a test suite based on the Arquillian functionality, so that environment can be used as a starting point for your own integration.

This setup will show you how to use dependencies from the JBoss Nexus Maven repository instead if you choose the Maven or Ant+Ivy based build environment.

Note

Note that, if you want to be able to deploy datasources you will need to deploy the `jdbc-local.rar` for `<datasource>` support, or `jdbc-xa.rar` for `<xa-datasource>` support. Both archives can be found in the `system/` directory.

9.3. Usage

IronJacamar Embedded supports both a simple and an advanced usage model, using pre-assembled resource adapter archives (.rar) or dynamic resource adapter archives based on ShrinkWrap.

The embedded environment supports registering resource adapters and datasources in the platform `MBeanServer` by setting the system property `ironjacamar.embedded.management` to `true` before starting the environment.

9.3.1. Simple usage

The code sample below shows a simple usage of deploying a pre-assembled resource adapter archive into the IronJacamar Embedded environment.

```
import org.jboss.jca.embedded.Embedded;
import org.jboss.jca.embedded.EmbeddedFactory;

import java.net.URL;

import javax.naming.Context;
import javax.naming.InitialContext;
import javax.naming.NamingException;

import org.junit.AfterClass;
import org.junit.BeforeClass;
import org.junit.Test;
import static org.junit.Assert.*;
```

```
public class MyTestCase
{
   /** Embedded */
   private static Embedded embedded;

   /** JNDI prefix */
   private static final String JNDI_PREFIX = "java:/eis/";

   /**
    * Simple test to verify deployment of myresourceadapter.rar
    * @throws Throwable throwable exception
    */
   @Test
   public void testDeployment() throws Throwable
   {
      URL archive = getURL("myresourceadapter.rar");

      Context context = null;

      try
      {
         embedded.deploy(archive);

         context = new InitialContext();
         Object o = context.lookup(JNDI_PREFIX + "myresourceadapter");
         assertNotNull(o);
      }
      catch (Throwable t)
      {
         fail(t.getMessage());
      }
      finally
      {
         embedded.undeploy(archive);

         if (context != null)
         {
            try
            {
               context.close();
            }
            catch (NamingException ne)
            {
               // Ignore
            }
         }
      }
   }

   @BeforeClass
   public static void beforeClass() throws Throwable
   {
      // Create an embedded JCA instance
      embedded = EmbeddedFactory.create();

      // Startup
      embedded.startup();
   }
```

```
   @AfterClass
   public static void afterClass() throws Throwable
   {
      // Shutdown
      embedded.shutdown();
   }
}
```

Note

Note that, the url for the archive must end with the `.rar` extension - either representing a file or a directory.

See the IronJacamar Embedded API documentation for additional functionality.

9.3.1.1. Automatic activation of archives

IronJacamar features a bean called `RAActivator` which will automatic create a JNDI binding for connection factories and administration objects. However, sometimes it is of benefit to define these bindings in a `-ra.xml` file, and therefore `RAActivator` has to be disabled during that deployment phase.

This done by using the following code snippet

```
import org.jboss.jca.deployers.fungal.RAActivator;

// Disable RAActivator
RAActivator raa = embedded.lookup("RAActivator", RAActivator.class);

if (raa == null)
   throw new IllegalStateException("RAActivator not defined");

raa.setEnabled(false);

embedded.deploy("myrar.rar");
embedded.deploy("myrar-ra.xml");

raa.setEnabled(true);
```

which disables the bean, does the deployments and then reenables the bean again.

9.3.2. Advanced usage

The IronJacamar Embedded container environment supports the following open source testing projects:

1. *ShrinkWrap* [http://www.jboss.org/shrinkwrap]

2. *Arquillian* [http://www.jboss.org/arquillian]

These extensions allow the developer to use the embedded platform with greater ease as there doesn't have to be a physical representation of the resource adapter archive located to the disk.

The Arquillian integration furthermore allows the developer to leave all the embedded container setup to the integration instead.

9.3.2.1. ShrinkWrap integration

The code sample below shows an advanced usage of deploying a dynamic ShrinkWrap resource adapter archive into the IronJacamar Embedded environment.

```
/*
 * JBoss, Home of Professional Open Source.
 * Copyright 2009, Red Hat Middleware LLC, and individual contributors
 * as indicated by the @author tags. See the copyright.txt file in the
 * distribution for a full listing of individual contributors.
 *
 * This is free software; you can redistribute it and/or modify it
 * under the terms of the GNU Lesser General Public License as
 * published by the Free Software Foundation; either version 2.1 of
 * the License, or (at your option) any later version.
 *
 * This software is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public
 * License along with this software; if not, write to the Free
 * Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA
 * 02110-1301 USA, or see the FSF site: http://www.fsf.org.
 */

package org.jboss.jca.embedded.unit;

import org.jboss.jca.embedded.Embedded;
import org.jboss.jca.embedded.EmbeddedFactory;
import org.jboss.jca.embedded.rars.simple.TestConnection;
import org.jboss.jca.embedded.rars.simple.TestConnectionFactory;

import java.util.UUID;

import javax.naming.Context;
import javax.naming.InitialContext;
import javax.naming.NamingException;

import org.jboss.logging.Logger;
import org.jboss.shrinkwrap.api.ShrinkWrap;
import org.jboss.shrinkwrap.api.spec.JavaArchive;
import org.jboss.shrinkwrap.api.spec.ResourceAdapterArchive;
```

```
import org.junit.AfterClass;
import org.junit.BeforeClass;
import org.junit.Test;
import static org.junit.Assert.*;

/**
 * Test cases for deploying resource adapter archives (.RAR)
 * using ShrinkWrap
 *
 * @author <a href="mailto:jesper.pedersen@jboss.org">Jesper Pedersen</a>
 * @version $Revision: $
 */
public class ShrinkWrapTestCase
{

   // ---------------------------------------------------------------------------||
   // Class Members -------------------------------------------------------------||
   // ---------------------------------------------------------------------------||

   private static Logger log = Logger.getLogger(ShrinkWrapTestCase.class);

   private static final String JNDI_PREFIX = "java:/eis/";

   /*
    * Embedded
    */
   private static Embedded embedded;

   // ---------------------------------------------------------------------------||
   // Tests ---------------------------------------------------------------------||
   // ---------------------------------------------------------------------------||

   /**
    * Null ShrinkWrap ResourceAdapterArchive test case
    * @exception Throwable Thrown if case of an error
    */
   @Test
   public void testNull() throws Throwable
   {
      ResourceAdapterArchive raa = null;

      try
      {
         embedded.deploy(raa);
         fail("Null deployment successful");
      }
      catch (Exception t)
      {
         // Ok
      }
      finally
      {
         try
         {
            embedded.undeploy(raa);
            fail("Null undeployment successful");
         }
         catch (Exception t)
         {
```

```
            // Ok
         }
      }
   }

   /**
    * Basic ShrinkWrap ResourceAdapterArchive test case
    * @exception Throwable Thrown if case of an error
    */
   @Test
   public void testBasic() throws Throwable
   {
      Context context = null;

      String name = UUID.randomUUID().toString();

      ResourceAdapterArchive raa =
         ShrinkWrap.create(ResourceAdapterArchive.class, name + ".rar");

      JavaArchive ja = ShrinkWrap.create(JavaArchive.class, UUID.randomUUID().toString() + ".jar");
      ja.addPackage(TestConnection.class.getPackage());

      raa.addAsLibrary(ja);
      raa.addAsManifestResource("simple.rar/META-INF/ra.xml", "ra.xml");

      try
      {
         embedded.deploy(raa);

         context = new InitialContext();
         TestConnectionFactory tcf = (TestConnectionFactory)context.lookup(JNDI_PREFIX + name);
         assertNotNull(tcf);

         TestConnection tc = tcf.getConnection();
         tc.callMe();
         tc.close();
      }
      catch (Throwable t)
      {
         log.error(t.getMessage(), t);
         fail(t.getMessage());
      }
      finally
      {
         if (context != null)
         {
            try
            {
               context.close();
            }
            catch (NamingException ne)
            {
               // Ignore
            }
         }

         embedded.undeploy(raa);
      }
   }
```

```
   // ------------------------------------------------------------------------||
   // Lifecycle Methods ------------------------------------------------------||
   // ------------------------------------------------------------------------||

   /**
    * Lifecycle start, before the suite is executed
    * @throws Throwable throwable exception
    */
   @BeforeClass
   public static void beforeClass() throws Throwable
   {
      // Create and set an embedded JCA instance
      embedded = EmbeddedFactory.create();

      // Startup
      embedded.startup();
   }

   /**
    * Lifecycle stop, after the suite is executed
    * @throws Throwable throwable exception
    */
   @AfterClass
   public static void afterClass() throws Throwable
   {
      // Shutdown embedded
      embedded.shutdown();

      // Set embedded to null
      embedded = null;
   }
}
```

Note

Note that, the name for the `ResourceAdapterArchive` must end with the `.rar` extension.

See the *ShrinkWrap* [http://www.jboss.org/shrinkwrap] web site for a full description of the project and additional documentation.

9.3.2.2. Arquillian integration

The code sample below shows an advanced usage of deploying a dynamic ShrinkWrap resource adapter archive into the IronJacamar Embedded environment using Arquillian.

This setup allows the developer to skip the entire IronJacamar Embedded container setup and handling of its lifecycle methods.

```
/*
 * JBoss, Home of Professional Open Source.
 * Copyright 2012, Red Hat Middleware LLC, and individual contributors
 * as indicated by the @author tags. See the copyright.txt file in the
 * distribution for a full listing of individual contributors.
 *
 * This is free software; you can redistribute it and/or modify it
 * under the terms of the GNU Lesser General Public License as
 * published by the Free Software Foundation; either version 2.1 of
 * the License, or (at your option) any later version.
 *
 * This software is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public
 * License along with this software; if not, write to the Free
 * Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA
 * 02110-1301 USA, or see the FSF site: http://www.fsf.org.
 */

package org.jboss.jca.embedded.unit;

import org.jboss.jca.embedded.rars.simple.TestConnection;
import org.jboss.jca.embedded.rars.simple.TestConnectionFactory;

import java.util.UUID;

import javax.annotation.Resource;

import org.jboss.arquillian.container.test.api.Deployment;
import org.jboss.arquillian.junit.Arquillian;
import org.jboss.logging.Logger;
import org.jboss.shrinkwrap.api.ShrinkWrap;
import org.jboss.shrinkwrap.api.spec.JavaArchive;
import org.jboss.shrinkwrap.api.spec.ResourceAdapterArchive;

import org.junit.Test;
import org.junit.runner.RunWith;

import static org.junit.Assert.assertNotNull;

/**
 * Unit test for Arquillian integration
 *
 * @author <a href="mailto:jesper.pedersen@jboss.org">Jesper Pedersen</a>
 */
@RunWith(Arquillian.class)
public class ArquillianTestCase
{
   // ---------------------------------------------------------------------------||
   // Class Members -------------------------------------------------------------||
   // ---------------------------------------------------------------------------||

   private static Logger log = Logger.getLogger(ArquillianTestCase.class);

   private static String deploymentName = "ArquillianTest";
```

```
   /**
    * Define the deployment
    * @return The deployment archive
    */
   @Deployment
   public static ResourceAdapterArchive createDeployment()
   {
      ResourceAdapterArchive raa =
         ShrinkWrap.create(ResourceAdapterArchive.class, deploymentName + ".rar");

      JavaArchive ja = ShrinkWrap.create(JavaArchive.class, UUID.randomUUID().toString() + ".jar");
      ja.addPackage(TestConnection.class.getPackage());

      raa.addAsLibrary(ja);
      raa.addAsManifestResource("simple.rar/META-INF/ra.xml", "ra.xml");

      return raa;
   }

   //-------------------------------------------------------------------------------------||
   // Tests ------------------------------------------------------------------------------||
   //-------------------------------------------------------------------------------------||

   @Resource(mappedName = "java:/eis/ArquillianTest")
   private TestConnectionFactory connectionFactory;

   /**
    * Basic
    * @exception Throwable Thrown if case of an error
    */
   @Test
   public void testBasic() throws Throwable
   {
      assertNotNull(connectionFactory);

      TestConnection c = connectionFactory.getConnection();
      assertNotNull(c);

      c.callMe();
      c.close();
   }
}
```

Note

Note that, the name for the `ResourceAdapterArchive` must end with the `.rar` extension.

See the *Arquillian* [http://www.jboss.org/arquillian] web site for a full description of the project and additional documentation.

10

Community

10.1. Website

The website contains the latest information about the project and links to important information.

The website is located at *http://www.jboss.org/ironjacamar/*

10.2. User forum

The user forum is where we discuss matters about the usage of the IronJacamar project.

Our forum is located at *http://community.jboss.org/en/ironjacamar*

10.3. Developer forum

The developer forum is where we discuss the implementation of the IronJacamar project. This means the internals of the project and not how the project is used.

User questions doesn't belong here - they should go in the user forum instead.

The forum is located at *http://community.jboss.org/en/ironjacamar/dev*

10.4. Issue tracking

We are using JIRA to manage our issues in the project.

These are divided into the following categories

- Feature Request: A feature that you would like see implemented.
- Bug: A software defect.

For all of these you should post your request to our user forum first.

The rest of the categories are for team use only.

Our issue tracking system located at *http://issues.jboss.org/browse/JBJCA*

11

Troubleshooting

11.1. I think I have found a bug

If you think you have found a bug you should verify this by posting to our forum first.

Our forum is located at *http://community.jboss.org/en/ironjacamar*

You can also search our issue tracking system located at *http://issues.jboss.org/browse/JBJCA*

11.2. I would like to implement a feature

So you have found an area where you are missing a feature and would like to submit a patch for it, great !

There are a couple of steps to get a feature included

First, you should create a new thread in our development forum where you describe the feature, its design and implementation.

Once there is an agreement on the feature and the design you should proceed with creating the patch.

To maximize your chances of getting the feature in the official build as soon as possible make sure that you run through the following steps:

```
ant clean test
ant clean checkstyle
ant clean findbugs
ant clean cobertura
```

All these should show that,

1. All your test cases for the feature is passing
2. Your code is correctly formatted according to project rules

3. There isn't any bug reports from the Findbugs environment

4. There is full code coverage based on the Cobertura report

when done, create a JIRA task (Feature Request) in our JIRA environment and attach the unified diff formatted patch. See the developer guide for additional details.

Happy Coding !

11.3. How do I ?

We can't cover every single issue in this guide, so feel free to drop by our forums to see if a solution has already been provided. Otherwise feel free to ask your question there.

Our forum is located at *http://community.jboss.org/en/ironjacamar*

Appendix A. Schemas

All the IronJacamar schemas are deployed under *http://www.jboss.org/ironjacamar/schema/*.

A.1. Java EE Connector Architecture 1.6

```
      <?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns="http://www.w3.org/2001/XMLSchema"
            targetNamespace="http://java.sun.com/xml/ns/javaee"
            xmlns:javaee="http://java.sun.com/xml/ns/javaee"
            xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            elementFormDefault="qualified"
            attributeFormDefault="unqualified"
            version="1.6">
  <xsd:annotation>
    <xsd:documentation>

      DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.

      Copyright 2003-2009 Sun Microsystems, Inc. All rights reserved.

      The contents of this file are subject to the terms of either the
      GNU General Public License Version 2 only ("GPL") or the Common
      Development and Distribution License("CDDL") (collectively, the
      "License").  You may not use this file except in compliance with
      the License. You can obtain a copy of the License at
      https://glassfish.dev.java.net/public/CDDL+GPL.html or
      glassfish/bootstrap/legal/LICENSE.txt.  See the License for the
      specific language governing permissions and limitations under the
      License.

      When distributing the software, include this License Header
      Notice in each file and include the License file at
      glassfish/bootstrap/legal/LICENSE.txt.  Sun designates this
      particular file as subject to the "Classpath" exception as
      provided by Sun in the GPL Version 2 section of the License file
      that accompanied this code.  If applicable, add the following
      below the License Header, with the fields enclosed by brackets []
      replaced by your own identifying information:
      "Portions Copyrighted [year] [name of copyright owner]"

      Contributor(s):

      If you wish your version of this file to be governed by only the
      CDDL or only the GPL Version 2, indicate your decision by adding
      "[Contributor] elects to include this software in this
      distribution under the [CDDL or GPL Version 2] license."  If you
      don't indicate a single choice of license, a recipient has the
      option to distribute your version of this file under either the
      CDDL, the GPL Version 2 or to extend the choice of license to its
      licensees as provided above.  However, if you add GPL Version 2
      code and therefore, elected the GPL Version 2 license, then the
      option applies only if the new code is made subject to such
      option by the copyright holder.
```

```
  </xsd:documentation>
</xsd:annotation>

<xsd:annotation>
  <xsd:documentation>

    DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.

    Copyright 2003-2009 Sun Microsystems, Inc. All rights reserved.

    The contents of this file are subject to the terms of either the
    GNU General Public License Version 2 only ("GPL") or the Common
    Development and Distribution License("CDDL") (collectively, the
    "License").  You may not use this file except in compliance with
    the License. You can obtain a copy of the License at
    https://glassfish.dev.java.net/public/CDDL+GPL.html or
    glassfish/bootstrap/legal/LICENSE.txt.  See the License for the
    specific language governing permissions and limitations under the
    License.

    When distributing the software, include this License Header
    Notice in each file and include the License file at
    glassfish/bootstrap/legal/LICENSE.txt.  Sun designates this
    particular file as subject to the "Classpath" exception as
    provided by Sun in the GPL Version 2 section of the License file
    that accompanied this code.  If applicable, add the following
    below the License Header, with the fields enclosed by brackets []
    replaced by your own identifying information:
    "Portions Copyrighted [year] [name of copyright owner]"

    Contributor(s):

    If you wish your version of this file to be governed by only the
    CDDL or only the GPL Version 2, indicate your decision by adding
    "[Contributor] elects to include this software in this
    distribution under the [CDDL or GPL Version 2] license."  If you
    don't indicate a single choice of license, a recipient has the
    option to distribute your version of this file under either the
    CDDL, the GPL Version 2 or to extend the choice of license to its
    licensees as provided above.  However, if you add GPL Version 2
    code and therefore, elected the GPL Version 2 license, then the
    option applies only if the new code is made subject to such
    option by the copyright holder.

  </xsd:documentation>
</xsd:annotation>

<xsd:annotation>
  <xsd:documentation>
    <![CDATA[
    This is the XML Schema for the Connector 1.6 deployment
    descriptor.  The deployment descriptor must be named
    "META-INF/ra.xml" in the connector's rar file.  All Connector
    deployment descriptors must indicate the connector resource
    adapter schema by using the Java EE namespace:

    http://java.sun.com/xml/ns/javaee
```

```
      and by indicating the version of the schema by
      using the version element as shown below:

      <connector xmlns="http://java.sun.com/xml/ns/javaee"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
         http://java.sun.com/xml/ns/javaee/connector_1_6.xsd"
      version="1.6">
      ...
      </connector>

      The instance documents may indicate the published version of
      the schema using the xsi:schemaLocation attribute for Java EE
      namespace with the following location:

      http://java.sun.com/xml/ns/javaee/connector_1_6.xsd

      ]]>
    </xsd:documentation>
  </xsd:annotation>

  <xsd:annotation>
    <xsd:documentation>

      The following conventions apply to all Java EE
      deployment descriptor elements unless indicated otherwise.

      - In elements that specify a pathname to a file within the
      same JAR file, relative filenames (i.e., those not
      starting with "/") are considered relative to the root of
      the JAR file's namespace.  Absolute filenames (i.e., those
      starting with "/") also specify names in the root of the
      JAR file's namespace.  In general, relative names are
      preferred.  The exception is .war files where absolute
      names are preferred for consistency with the Servlet API.

    </xsd:documentation>
  </xsd:annotation>

  <xsd:include schemaLocation="javaee_6.xsd"/>


<!-- **************************************************** -->

  <xsd:element name="connector"
               type="javaee:connectorType">
    <xsd:annotation>
      <xsd:documentation>

        The connector element is the root element of the deployment
        descriptor for the resource adapter. This element includes
        general information - vendor name, resource adapter version,
        icon - about the resource adapter module. It also includes
        information specific to the implementation of the resource
        adapter library as specified through the element
        resourceadapter.

      </xsd:documentation>
    </xsd:annotation>
```

```
  </xsd:element>


<!-- ********************************************** -->

  <xsd:complexType name="activationspecType">
    <xsd:annotation>
      <xsd:documentation>

        The activationspecType specifies an activation
        specification.  The information includes fully qualified
        Java class name of an activation specification and a set of
        required configuration property names.

      </xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element name="activationspec-class"
                   type="javaee:fully-qualified-classType">
        <xsd:annotation>
          <xsd:documentation>
            <![CDATA[
            The element activationspec-class specifies the fully
            qualified Java class name of the activation
            specification class. This class must implement the
            javax.resource.spi.ActivationSpec interface. The
            implementation of this class is required to be a
            JavaBean.

            Example:
                  <activationspec-class>com.wombat.ActivationSpecImpl
                  </activationspec-class>

            ]]>
          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="required-config-property"
                   type="javaee:required-config-propertyType"
                   minOccurs="0"
                   maxOccurs="unbounded">
        <xsd:annotation>
          <xsd:documentation>

            The required-config-property element is deprecated since
            Connectors 1.6 specification. The resource adapter
            implementation is recommended to use the @NotNull
            Bean Validation annotation or its XML validation
            descriptor equivalent to indicate that a configuration
            property is required to be specified by the deployer.
            See the Connectors specification for more information.

          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="config-property"
                   type="javaee:config-propertyType"
                   minOccurs="0"
                   maxOccurs="unbounded"/>
```

```
    </xsd:sequence>
    <xsd:attribute name="id"
                   type="xsd:ID"/>
  </xsd:complexType>


<!-- ********************************************** -->

  <xsd:complexType name="adminobjectType">
    <xsd:annotation>
      <xsd:documentation>

        The adminobjectType specifies information about an
        administered object.  Administered objects are specific to a
        messaging style or message provider.  This contains
        information on the Java type of the interface implemented by
        an administered object, its Java class name and its
        configuration properties.

      </xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element name="adminobject-interface"
                   type="javaee:fully-qualified-classType">
        <xsd:annotation>
          <xsd:documentation>
            <![CDATA[
            The element adminobject-interface specifies the
            fully qualified name of the Java type of the
            interface implemented by an administered object.

            Example:
                <adminobject-interface>javax.jms.Destination
                </adminobject-interface>

            ]]>
          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="adminobject-class"
                   type="javaee:fully-qualified-classType">
        <xsd:annotation>
          <xsd:documentation>
            <![CDATA[
            The element adminobject-class specifies the fully
            qualified Java class name of an administered object.

            Example:
                  <adminobject-class>com.wombat.DestinationImpl
                  </adminobject-class>

            ]]>
          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="config-property"
                   type="javaee:config-propertyType"
                   minOccurs="0"
                   maxOccurs="unbounded"/>
```

```
    </xsd:sequence>
    <xsd:attribute name="id"
                   type="xsd:ID"/>
  </xsd:complexType>


<!-- ********************************************** -->

  <xsd:complexType name="authentication-mechanismType">
    <xsd:annotation>
      <xsd:documentation>

        The authentication-mechanismType specifies an authentication
        mechanism supported by the resource adapter. Note that this
        support is for the resource adapter and not for the
        underlying EIS instance. The optional description specifies
        any resource adapter specific requirement for the support of
        security contract and authentication mechanism.

        Note that BasicPassword mechanism type should support the
        javax.resource.spi.security.PasswordCredential interface.
        The Kerbv5 mechanism type should support the
        org.ietf.jgss.GSSCredential interface or the deprecated
        javax.resource.spi.security.GenericCredential interface.

      </xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element name="description"
                   type="javaee:descriptionType"
                   minOccurs="0"
                   maxOccurs="unbounded"/>
      <xsd:element name="authentication-mechanism-type"
                   type="javaee:xsdStringType">
        <xsd:annotation>
          <xsd:documentation>
            <![CDATA[
            The element authentication-mechanism-type specifies
            type of an authentication mechanism.

            The example values are:

            <authentication-mechanism-type>BasicPassword
            </authentication-mechanism-type>

            <authentication-mechanism-type>Kerbv5
            </authentication-mechanism-type>

            Any additional security mechanisms are outside the
            scope of the Connector architecture specification.

            ]]>
          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="credential-interface"
                   type="javaee:credential-interfaceType"/>
    </xsd:sequence>
    <xsd:attribute name="id"
```

```
                    type="xsd:ID"/>
  </xsd:complexType>


<!-- ********************************************** -->

  <xsd:complexType name="config-property-nameType">
    <xsd:annotation>
      <xsd:documentation>
        <![CDATA[
        The config-property-nameType contains the name of a
        configuration property.

        The connector architecture defines a set of well-defined
        properties all of type java.lang.String. These are as
        follows.

        ServerName
        PortNumber
        UserName
        Password
        ConnectionURL

        A resource adapter provider can extend this property set to
        include properties specific to the resource adapter and its
        underlying EIS.

        Possible values include
              ServerName
              PortNumber
              UserName
              Password
              ConnectionURL

        Example: <config-property-name>ServerName</config-property-name>

        ]]>
      </xsd:documentation>
    </xsd:annotation>
    <xsd:simpleContent>
      <xsd:restriction base="javaee:xsdStringType"/>
    </xsd:simpleContent>
  </xsd:complexType>


<!-- ********************************************** -->

  <xsd:complexType name="config-property-typeType">
    <xsd:annotation>
      <xsd:documentation>
        <![CDATA[
        The config-property-typeType contains the fully
        qualified Java type of a configuration property.

        The following are the legal values:
        java.lang.Boolean, java.lang.String, java.lang.Integer,
        java.lang.Double, java.lang.Byte, java.lang.Short,
        java.lang.Long, java.lang.Float, java.lang.Character
```

```
        Used in: config-property

        Example:
        <config-property-type>java.lang.String</config-property-type>

        ]]>
      </xsd:documentation>
    </xsd:annotation>
    <xsd:simpleContent>
      <xsd:restriction base="javaee:string">
        <xsd:enumeration value="java.lang.Boolean"/>
        <xsd:enumeration value="java.lang.String"/>
        <xsd:enumeration value="java.lang.Integer"/>
        <xsd:enumeration value="java.lang.Double"/>
        <xsd:enumeration value="java.lang.Byte"/>
        <xsd:enumeration value="java.lang.Short"/>
        <xsd:enumeration value="java.lang.Long"/>
        <xsd:enumeration value="java.lang.Float"/>
        <xsd:enumeration value="java.lang.Character"/>
      </xsd:restriction>
    </xsd:simpleContent>
  </xsd:complexType>


<!-- ********************************************** -->

  <xsd:complexType name="config-propertyType">
    <xsd:annotation>
      <xsd:documentation>

        The config-propertyType contains a declaration of a single
        configuration property that may be used for providing
        configuration information.

        The declaration consists of an optional description, name,
        type and an optional value of the configuration property. If
        the resource adapter provider does not specify a value than
        the deployer is responsible for providing a valid value for
        a configuration property.

        Any bounds or well-defined values of properties should be
        described in the description element.

      </xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element name="description"
                   type="javaee:descriptionType"
                   minOccurs="0"
                   maxOccurs="unbounded"/>
      <xsd:element name="config-property-name"
                   type="javaee:config-property-nameType"/>
      <xsd:element name="config-property-type"
                   type="javaee:config-property-typeType"/>
      <xsd:element name="config-property-value"
                   type="javaee:xsdStringType"
                   minOccurs="0">
        <xsd:annotation>
          <xsd:documentation>
```

```
            <![CDATA[
            The element config-property-value contains the value
            of a configuration entry. Note, it is possible for a
            resource adapter deployer to override this
            configuration information during deployment.

            Example:
            <config-property-value>WombatServer</config-property-value>

            ]]>
        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="config-property-ignore"
                 type="javaee:true-falseType"
                 minOccurs="0"
                 maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>

          The element config-property-ignore is used to specify
          whether the configuration tools must ignore considering the
          configuration property during auto-discovery of
          Configuration properties. See the Connector specification for
          more details. If unspecified, the container must not ignore
          the configuration property during auto-discovery.
          This element must be one of the following, "true" or "false".

        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="config-property-supports-dynamic-updates"
                 type="javaee:true-falseType"
                 minOccurs="0"
                 maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>

          The element config-property-supports-dynamic-updates is used to specify
          whether the configuration property allows its value to be updated, by
          application server's configuration tools, during the lifetime of
          the JavaBean instance. See the Connector specification for
          more details. If unspecified, the container must not dynamically
          reconfigure the property.
          This element must be one of the following, "true" or "false".

        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="config-property-confidential"
                 type="javaee:true-falseType"
                 minOccurs="0"
                 maxOccurs="1">
      <xsd:annotation>
        <xsd:documentation>

          The element config-property-confidential is used to specify
          whether the configuration property is confidential and
          recommends application server's configuration tools to use special
```

```
            visual aids for editing them. See the Connector specification for
            more details. If unspecified, the container must not treat the
            property as confidential.
            This element must be one of the following, "true" or "false".

          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
    <xsd:attribute name="id"
                   type="xsd:ID"/>
  </xsd:complexType>


<!-- ********************************************** -->

  <xsd:complexType name="connection-definitionType">
    <xsd:annotation>
      <xsd:documentation>

        The connection-definitionType defines a set of connection
        interfaces and classes pertaining to a particular connection
        type. This also includes configurable properties for
        ManagedConnectionFactory instances that may be produced out
        of this set.

      </xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element name="managedconnectionfactory-class"
                   type="javaee:fully-qualified-classType">
        <xsd:annotation>
          <xsd:documentation>
            <![CDATA[
            The element managedconnectionfactory-class specifies
            the fully qualified name of the Java class that
            implements the
            javax.resource.spi.ManagedConnectionFactory interface.
            This Java class is provided as part of resource
            adapter's implementation of connector architecture
            specified contracts. The implementation of this
            class is required to be a JavaBean.

            Example:
            <managedconnectionfactory-class>
                  com.wombat.ManagedConnectionFactoryImpl
            </managedconnectionfactory-class>

            ]]>
          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="config-property"
                   type="javaee:config-propertyType"
                   minOccurs="0"
                   maxOccurs="unbounded"/>
      <xsd:element name="connectionfactory-interface"
                   type="javaee:fully-qualified-classType">
        <xsd:annotation>
```

```
          <xsd:documentation>
            <![CDATA[
            The element connectionfactory-interface specifies
            the fully qualified name of the ConnectionFactory
            interface supported by the resource adapter.

            Example:
            <connectionfactory-interface>com.wombat.ConnectionFactory
            </connectionfactory-interface>

            OR

            <connectionfactory-interface>javax.resource.cci.ConnectionFactory
            </connectionfactory-interface>

            ]]>
          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="connectionfactory-impl-class"
                   type="javaee:fully-qualified-classType">
        <xsd:annotation>
          <xsd:documentation>
            <![CDATA[
            The element connectionfactory-impl-class specifies
            the fully qualified name of the ConnectionFactory
            class that implements resource adapter
            specific ConnectionFactory interface.

            Example:

            <connectionfactory-impl-class>com.wombat.ConnectionFactoryImpl
            </connectionfactory-impl-class>

            ]]>
          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="connection-interface"
                   type="javaee:fully-qualified-classType">
        <xsd:annotation>
          <xsd:documentation>
            <![CDATA[
            The connection-interface element specifies the fully
            qualified name of the Connection interface supported
            by the resource adapter.

            Example:

                  <connection-interface>javax.resource.cci.Connection
                  </connection-interface>

            ]]>
          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="connection-impl-class"
                   type="javaee:fully-qualified-classType">
        <xsd:annotation>
```

```
          <xsd:documentation>
            <![CDATA[
            The connection-impl-classType specifies the fully
            qualified name of the Connection class that
            implements resource adapter specific Connection
            interface.  It is used by the connection-impl-class
            elements.

            Example:

                  <connection-impl-class>com.wombat.ConnectionImpl
                  </connection-impl-class>

            ]]>
          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
    <xsd:attribute name="id"
                   type="xsd:ID"/>
  </xsd:complexType>


<!-- ********************************************** -->

  <xsd:complexType name="connectorType">
    <xsd:annotation>
      <xsd:documentation>

        The connectorType defines a resource adapter.

      </xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element name="module-name"
                   type="javaee:string"
                   minOccurs="0">
        <xsd:annotation>
          <xsd:documentation>

            The element module-name specifies the name of the
            resource adapter.

            If there is no module-name specified, the module-name
            is determined as defined in Section EE.8.1.1 and EE.8.1.2
            of the Java Platform, Enterprise Edition (Java EE)
            Specification, version 6.

          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:group ref="javaee:descriptionGroup"/>
      <xsd:element name="vendor-name"
                   type="javaee:xsdStringType"
                   minOccurs="0">
        <xsd:annotation>
          <xsd:documentation>

            The element vendor-name specifies the name of
```

```
            resource adapter provider vendor.

            If there is no vendor-name specified, the application
            server must consider the default "" (empty string) as
            the name of the resource adapter provider vendor.

        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="eis-type"
                 type="javaee:xsdStringType"
                 minOccurs="0">
      <xsd:annotation>
        <xsd:documentation>

            The element eis-type contains information about the
            type of the EIS. For example, the type of an EIS can
            be product name of EIS independent of any version
            info.

            This helps in identifying EIS instances that can be
            used with this resource adapter.

            If there is no eis-type specified, the application
            server must consider the default "" (empty string) as
            the type of the EIS.

        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="resourceadapter-version"
                 type="javaee:xsdStringType"
                 minOccurs="0">
      <xsd:annotation>
        <xsd:documentation>

            The element resourceadapter-version specifies a string-based version
            of the resource adapter from the resource adapter
            provider.

            If there is no resourceadapter-version specified, the application
            server must consider the default "" (empty string) as
            the version of the resource adapter.


        </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="license"
                 type="javaee:licenseType"
                 minOccurs="0"/>
    <xsd:element name="resourceadapter"
                 type="javaee:resourceadapterType"/>
    <xsd:element name="required-work-context"
                 type="javaee:fully-qualified-classType"
                 minOccurs="0"
                 maxOccurs="unbounded">
      <xsd:annotation>
        <xsd:documentation>
```

```
            The element required-work-context specifies a fully qualified class
            name that implements WorkContext interface, that the resource adapter
            requires the application server to support.

          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
    <xsd:attribute name="version"
                   type="javaee:dewey-versionType"
                   fixed="1.6"
                   use="required">
      <xsd:annotation>
        <xsd:documentation>

          The version indicates the version of the schema to be used by the
          deployment tool. This element doesn't have a default, and the resource adapter
          developer/deployer is required to specify it. The element allows the deployment
          tool to choose which schema to validate the descriptor against.


        </xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="metadata-complete"
                   type="xsd:boolean">
      <xsd:annotation>
        <xsd:documentation>

          The metadata-complete attribute defines whether the deployment
          descriptor for the resource adapter module is complete, or whether
          the class files available to the module and packaged with the resource
          adapter should be examined for annotations that specify deployment
          information.

          If metadata-complete is set to "true", the deployment tool of the
          application server must ignore any annotations that specify deployment
          information, which might be present in the class files of the
          application.If metadata-complete is not specified or is set to "false",
          the deployment tool must examine the class files of the application for
          annotations, as specified by this specification. If the
          deployment descriptor is not included or is included but not marked
          metadata-complete, the deployment tool will process annotations.

          Application servers must assume that metadata-complete is true for
          resource adapter modules with deployment descriptor version
          lower than 1.6.

        </xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="id"
                   type="xsd:ID"/>
  </xsd:complexType>


<!-- ********************************************** -->
```

```
  <xsd:complexType name="credential-interfaceType">
    <xsd:annotation>
      <xsd:documentation>

        The credential-interfaceType specifies the
        interface that the resource adapter implementation
        supports for the representation of the
        credentials. This element(s) that use this type,
        i.e. credential-interface,  should be used by
        application server to find out the Credential
        interface it should use as part of the security
        contract.

        The possible values are:

        javax.resource.spi.security.PasswordCredential
        org.ietf.jgss.GSSCredential
        javax.resource.spi.security.GenericCredential

      </xsd:documentation>
    </xsd:annotation>
    <xsd:simpleContent>
      <xsd:restriction base="javaee:fully-qualified-classType">
        <xsd:enumeration value="javax.resource.spi.security.PasswordCredential"/>
        <xsd:enumeration value="org.ietf.jgss.GSSCredential"/>
        <xsd:enumeration value="javax.resource.spi.security.GenericCredential"/>
      </xsd:restriction>
    </xsd:simpleContent>
  </xsd:complexType>


<!-- ************************************************ -->

  <xsd:complexType name="inbound-resourceadapterType">
    <xsd:annotation>
      <xsd:documentation>

        The inbound-resourceadapterType specifies information
        about an inbound resource adapter. This contains information
        specific to the implementation of the resource adapter
        library as specified through the messageadapter element.

      </xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element name="messageadapter"
                   type="javaee:messageadapterType"
                   minOccurs="0">
        <xsd:unique name="messagelistener-type-uniqueness">
          <xsd:annotation>
            <xsd:documentation>

              The messagelistener-type element content must be
              unique in the messageadapter. Several messagelisteners
              can not use the same messagelistener-type.

            </xsd:documentation>
          </xsd:annotation>
          <xsd:selector xpath="javaee:messagelistener"/>
```

```
          <xsd:field xpath="javaee:messagelistener-type"/>
        </xsd:unique>
      </xsd:element>
    </xsd:sequence>
    <xsd:attribute name="id"
                   type="xsd:ID"/>
  </xsd:complexType>


<!-- ********************************************** -->

  <xsd:complexType name="licenseType">
    <xsd:annotation>
      <xsd:documentation>

        The licenseType specifies licensing requirements for the
        resource adapter module. This type specifies whether a
        license is required to deploy and use this resource adapter,
        and an optional description of the licensing terms
        (examples: duration of license, number of connection
        restrictions). It is used by the license element.

      </xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element name="description"
                   type="javaee:descriptionType"
                   minOccurs="0"
                   maxOccurs="unbounded"/>
      <xsd:element name="license-required"
                   type="javaee:true-falseType">
        <xsd:annotation>
          <xsd:documentation>

            The element license-required specifies whether a
            license is required to deploy and use the
            resource adapter. This element must be one of
            the following, "true" or "false".

          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
    <xsd:attribute name="id"
                   type="xsd:ID"/>
  </xsd:complexType>


<!-- ********************************************** -->

  <xsd:complexType name="messageadapterType">
    <xsd:annotation>
      <xsd:documentation>

        The messageadapterType specifies information about the
        messaging capabilities of the resource adapter. This
        contains information specific to the implementation of the
        resource adapter library as specified through the
        messagelistener element.
```

```
      </xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element name="messagelistener"
                   type="javaee:messagelistenerType"
                   maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="id"
                   type="xsd:ID"/>
  </xsd:complexType>


<!-- **************************************************** -->

  <xsd:complexType name="messagelistenerType">
    <xsd:annotation>
      <xsd:documentation>

        The messagelistenerType specifies information about a
        specific message listener supported by the messaging
        resource adapter. It contains information on the Java type
        of the message listener interface and an activation
        specification.

      </xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element name="messagelistener-type"
                   type="javaee:fully-qualified-classType">
        <xsd:annotation>
          <xsd:documentation>
            <![CDATA[
            The element messagelistener-type specifies the fully
            qualified name of the Java type of a message
            listener interface.

            Example:

                <messagelistener-type>javax.jms.MessageListener
                </messagelistener-type>

            ]]>
          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="activationspec"
                   type="javaee:activationspecType"/>
    </xsd:sequence>
    <xsd:attribute name="id"
                   type="xsd:ID"/>
  </xsd:complexType>


<!-- **************************************************** -->

  <xsd:complexType name="outbound-resourceadapterType">
    <xsd:annotation>
      <xsd:documentation>
```

```
      The outbound-resourceadapterType specifies information about
      an outbound resource adapter. The information includes fully
      qualified names of classes/interfaces required as part of
      the connector architecture specified contracts for
      connection management, level of transaction support
      provided, one or more authentication mechanisms supported
      and additional required security permissions.

      If any of the outbound resource adapter elements (transaction-support,
      authentication-mechanism, reauthentication-support) is specified through
      this element or metadata annotations, and no  connection-definition is
      specified as part of this element or through annotations, the
      application server must consider this an error and fail deployment.

      If there is no authentication-mechanism specified as part of
      this element or metadata annotations, then the resource adapter does
      not support any standard security authentication mechanisms as
      part of security contract. The application server ignores the security
      part of the system contracts in this case.

      If there is no transaction-support specified as part of this element
      or metadata annotation, then the application server must consider that
      the resource adapter does not support either the resource manager local
      or JTA transactions and must consider the transaction support as
      NoTransaction. Note that resource adapters may specify the level of
      transaction support to be used at runtime for a ManagedConnectionFactory
      through the TransactionSupport interface.

      If there is no reauthentication-support specified as part of
      this element or metadata annotation, then the application server must consider
      that the resource adapter does not support re-authentication of
      ManagedConnections.

    </xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="connection-definition"
                 type="javaee:connection-definitionType"
                 maxOccurs="unbounded"
                 minOccurs="0"/>
    <xsd:element name="transaction-support"
                 type="javaee:transaction-supportType"
                 minOccurs="0"/>
    <xsd:element name="authentication-mechanism"
                 type="javaee:authentication-mechanismType"
                 minOccurs="0"
                 maxOccurs="unbounded"/>
    <xsd:element name="reauthentication-support"
                 type="javaee:true-falseType"
                 minOccurs="0">
      <xsd:annotation>
        <xsd:documentation>

                  The element reauthentication-support specifies
                  whether the resource adapter implementation supports
                  re-authentication of existing Managed- Connection
                  instance. Note that this information is for the
                  resource adapter implementation and not for the
```

```
                    underlying EIS instance. This element must have
                    either a "true" or "false" value.


          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
    <xsd:attribute name="id"
                   type="xsd:ID"/>
  </xsd:complexType>


<!-- **************************************************** -->

  <xsd:complexType name="required-config-propertyType">
    <xsd:annotation>
      <xsd:documentation>
        <![CDATA[
        The required-config-propertyType contains a declaration
        of a single configuration property used for specifying a
        required configuration property name. It is used
        by required-config-property elements.

        Usage of this type is deprecated from Connectors 1.6 specification.
        Refer to required-config-property element for more information.

        Example:

        <required-config-property>
        <config-property-name>Destination</config-property-name>
        </required-config-property>

        ]]>
      </xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element name="description"
                   type="javaee:descriptionType"
                   minOccurs="0"
                   maxOccurs="unbounded"/>
      <xsd:element name="config-property-name"
                   type="javaee:config-property-nameType"/>
    </xsd:sequence>
    <xsd:attribute name="id"
                   type="xsd:ID"/>
  </xsd:complexType>


<!-- **************************************************** -->

  <xsd:complexType name="resourceadapterType">
    <xsd:annotation>
      <xsd:documentation>

        The resourceadapterType specifies information about the
        resource adapter. The information includes fully qualified
        resource adapter Java class name, configuration properties,
        information specific to the implementation of the resource
```

```
        adapter library as specified through the
        outbound-resourceadapter and inbound-resourceadapter
        elements, and an optional set of administered objects.

      </xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element name="resourceadapter-class"
                   type="javaee:fully-qualified-classType"
                   minOccurs="0">
        <xsd:annotation>
          <xsd:documentation>

            The element resourceadapter-class specifies the
            fully qualified name of a Java class that implements
            the javax.resource.spi.ResourceAdapter
            interface. This Java class is provided as part of
            resource adapter's implementation of connector
            architecture specified contracts. The implementation
            of this class is required to be a JavaBean.

          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="config-property"
                   type="javaee:config-propertyType"
                   minOccurs="0"
                   maxOccurs="unbounded"/>
      <xsd:element name="outbound-resourceadapter"
                   type="javaee:outbound-resourceadapterType"
                   minOccurs="0">
        <xsd:unique name="connectionfactory-interface-uniqueness">
          <xsd:annotation>
            <xsd:documentation>

              The connectionfactory-interface element content
              must be unique in the outbound-resourceadapter.
              Multiple connection-definitions can not use the
              same connectionfactory-type.

            </xsd:documentation>
          </xsd:annotation>
          <xsd:selector xpath="javaee:connection-definition"/>
          <xsd:field xpath="javaee:connectionfactory-interface"/>
        </xsd:unique>
      </xsd:element>
      <xsd:element name="inbound-resourceadapter"
                   type="javaee:inbound-resourceadapterType"
                   minOccurs="0"/>
      <xsd:element name="adminobject"
                   type="javaee:adminobjectType"
                   minOccurs="0"
                   maxOccurs="unbounded">
        <xsd:unique name="adminobject-type-uniqueness">
          <xsd:annotation>
            <xsd:documentation>

              The adminobject-interface and adminobject-class element content must be
              unique in the resourceadapterType. Several admin objects
```

```
              can not use the same adminobject-interface and adminobject-class.

            </xsd:documentation>
          </xsd:annotation>
          <xsd:selector xpath="javaee:adminobject"/>
          <xsd:field xpath="javaee:adminobject-interface"/>
          <xsd:field xpath="javaee:adminobject-class"/>
        </xsd:unique>
      </xsd:element>
      <xsd:element name="security-permission"
                   type="javaee:security-permissionType"
                   minOccurs="0"
                   maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="id"
                   type="xsd:ID"/>
  </xsd:complexType>


<!-- ********************************************** -->

  <xsd:complexType name="security-permissionType">
    <xsd:annotation>
      <xsd:documentation>

        The security-permissionType specifies a security
        permission that is required by the resource adapter code.

        The security permission listed in the deployment descriptor
        are ones that are different from those required by the
        default permission set as specified in the connector
        specification. The optional description can mention specific
        reason that resource adapter requires a given security
        permission.

      </xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element name="description"
                   type="javaee:descriptionType"
                   minOccurs="0"
                   maxOccurs="unbounded"/>
      <xsd:element name="security-permission-spec"
                   type="javaee:xsdStringType">
        <xsd:annotation>
          <xsd:documentation>

            The element security-permission-spec specifies a security
            permission based on the Security policy file
            syntax. Refer to the following URL for Sun's
            implementation of the security permission
            specification:

            http://java.sun.com/javase/6/docs/technotes/guides/security/PolicyFiles.html

          </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
```

```
    <xsd:attribute name="id"
                   type="xsd:ID"/>
  </xsd:complexType>


<!-- **************************************************** -->

  <xsd:complexType name="transaction-supportType">
    <xsd:annotation>
      <xsd:documentation>

        The transaction-supportType specifies the level of
        transaction support provided by the resource adapter. It is
        used by transaction-support elements.

        The value must be one of the following:

        NoTransaction
        LocalTransaction
        XATransaction

      </xsd:documentation>
    </xsd:annotation>
    <xsd:simpleContent>
      <xsd:restriction base="javaee:string">
        <xsd:enumeration value="NoTransaction"/>
        <xsd:enumeration value="LocalTransaction"/>
        <xsd:enumeration value="XATransaction"/>
      </xsd:restriction>
    </xsd:simpleContent>
  </xsd:complexType>

</xsd:schema>
```

A.2. Java EE Connector Architecture 1.5

```
      <?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns="http://www.w3.org/2001/XMLSchema"
        targetNamespace="http://java.sun.com/xml/ns/j2ee"
        xmlns:j2ee="http://java.sun.com/xml/ns/j2ee"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema"
        elementFormDefault="qualified"
        attributeFormDefault="unqualified"
        version="1.5">
  <xsd:annotation>
    <xsd:documentation>
      @(#)connector_1_5.xsds    1.27 06/17/03
    </xsd:documentation>
  </xsd:annotation>

  <xsd:annotation>
    <xsd:documentation>
```

```
      DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.

      Copyright 2003-2007 Sun Microsystems, Inc. All rights reserved.

      The contents of this file are subject to the terms of either the
      GNU General Public License Version 2 only ("GPL") or the Common
      Development and Distribution License("CDDL") (collectively, the
      "License").  You may not use this file except in compliance with
      the License. You can obtain a copy of the License at
      https://glassfish.dev.java.net/public/CDDL+GPL.html or
      glassfish/bootstrap/legal/LICENSE.txt.  See the License for the
      specific language governing permissions and limitations under the
      License.

      When distributing the software, include this License Header
      Notice in each file and include the License file at
      glassfish/bootstrap/legal/LICENSE.txt.  Sun designates this
      particular file as subject to the "Classpath" exception as
      provided by Sun in the GPL Version 2 section of the License file
      that accompanied this code.  If applicable, add the following
      below the License Header, with the fields enclosed by brackets []
      replaced by your own identifying information:
      "Portions Copyrighted [year] [name of copyright owner]"

      Contributor(s):

      If you wish your version of this file to be governed by only the
      CDDL or only the GPL Version 2, indicate your decision by adding
      "[Contributor] elects to include this software in this
      distribution under the [CDDL or GPL Version 2] license."  If you
      don't indicate a single choice of license, a recipient has the
      option to distribute your version of this file under either the
      CDDL, the GPL Version 2 or to extend the choice of license to its
      licensees as provided above.  However, if you add GPL Version 2
      code and therefore, elected the GPL Version 2 license, then the
      option applies only if the new code is made subject to such
      option by the copyright holder.

    </xsd:documentation>
  </xsd:annotation>

  <xsd:annotation>
    <xsd:documentation>
      <![CDATA[

    This is the XML Schema for the Connector 1.5 deployment
    descriptor.  The deployment descriptor must be named
    "META-INF/ra.xml" in the connector's rar file.  All Connector
    deployment descriptors must indicate the connector resource
    adapter schema by using the J2EE namespace:

    http://java.sun.com/xml/ns/j2ee

    and by indicating the version of the schema by
    using the version element as shown below:

        <connector xmlns="http://java.sun.com/xml/ns/j2ee"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
```

```
          http://java.sun.com/xml/ns/j2ee/connector_1_5.xsd"
           version="1.5">
           ...
         </connector>

    The instance documents may indicate the published version of
    the schema using the xsi:schemaLocation attribute for J2EE
    namespace with the following location:

    http://java.sun.com/xml/ns/j2ee/connector_1_5.xsd

    ]]>
    </xsd:documentation>
  </xsd:annotation>

  <xsd:annotation>
    <xsd:documentation>

      The following conventions apply to all J2EE
      deployment descriptor elements unless indicated otherwise.

      - In elements that specify a pathname to a file within the
    same JAR file, relative filenames (i.e., those not
    starting with "/") are considered relative to the root of
    the JAR file's namespace.  Absolute filenames (i.e., those
    starting with "/") also specify names in the root of the
    JAR file's namespace.  In general, relative names are
    preferred.  The exception is .war files where absolute
    names are preferred for consistency with the Servlet API.

    </xsd:documentation>
  </xsd:annotation>

  <xsd:include schemaLocation="j2ee_1_4.xsd"/>


<!-- **************************************************** -->


  <xsd:element name="connector" type="j2ee:connectorType">
    <xsd:annotation>
      <xsd:documentation>

    The connector element is the root element of the deployment
    descriptor for the resource adapter. This element includes
    general information - vendor name, resource adapter version,
    icon - about the resource adapter module. It also includes
    information specific to the implementation of the resource
    adapter library as specified through the element
    resourceadapter.

      </xsd:documentation>
    </xsd:annotation>

  </xsd:element>

<!-- **************************************************** -->

  <xsd:complexType name="activationspecType">
```

```
    <xsd:annotation>
      <xsd:documentation>

    The activationspecType specifies an activation
    specification.  The information includes fully qualified
    Java class name of an activation specification and a set of
    required configuration property names.

      </xsd:documentation>
    </xsd:annotation>

    <xsd:sequence>
      <xsd:element name="activationspec-class"
           type="j2ee:fully-qualified-classType">
    <xsd:annotation>
      <xsd:documentation>
        <![CDATA[

          The element activationspec-class specifies the fully
          qualified Java class name of the activation
          specification class. This class must implement the
          javax.resource.spi.ActivationSpec interface. The
          implementation of this class is required to be a
          JavaBean.

          Example:
          <activationspec-class>com.wombat.ActivationSpecImpl
          </activationspec-class>

          ]]>
      </xsd:documentation>
    </xsd:annotation>
      </xsd:element>
      <xsd:element name="required-config-property"
           type="j2ee:required-config-propertyType"
           minOccurs="0"
           maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="id" type="xsd:ID"/>
  </xsd:complexType>

<!-- **************************************************** -->

  <xsd:complexType name="adminobjectType">
    <xsd:annotation>
      <xsd:documentation>

    The adminobjectType specifies information about an
    administered object.  Administered objects are specific to a
    messaging style or message provider.  This contains
    information on the Java type of the interface implemented by
    an administered object, its Java class name and its
    configuration properties.

      </xsd:documentation>
    </xsd:annotation>

    <xsd:sequence>
      <xsd:element name="adminobject-interface"
```

```
            type="j2ee:fully-qualified-classType">
    <xsd:annotation>
      <xsd:documentation>
        <![CDATA[

          The element adminobject-interface specifies the
          fully qualified name of the Java type of the
          interface implemented by an administered object.

          Example:
        <adminobject-interface>javax.jms.Destination
        </adminobject-interface>

          ]]>
      </xsd:documentation>
    </xsd:annotation>

      </xsd:element>
      <xsd:element name="adminobject-class"
            type="j2ee:fully-qualified-classType">
    <xsd:annotation>
      <xsd:documentation>
        <![CDATA[

          The element adminobject-class specifies the fully
          qualified Java class name of an administered object.

          Example:
          <adminobject-class>com.wombat.DestinationImpl
          </adminobject-class>

          ]]>
      </xsd:documentation>
    </xsd:annotation>
      </xsd:element>

      <xsd:element name="config-property"
           type="j2ee:config-propertyType"
           minOccurs="0"
           maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="id" type="xsd:ID"/>
  </xsd:complexType>

<!-- ********************************************** -->

  <xsd:complexType name="authentication-mechanismType">
    <xsd:annotation>
      <xsd:documentation>

    The authentication-mechanismType specifies an authentication
    mechanism supported by the resource adapter. Note that this
    support is for the resource adapter and not for the
    underlying EIS instance. The optional description specifies
    any resource adapter specific requirement for the support of
    security contract and authentication mechanism.

    Note that BasicPassword mechanism type should support the
    javax.resource.spi.security.PasswordCredential interface.
```

```
    The Kerbv5 mechanism type should support the
    org.ietf.jgss.GSSCredential interface or the deprecated
    javax.resource.spi.security.GenericCredential interface.

      </xsd:documentation>
    </xsd:annotation>

    <xsd:sequence>
      <xsd:element name="description"
           type="j2ee:descriptionType"
           minOccurs="0"
           maxOccurs="unbounded"/>
      <xsd:element name="authentication-mechanism-type"
           type="j2ee:xsdStringType">
    <xsd:annotation>
      <xsd:documentation>
        <![CDATA[

          The element authentication-mechanism-type specifies
          type of an authentication mechanism.

          The example values are:

          <authentication-mechanism-type>BasicPassword
          </authentication-mechanism-type>

          <authentication-mechanism-type>Kerbv5
          </authentication-mechanism-type>

          Any additional security mechanisms are outside the
          scope of the Connector architecture specification.

          ]]>
      </xsd:documentation>
    </xsd:annotation>

      </xsd:element>
      <xsd:element name="credential-interface"
           type="j2ee:credential-interfaceType"/>
    </xsd:sequence>
    <xsd:attribute name="id" type="xsd:ID"/>
  </xsd:complexType>

<!-- ********************************************** -->

  <xsd:complexType name="config-property-nameType">
    <xsd:annotation>
      <xsd:documentation>
    <![CDATA[

      The config-property-nameType contains the name of a
      configuration property.

      The connector architecture defines a set of well-defined
      properties all of type java.lang.String. These are as
      follows.

           ServerName
           PortNumber
```

```
            UserName
            Password
            ConnectionURL

      A resource adapter provider can extend this property set to
      include properties specific to the resource adapter and its
      underlying EIS.

      Possible values include
          ServerName
          PortNumber
          UserName
          Password
          ConnectionURL

      Example: <config-property-name>ServerName</config-property-name>

      ]]>
      </xsd:documentation>
    </xsd:annotation>
    <xsd:simpleContent>
      <xsd:restriction base="j2ee:xsdStringType"/>
    </xsd:simpleContent>
  </xsd:complexType>

<!-- ********************************************** -->

  <xsd:complexType name="config-property-typeType">
    <xsd:annotation>
      <xsd:documentation>
    <![CDATA[

      The config-property-typeType contains the fully
      qualified Java type of a configuration property.

      The following are the legal values:
         java.lang.Boolean, java.lang.String, java.lang.Integer,
         java.lang.Double, java.lang.Byte, java.lang.Short,
         java.lang.Long, java.lang.Float, java.lang.Character

      Used in: config-property

      Example:
      <config-property-type>java.lang.String</config-property-type>

      ]]>
      </xsd:documentation>
    </xsd:annotation>
    <xsd:simpleContent>
      <xsd:restriction base="j2ee:string">
    <xsd:enumeration value="java.lang.Boolean"/>
    <xsd:enumeration value="java.lang.String"/>
    <xsd:enumeration value="java.lang.Integer"/>
    <xsd:enumeration value="java.lang.Double"/>
    <xsd:enumeration value="java.lang.Byte"/>
    <xsd:enumeration value="java.lang.Short"/>
    <xsd:enumeration value="java.lang.Long"/>
    <xsd:enumeration value="java.lang.Float"/>
    <xsd:enumeration value="java.lang.Character"/>
```

```
      </xsd:restriction>
    </xsd:simpleContent>
  </xsd:complexType>

<!-- **************************************************** -->

  <xsd:complexType name="config-propertyType">
    <xsd:annotation>
      <xsd:documentation>

    The config-propertyType contains a declaration of a single
    configuration property that may be used for providing
    configuration information.

    The declaration consists of an optional description, name,
    type and an optional value of the configuration property. If
    the resource adapter provider does not specify a value than
    the deployer is responsible for providing a valid value for
    a configuration property.

    Any bounds or well-defined values of properties should be
    described in the description element.

      </xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element name="description"
           type="j2ee:descriptionType"
           minOccurs="0"
           maxOccurs="unbounded"/>
      <xsd:element name="config-property-name"
           type="j2ee:config-property-nameType"/>
      <xsd:element name="config-property-type"
           type="j2ee:config-property-typeType"/>
      <xsd:element name="config-property-value"
           type="j2ee:xsdStringType"
           minOccurs="0">
    <xsd:annotation>
      <xsd:documentation>
        <![CDATA[

          The element config-property-value contains the value
          of a configuration entry. Note, it is possible for a
          resource adapter deployer to override this
          configuration information during deployment.

          Example:
          <config-property-value>WombatServer</config-property-value>

          ]]>
      </xsd:documentation>
    </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
    <xsd:attribute name="id" type="xsd:ID"/>
  </xsd:complexType>

<!-- **************************************************** -->
```

```
<xsd:complexType name="connection-definitionType">
  <xsd:annotation>
    <xsd:documentation>

  The connection-definitionType defines a set of connection
  interfaces and classes pertaining to a particular connection
  type. This also includes configurable properties for
  ManagedConnectionFactory instances that may be produced out
  of this set.

    </xsd:documentation>
  </xsd:annotation>

  <xsd:sequence>
    <xsd:element name="managedconnectionfactory-class"
         type="j2ee:fully-qualified-classType">
  <xsd:annotation>
    <xsd:documentation>
      <![CDATA[

        The element managedconnectionfactory-class specifies
        the fully qualified name of the Java class that
        implements the
        javax.resource.spi.ManagedConnectionFactory interface.
        This Java class is provided as part of resource
        adapter's implementation of connector architecture
        specified contracts. The implementation of this
        class is required to be a JavaBean.

        Example:
        <managedconnectionfactory-class>
        com.wombat.ManagedConnectionFactoryImpl
        </managedconnectionfactory-class>

        ]]>
    </xsd:documentation>
  </xsd:annotation>
    </xsd:element>
    <xsd:element name="config-property"
         type="j2ee:config-propertyType"
         minOccurs="0"
         maxOccurs="unbounded"/>
    <xsd:element name="connectionfactory-interface"
         type="j2ee:fully-qualified-classType">
  <xsd:annotation>
    <xsd:documentation>
      <![CDATA[

        The element connectionfactory-interface specifies
        the fully qualified name of the ConnectionFactory
        interface supported by the resource adapter.

        Example:
        <connectionfactory-interface>com.wombat.ConnectionFactory
        </connectionfactory-interface>

        OR

        <connectionfactory-interface>javax.resource.cci.ConnectionFactory
```

```
        </connectionfactory-interface>

        ]]>
    </xsd:documentation>
  </xsd:annotation>
    </xsd:element>

    <xsd:element name="connectionfactory-impl-class"
         type="j2ee:fully-qualified-classType">
  <xsd:annotation>
    <xsd:documentation>
      <![CDATA[

        The element connectionfactory-impl-class specifies
        the fully qualified name of the ConnectionFactory
        class that implements resource adapter
        specific ConnectionFactory interface.

        Example:

        <connectionfactory-impl-class>com.wombat.ConnectionFactoryImpl
        </connectionfactory-impl-class>

        ]]>
    </xsd:documentation>
  </xsd:annotation>
    </xsd:element>
    <xsd:element name="connection-interface"
         type="j2ee:fully-qualified-classType">
  <xsd:annotation>
    <xsd:documentation>
      <![CDATA[

        The connection-interface element specifies the fully
        qualified name of the Connection interface supported
        by the resource adapter.

        Example:

        <connection-interface>javax.resource.cci.Connection
        </connection-interface>

        ]]>
    </xsd:documentation>
  </xsd:annotation>
    </xsd:element>
    <xsd:element name="connection-impl-class"
         type="j2ee:fully-qualified-classType">
  <xsd:annotation>
    <xsd:documentation>
      <![CDATA[

        The connection-impl-classType specifies the fully
        qualified name of the Connection class that
        implements resource adapter specific Connection
        interface.  It is used by the connection-impl-class
        elements.

        Example:
```

```
          <connection-impl-class>com.wombat.ConnectionImpl
          </connection-impl-class>

          ]]>
      </xsd:documentation>
    </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
    <xsd:attribute name="id" type="xsd:ID"/>
  </xsd:complexType>

<!-- ********************************************** -->

  <xsd:complexType name="connectorType">
    <xsd:annotation>
      <xsd:documentation>

    The connectorType defines a resource adapter.

      </xsd:documentation>
    </xsd:annotation>

    <xsd:sequence>
      <xsd:group ref="j2ee:descriptionGroup"/>
      <xsd:element name="vendor-name"
           type="j2ee:xsdStringType">
    <xsd:annotation>
      <xsd:documentation>

        The element vendor-name specifies the name of
        resource adapter provider vendor.

      </xsd:documentation>
    </xsd:annotation>
      </xsd:element>
      <xsd:element name="eis-type"
           type="j2ee:xsdStringType">
    <xsd:annotation>
      <xsd:documentation>

        The element eis-type contains information about the
        type of the EIS. For example, the type of an EIS can
        be product name of EIS independent of any version
        info.

        This helps in identifying EIS instances that can be
        used with this resource adapter.

      </xsd:documentation>
    </xsd:annotation>
      </xsd:element>
      <xsd:element name="resourceadapter-version"
           type="j2ee:xsdStringType">
    <xsd:annotation>
      <xsd:documentation>

        The element resourceadapter-version specifies a string-based version
        of the resource adapter from the resource adapter
```

```
        provider.

      </xsd:documentation>
    </xsd:annotation>

      </xsd:element>
      <xsd:element name="license"
           type="j2ee:licenseType"
           minOccurs="0"/>
      <xsd:element name="resourceadapter"
           type="j2ee:resourceadapterType"/>

    </xsd:sequence>
    <xsd:attribute name="version"
           type="j2ee:dewey-versionType"
           fixed="1.5"
           use="required">
      <xsd:annotation>
    <xsd:documentation>

      The version specifies the version of the
      connector architecture specification that is
      supported by this resource adapter. This information
      enables deployer to configure the resource adapter to
      support deployment and runtime requirements of the
      corresponding connector architecture specification.

    </xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="id" type="xsd:ID"/>
  </xsd:complexType>

<!-- **************************************************** -->

  <xsd:complexType name="credential-interfaceType">
    <xsd:annotation>
      <xsd:documentation>

    The credential-interfaceType specifies the
    interface that the resource adapter implementation
    supports for the representation of the
    credentials. This element(s) that use this type,
    i.e. credential-interface,  should be used by
    application server to find out the Credential
    interface it should use as part of the security
    contract.

    The possible values are:

    javax.resource.spi.security.PasswordCredential
    org.ietf.jgss.GSSCredential
    javax.resource.spi.security.GenericCredential

      </xsd:documentation>
    </xsd:annotation>
    <xsd:simpleContent>
      <xsd:restriction base="j2ee:fully-qualified-classType">
    <xsd:enumeration
```

```
          value="javax.resource.spi.security.PasswordCredential"/>
      <xsd:enumeration
          value="org.ietf.jgss.GSSCredential"/>
      <xsd:enumeration
          value="javax.resource.spi.security.GenericCredential"/>
        </xsd:restriction>
      </xsd:simpleContent>
    </xsd:complexType>

<!-- ********************************************** -->

    <xsd:complexType name="inbound-resourceadapterType">
      <xsd:annotation>
        <xsd:documentation>

      The inbound-resourceadapterType specifies information
      about an inbound resource adapter. This contains information
      specific to the implementation of the resource adapter
      library as specified through the messageadapter element.

        </xsd:documentation>
      </xsd:annotation>

      <xsd:sequence>
        <xsd:element name="messageadapter"
            type="j2ee:messageadapterType"
            minOccurs="0">
      <xsd:unique name="messagelistener-type-uniqueness">
        <xsd:annotation>
          <xsd:documentation>

            The messagelistener-type element content must be
            unique in the messageadapter. Several messagelisteners
            can not use the same messagelistener-type.

          </xsd:documentation>
        </xsd:annotation>
        <xsd:selector xpath="j2ee:messagelistener"/>
        <xsd:field    xpath="j2ee:messagelistener-type"/>
      </xsd:unique>
        </xsd:element>
      </xsd:sequence>
      <xsd:attribute name="id" type="xsd:ID"/>
    </xsd:complexType>

<!-- ********************************************** -->

    <xsd:complexType name="licenseType">
      <xsd:annotation>
        <xsd:documentation>

      The licenseType specifies licensing requirements for the
      resource adapter module. This type specifies whether a
      license is required to deploy and use this resource adapter,
      and an optional description of the licensing terms
      (examples: duration of license, number of connection
      restrictions). It is used by the license element.

        </xsd:documentation>
```

```
    </xsd:annotation>

    <xsd:sequence>
      <xsd:element name="description"
           type="j2ee:descriptionType"
           minOccurs="0"
           maxOccurs="unbounded"/>
      <xsd:element name="license-required"
           type="j2ee:true-falseType">
    <xsd:annotation>
      <xsd:documentation>

        The element license-required specifies whether a
        license is required to deploy and use the
        resource adapter. This element must be one of
        the following, "true" or "false".

      </xsd:documentation>
    </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
    <xsd:attribute name="id" type="xsd:ID"/>
  </xsd:complexType>

<!-- ********************************************** -->

  <xsd:complexType name="messageadapterType">
    <xsd:annotation>
      <xsd:documentation>

    The messageadapterType specifies information about the
    messaging capabilities of the resource adapter. This
    contains information specific to the implementation of the
    resource adapter library as specified through the
    messagelistener element.

      </xsd:documentation>
    </xsd:annotation>

    <xsd:sequence>
      <xsd:element name="messagelistener"
           type="j2ee:messagelistenerType"
           maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="id" type="xsd:ID"/>
  </xsd:complexType>

<!-- ********************************************** -->

  <xsd:complexType name="messagelistenerType">
    <xsd:annotation>
      <xsd:documentation>

    The messagelistenerType specifies information about a
    specific message listener supported by the messaging
    resource adapter. It contains information on the Java type
    of the message listener interface and an activation
    specification.
```

```
      </xsd:documentation>
    </xsd:annotation>

    <xsd:sequence>
      <xsd:element name="messagelistener-type"
           type="j2ee:fully-qualified-classType">
    <xsd:annotation>
      <xsd:documentation>
        <![CDATA[

          The element messagelistener-type specifies the fully
          qualified name of the Java type of a message
          listener interface.

          Example:

        <messagelistener-type>javax.jms.MessageListener
        </messagelistener-type>

          ]]>
      </xsd:documentation>
    </xsd:annotation>

      </xsd:element>
      <xsd:element name="activationspec"
           type="j2ee:activationspecType"/>
    </xsd:sequence>
    <xsd:attribute name="id" type="xsd:ID"/>
  </xsd:complexType>

<!-- ********************************************** -->

  <xsd:complexType name="outbound-resourceadapterType">
    <xsd:annotation>
      <xsd:documentation>

    The outbound-resourceadapterType specifies information about
    an outbound resource adapter. The information includes fully
    qualified names of classes/interfaces required as part of
    the connector architecture specified contracts for
    connection management, level of transaction support
    provided, one or more authentication mechanisms supported
    and additional required security permissions.

    If there is no authentication-mechanism specified as part of
    resource adapter element then the resource adapter does not
    support any standard security authentication mechanisms as
    part of security contract. The application server ignores
    the security part of the system contracts in this case.

      </xsd:documentation>
    </xsd:annotation>

    <xsd:sequence>
      <xsd:element name="connection-definition"
           type="j2ee:connection-definitionType"
           maxOccurs="unbounded"/>
      <xsd:element name="transaction-support"
           type="j2ee:transaction-supportType"/>
```

```
      <xsd:element name="authentication-mechanism"
           type="j2ee:authentication-mechanismType"
           minOccurs="0"
           maxOccurs="unbounded"/>
      <xsd:element name="reauthentication-support"
           type="j2ee:true-falseType">
    <xsd:annotation>
      <xsd:documentation>

        The element reauthentication-support specifies
        whether the resource adapter implementation supports
        re-authentication of existing Managed- Connection
        instance. Note that this information is for the
        resource adapter implementation and not for the
        underlying EIS instance. This element must have
        either a "true" or "false" value.

      </xsd:documentation>
    </xsd:annotation>

      </xsd:element>
    </xsd:sequence>
    <xsd:attribute name="id" type="xsd:ID"/>
  </xsd:complexType>

<!-- ********************************************** -->

  <xsd:complexType name="required-config-propertyType">
    <xsd:annotation>
      <xsd:documentation>
    <![CDATA[

      The required-config-propertyType contains a declaration
      of a single configuration property used for specifying a
      required configuration property name. It is used
      by required-config-property elements.

      Example:

      <required-config-property>Destination</required-config-property>

      ]]>
      </xsd:documentation>
    </xsd:annotation>

    <xsd:sequence>
      <xsd:element name="description"
           type="j2ee:descriptionType"
           minOccurs="0"
           maxOccurs="unbounded"/>
      <xsd:element name="config-property-name"
           type="j2ee:config-property-nameType"/>
    </xsd:sequence>
    <xsd:attribute name="id" type="xsd:ID"/>
  </xsd:complexType>

<!-- ********************************************** -->

  <xsd:complexType name="resourceadapterType">
```

```
    <xsd:annotation>
      <xsd:documentation>

    The resourceadapterType specifies information about the
    resource adapter. The information includes fully qualified
    resource adapter Java class name, configuration properties,
    information specific to the implementation of the resource
    adapter library as specified through the
    outbound-resourceadapter and inbound-resourceadapter
    elements, and an optional set of administered objects.

      </xsd:documentation>
    </xsd:annotation>

    <xsd:sequence>
      <xsd:element name="resourceadapter-class"
           type="j2ee:fully-qualified-classType"
           minOccurs="0">
    <xsd:annotation>
      <xsd:documentation>

        The element resourceadapter-class specifies the
        fully qualified name of a Java class that implements
        the javax.resource.spi.ResourceAdapter
        interface. This Java class is provided as part of
        resource adapter's implementation of connector
        architecture specified contracts. The implementation
        of this class is required to be a JavaBean.

      </xsd:documentation>
    </xsd:annotation>

      </xsd:element>
      <xsd:element name="config-property"
           type="j2ee:config-propertyType"
           minOccurs="0"
           maxOccurs="unbounded"/>
      <xsd:element name="outbound-resourceadapter"
           type="j2ee:outbound-resourceadapterType"
           minOccurs="0">
    <xsd:unique name="connectionfactory-interface-uniqueness">
      <xsd:annotation>
        <xsd:documentation>

          The connectionfactory-interface element content
          must be unique in the outbound-resourceadapter.
          Multiple connection-definitions can not use the
          same connectionfactory-type.

        </xsd:documentation>
      </xsd:annotation>
      <xsd:selector xpath="j2ee:connection-definition"/>
      <xsd:field    xpath="j2ee:connectionfactory-interface"/>
    </xsd:unique>
      </xsd:element>
      <xsd:element name="inbound-resourceadapter"
           type="j2ee:inbound-resourceadapterType"
           minOccurs="0"/>
      <xsd:element name="adminobject"
```

```
            type="j2ee:adminobjectType"
            minOccurs="0"
            maxOccurs="unbounded"/>
      <xsd:element name="security-permission"
            type="j2ee:security-permissionType"
            minOccurs="0"
            maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:attribute name="id" type="xsd:ID"/>
  </xsd:complexType>

<!-- ********************************************** -->

  <xsd:complexType name="security-permissionType">
    <xsd:annotation>
      <xsd:documentation>

    The security-permissionType specifies a security
    permission that is required by the resource adapter code.

    The security permission listed in the deployment descriptor
    are ones that are different from those required by the
    default permission set as specified in the connector
    specification. The optional description can mention specific
    reason that resource adapter requires a given security
    permission.

      </xsd:documentation>
    </xsd:annotation>

    <xsd:sequence>
      <xsd:element name="description"
            type="j2ee:descriptionType"
            minOccurs="0"
            maxOccurs="unbounded"/>
      <xsd:element name="security-permission-spec"
            type="j2ee:xsdStringType">
    <xsd:annotation>
      <xsd:documentation>

        The element security-permission-spec specifies a security
        permission based on the Security policy file
        syntax. Refer to the following URL for Sun's
        implementation of the security permission
        specification:

        http://java.sun.com/products/jdk/1.4/docs/guide/security/PolicyFiles.html#FileSyntax
      </xsd:documentation>
    </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
    <xsd:attribute name="id" type="xsd:ID"/>
  </xsd:complexType>

<!-- ********************************************** -->

  <xsd:complexType name="transaction-supportType">
    <xsd:annotation>
      <xsd:documentation>
```

```
    The transaction-supportType specifies the level of
    transaction support provided by the resource adapter. It is
    used by transaction-support elements.

    The value must be one of the following:

        NoTransaction
        LocalTransaction
        XATransaction

      </xsd:documentation>
    </xsd:annotation>
    <xsd:simpleContent>
      <xsd:restriction base="j2ee:string">
    <xsd:enumeration value="NoTransaction"/>
    <xsd:enumeration value="LocalTransaction"/>
    <xsd:enumeration value="XATransaction"/>
      </xsd:restriction>
    </xsd:simpleContent>
  </xsd:complexType>

</xsd:schema>
```

A.3. Java EE Connector Architecture 1.0

```
      <!--
DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS HEADER.

Copyright 2000-2007 Sun Microsystems, Inc. All rights reserved.

The contents of this file are subject to the terms of either the GNU
General Public License Version 2 only ("GPL") or the Common Development
and Distribution License("CDDL") (collectively, the "License").  You
may not use this file except in compliance with the License. You can obtain
a copy of the License at https://glassfish.dev.java.net/public/CDDL+GPL.html
or glassfish/bootstrap/legal/LICENSE.txt.  See the License for the specific
language governing permissions and limitations under the License.

When distributing the software, include this License Header Notice in each
file and include the License file at glassfish/bootstrap/legal/LICENSE.txt.
Sun designates this particular file as subject to the "Classpath" exception
as provided by Sun in the GPL Version 2 section of the License file that
accompanied this code.  If applicable, add the following below the License
Header, with the fields enclosed by brackets [] replaced by your own
identifying information: "Portions Copyrighted [year]
[name of copyright owner]"

Contributor(s):

If you wish your version of this file to be governed by only the CDDL or
only the GPL Version 2, indicate your decision by adding "[Contributor]
```

```
elects to include this software in this distribution under the [CDDL or GPL
Version 2] license."  If you don't indicate a single choice of license, a
recipient has the option to distribute your version of this file under
either the CDDL, the GPL Version 2 or to extend the choice of license to
its licensees as provided above.  However, if you add GPL Version 2 code
and therefore, elected the GPL Version 2 license, then the option applies
only if the new code is made subject to such option by the copyright
holder.
-->

<!--
This is the XML DTD for the Connector 1.0 deployment descriptor.
All Connector 1.0 deployment descriptors must include a DOCTYPE
of the following form:

  <!DOCTYPE connector PUBLIC
    "-//Sun Microsystems, Inc.//DTD Connector 1.0//EN"
    "http://java.sun.com/dtd/connector_1_0.dtd">

-->

<!--
The following conventions apply to all J2EE deployment descriptor
elements unless indicated otherwise.

- In elements that contain PCDATA, leading and trailing whitespace
  in the data may be ignored.

- In elements whose value is an "enumerated type", the value is
  case sensitive.

- In elements that specify a pathname to a file within the same
  JAR file, relative filenames (i.e., those not starting with "/")
  are considered relative to the root of the JAR file's namespace.
  Absolute filenames (i.e., those starting with "/") also specify
  names in the root of the JAR file's namespace.  In general, relative
  names are preferred.  The exception is .war files where absolute
  names are preferred for consistency with the servlet API.
-->


<!--
The connector element is the root element of the deployment descriptor
for the resource adapter. This element includes general information - vendor
name, version, specification version supported, icon -  about the
resource adapter module. It also includes information specific to the
implementation of the resource adapter library as specified through
the element resourceadapter.
-->
<!ELEMENT connector (display-name?, description?, icon?, vendor-name,
spec-version, eis-type, version, license?, resourceadapter)>

<!--
The element authentication-mechanism specifies an authentication mechanism
supported by the resource adapter. Note that this support is for
the resource adapter and not for the underlying EIS instance. The
optional description specifies any resource adapter specific requirement
for the support of security contract and authentication mechanism.
```

```
Note that BasicPassword mechanism type should support the
javax.resource.spi.security.PasswordCredential interface.
The Kerbv5 mechanism type should support the
javax.resource.spi.security.GenericCredential interface.

Used in: resourceadapter
-->
<!ELEMENT authentication-mechanism (
description?, authentication-mechanism-type, credential-interface)>

<!--
The element authentication-mechanism-type specifies type of an authentication
mechanism.

The example values are:
   <authentication-mechanism-type>BasicPassword
                  </authentication-mechanism-type>
   <authentication-mechanism-type>Kerbv5
                  </authentication-mechanism-type>

Any additional security mechanisms are outside the scope of the
Connector architecture specification.

Used in: authentication-mechanism
-->
<!ELEMENT authentication-mechanism-type (#PCDATA)>

<!--
The element config-property contains a declaration of a single
configuration property for a ManagedConnectionFactory instance.

Each ManagedConnectionFactory instance creates connections to a
specific EIS instance based on the properties configured on the
ManagedConnectionFactory instance. The configurable properties are
specified only once in the deployment descriptor, even though a
resource adapter can be used to configure multiple ManagedConnnection-
Factory instances (that create connections to different instances of
the same EIS).

The declaration consists of an optional description, name, type
and an optional value of the configuration property. If the resource
adapter provider does not specify a value than the deployer is
responsible for providing a valid value for a configuration property.

Any bounds or well-defined values of properties should be described
in the description element.

Used in: resourceadapter
-->
<!ELEMENT config-property (description?, config-property-name,
config-property-type, config-property-value?)>

<!--
The element config-property-name contains the name of a configuration
property.

The connector architecture defines a set of well-defined properties
all of type java.lang.String. These are as follows:
     <config-property-name>ServerName</config-property-name>
```

```
     <config-property-name>PortNumber</config-property-name>
     <config-property-name>UserName</config-property-name>
     <config-property-name>Password</config-property-name>
     <config-property-name>ConnectionURL</config-property-name>

A resource adapter provider can extend this property set to include
properties specific to the resource adapter and its underlying EIS.

Used in: config-property

Example: <config-property-name>ServerName</config-property-name>
-->
<!ELEMENT config-property-name (#PCDATA)>

<!--
The element config-property-type contains the fully qualified Java
type of a configuration property as required by ManagedConnection-
Factory instance.

The following are the legal values of config-property-type:
   java.lang.Boolean, java.lang.String, java.lang.Integer,
   java.lang.Double, java.lang.Byte, java.lang.Short,
   java.lang.Long, java.lang.Float, java.lang.Character

Used in: config-property

Example: <config-property-type>java.lang.String</config-property-type>
-->
<!ELEMENT config-property-type (#PCDATA)>

<!--
The element config-property-value contains the value of a configuration
entry.

Used in: config-property

Example: <config-property-value>WombatServer</config-property-value>
-->
<!ELEMENT config-property-value (#PCDATA)>

<!--
The element connection-impl-class specifies the fully-qualified
name of the Connection class that implements resource adapter
specific Connection interface.

Used in: resourceadapter

Example: <connection-impl-class>com.wombat.ConnectionImpl
         </connection-impl-class>
-->
<!ELEMENT connection-impl-class (#PCDATA)>

<!--
The element connection-interface specifies the fully-qualified
name of the Connection interface supported by the resource
adapter.

Used in: resourceadapter
```

```
Example: <connection-interface>javax.resource.cci.Connection
         </connection-interface>
-->
<!ELEMENT connection-interface (#PCDATA)>

<!--
The element connectionfactory-impl-class specifies the fully-qualified
name of the ConnectionFactory class that implements resource adapter
specific ConnectionFactory interface.

Used in: resourceadapter

Example: <connectionfactory-impl-class>com.wombat.ConnectionFactoryImpl
         </connectionfactory-impl-class>
-->
<!ELEMENT connectionfactory-impl-class (#PCDATA)>

<!--
The element connectionfactory-interface specifies the fully-qualified
name of the ConnectionFactory interface supported by the resource
adapter.

Used in: resourceadapter

Example: <connectionfactory-interface>com.wombat.ConnectionFactory
         </connectionfactory-interface>
OR
<connectionfactory-interface>javax.resource.cci.ConnectionFactory
         </connectionfactory-interface>
-->
<!ELEMENT connectionfactory-interface (#PCDATA)>

<!--
The element credential-interface specifies the interface that the
resource adapter implementation supports for the representation
of the credentials. This element should be used by application server
to find out the Credential interface it should use as part of the
security contract.

The possible values are:
   <credential-interface>javax.resource.spi.security.PasswordCredential
        </credential-interface>
   <credential-interface>javax.resource.spi.security.GenericCredential
        </credential-interface>

Used in: authentication-mechanism
-->
<!ELEMENT credential-interface (#PCDATA)>

<!--
The description element is used to provide text describing the parent
element.  The description element should include any information that
the resource adapter rar file producer wants to provide to the consumer of
the resource adapter rar file (i.e., to the Deployer). Typically, the tools
used by the resource adapter rar file consumer will display the description
when processing the parent element that contains the description.

Used in: authentication-mechanism, config-property, connector, license,
security-permission
```

```
-->
<!ELEMENT description (#PCDATA)>

<!--
The display-name element contains a short name that is intended to be
displayed by tools.  The display name need not be unique.

Used in: connector

Example:

<display-name>Employee Self Service</display-name>
-->
<!ELEMENT display-name (#PCDATA)>

<!--
The element eis-type contains information about the type of the
EIS. For example, the type of an EIS can be product name of EIS
independent of any version info.

This helps in identifying EIS instances that can be used with
this resource adapter.

Used in: connector
-->
<!ELEMENT eis-type (#PCDATA)>

<!--
The icon element contains small-icon and large-icon elements that
specify the file names for small and a large GIF or JPEG icon images
used to represent the parent element in a GUI tool.

Used in: connector
-->
<!ELEMENT icon (small-icon?, large-icon?)>

<!--
The large-icon element contains the name of a file
containing a large (32 x 32) icon image. The file
name is a relative path within the resource adapter's
rar file.

The image may be either in the JPEG or GIF format.
The icon can be used by tools.

Used in: icon

Example:

<large-icon>employee-service-icon32x32.jpg</large-icon>
-->
<!ELEMENT large-icon (#PCDATA)>

<!--
The element license specifies licensing requirements for the resource
adapter module. This element specifies whether a license is required
to deploy and use this resource adapter, and an optional description
of the licensing terms (examples: duration of license, number of
connection restrictions).
```

```
Used in: connector
-->
<!ELEMENT license (description?, license-required)>

<!--
The element license-required specifies whether a license is required
to deploy and use the resource adapter. This element must be one of
the following:

    <license-required>true</license-required>
    <license-required>false</license-required>

Used in: license
-->
<!ELEMENT license-required (#PCDATA)>

<!--
The element managedconnectionfactory-class specifies the fully qualified
name of the Java class that implements the javax.resource.spi.Managed-
ConnectionFactory interface. This Java class is provided as part of
resource adapter's implementation of connector architecture specified
contracts.

Used in: resourceadapter

Example:
  <managedconnectionfactory-class>com.wombat.ManagedConnectionFactoryImpl
  </managedconnectionfactory-class>
-->
<!ELEMENT managedconnectionfactory-class (#PCDATA)>

<!--
The element reauthentication-support specifies whether the resource
adapter implementation supports re-authentication of existing Managed-
Connection instance. Note that this information is for the resource
adapter implementation and not for the underlying EIS instance.

This element must be one of the following:
        <reauthentication-support>true</reauthentication-support>
        <reauthentication-support>false</reauthentication-support>
Used in: resourceadapter
-->
<!ELEMENT reauthentication-support (#PCDATA)>

<!--
The element resourceadapter specifies information about the resource
adapter. The information includes fully-qualified names of
class/interfaces required as part of the connector architecture
specified contracts, level of transaction support provided,
configurable properties for ManagedConnectionFactory instances,
one or more authentication mechanisms supported and additional
required security permissions.

If there is no authentication-mechanism specified as part of
resource adapter element then the resource adapter does not
support any standard security authentication mechanisms as part
of security contract. The application server ignores the security
part of the system contracts in this case.
```

```
Used in: connector
-->
<!ELEMENT resourceadapter (
managedconnectionfactory-class, connectionfactory-interface,
connectionfactory-impl-class, connection-interface,
connection-impl-class, transaction-support, config-property*,
authentication-mechanism*, reauthentication-support, security-permission*
)>

<!--
The element security permission specifies a security permission that
is required by the resource adapter code.

The security permission listed in the deployment descriptor are ones
that are different from those required by the default permission set
as specified in the connector specification. The optional description
can mention specific reason that resource adapter requires a given
security permission.

Used in: resourceadapter
-->
<!ELEMENT security-permission (description?, security-permission-spec)>

<!--
The element permission-spec specifies a security permission based
on the Security policy file syntax. Refer to the following URL for
Sun's implementation of the security permission specification:

http://java.sun.com/products/jdk/1.3/docs/guide/security/PolicyFiles.html#FileSyntax

Used in: security-permission
-->
<!ELEMENT security-permission-spec (#PCDATA)>

<!--
The small-icon element contains the name of a file
containing a small (16 x 16) icon image. The file
name is a relative path within the resource adapter's
rar file.

The image may be either in the JPEG or GIF format.
The icon can be used by tools.

Used in: icon

Example:

<small-icon>employee-service-icon16x16.jpg</small-icon>
-->
<!ELEMENT small-icon (#PCDATA)>

<!--
The element spec-version specifies the version of the connector
architecture specification that is supported by this resource
adapter. This information enables deployer to configure the resource
adapter to support deployment and runtime requirements of the
corresponding connector architecture specification.
```

```
Used in: connector

Example:
  <spec-version>1.0</spec-version>
-->
<!ELEMENT spec-version (#PCDATA)>

<!--
The transaction-support element specifies the level of transaction
support provided by the resource adapter.
The value of transaction-support must be one of the following:
 <transaction-support>NoTransaction</transaction-support>
 <transaction-support>LocalTransaction</transaction-support>
 <transaction-support>XATransaction</transaction-support>

Used in: resourceadapter
-->
<!ELEMENT transaction-support (#PCDATA)>

<!--
The element vendor-name specifies the name of resource adapter provider
vendor.

Used in: connector

Example:
  <vendor-name>Wombat Corp.</vendor-name>
-->
<!ELEMENT vendor-name (#PCDATA)>

<!--
The element version specifies a string-based version of the
resource adapter from the resource adapter provider.

Used in: connector

Example:
  <version>1.0</version>
-->
<!ELEMENT version (#PCDATA)>

<!--
The ID mechanism is to allow tools that produce additional deployment
information (i.e., information beyond the standard deployment
descriptor information) to store the non-standard information in a
separate file, and easily refer from these tool-specific files to the
information in the standard deployment descriptor.

Tools are not allowed to add the non-standard information into the
standard deployment descriptor.
-->

<!ATTLIST authentication-mechanism id ID #IMPLIED>
<!ATTLIST authentication-mechanism-type id ID #IMPLIED>
<!ATTLIST config-property id ID #IMPLIED>
<!ATTLIST config-property-name id ID #IMPLIED>
<!ATTLIST config-property-type id ID #IMPLIED>
<!ATTLIST config-property-value id ID #IMPLIED>
<!ATTLIST connection-impl-class id ID #IMPLIED>
```

```
<!ATTLIST connection-interface id ID #IMPLIED>
<!ATTLIST connectionfactory-impl-class id ID #IMPLIED>
<!ATTLIST connectionfactory-interface id ID #IMPLIED>
<!ATTLIST connector id ID #IMPLIED>
<!ATTLIST credential-interface id ID #IMPLIED>
<!ATTLIST description id ID #IMPLIED>
<!ATTLIST display-name id ID #IMPLIED>
<!ATTLIST eis-type id ID #IMPLIED>
<!ATTLIST icon id ID #IMPLIED>
<!ATTLIST large-icon id ID #IMPLIED>
<!ATTLIST license id ID #IMPLIED>
<!ATTLIST license-required id ID #IMPLIED>
<!ATTLIST managedconnectionfactory-class id ID #IMPLIED>
<!ATTLIST reauthentication-support id ID #IMPLIED>
<!ATTLIST resourceadapter id ID #IMPLIED>
<!ATTLIST security-permission id ID #IMPLIED>
<!ATTLIST security-permission-spec id ID #IMPLIED>
<!ATTLIST small-icon id ID #IMPLIED>
<!ATTLIST spec-version id ID #IMPLIED>
<!ATTLIST transaction-support id ID #IMPLIED>
<!ATTLIST vendor-name id ID #IMPLIED>
<!ATTLIST version id ID #IMPLIED>
```

A.4. IronJacamar 1.0

```
      <?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
           elementFormDefault="qualified"
           targetNamespace="http://www.jboss.org/ironjacamar/schema"
           xmlns="http://www.jboss.org/ironjacamar/schema"
           version="1.0">

  <xs:complexType name="boolean-presenceType"></xs:complexType>

  <xs:complexType name="config-propertyType" mixed="true">
    <xs:annotation>
      <xs:documentation>
        <![CDATA[
          Specifies an override for a config-property element in ra.xml or a @ConfigProperty
        ]]>
      </xs:documentation>
    </xs:annotation>
    <xs:simpleContent>
      <xs:extension base="xs:token">
        <xs:attribute use="required" name="name" type="xs:token">
          <xs:annotation>
            <xs:documentation>
              <![CDATA[
                Specifies the name of the config-property
              ]]>
            </xs:documentation>
          </xs:annotation>
        </xs:attribute>
```

```
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>

  <xs:complexType name="ironjacamarType">
    <xs:sequence>
                        <xs:element    name="bean-validation-groups"    type="bean-validation-
groupsType" minOccurs="0" maxOccurs="1">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Specifies bean validation group that should be used
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="bootstrap-context" type="xs:token" minOccurs="0" maxOccurs="1">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Specifies the unique name of the bootstrap context that should be used
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
                                    <xs:element      name="config-property"      type="config-
propertyType" minOccurs="0" maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
               The config-property specifies resource adapter configuration properties.
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="transaction-support" type="transaction-supportType" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Specifies the transaction support level of the resource adapter
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
                            <xs:element     name="connection-definitions"     type="connection-
definitionsType" minOccurs="0" maxOccurs="1">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Specifies the connection definitions
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="admin-objects" type="admin-objectsType" minOccurs="0" maxOccurs="1">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Specifies the administration objects
```

```
            ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>

  <xs:simpleType name="transaction-supportType">
    <xs:annotation>
      <xs:documentation>
        <![CDATA[
          Define the type of transaction supported by this resource adapter.
          Valid values are: NoTransaction, LocalTransaction, XATransaction
         ]]>
      </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:token">
      <xs:enumeration value="NoTransaction" />
      <xs:enumeration value="LocalTransaction" />
      <xs:enumeration value="XATransaction" />
    </xs:restriction>
  </xs:simpleType>

  <xs:attributeGroup name="common-attribute">
    <xs:attribute name="class-name" type="xs:token" use="optional">
      <xs:annotation>
        <xs:documentation>
          <![CDATA[
            Specifies the the fully qualified class name of a managed connection factory
            or admin object
           ]]>
        </xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="jndi-name" type="xs:token" use="required">
      <xs:annotation>
        <xs:documentation>
          <![CDATA[
            Specifies the JNDI name
           ]]>
        </xs:documentation>
      </xs:annotation>
    </xs:attribute>

  <xs:attribute name="enabled" type="xs:boolean" default="true" form="unqualified" use="optional">
      <xs:annotation>
        <xs:documentation>
          <![CDATA[
            Should the object in question be activated
           ]]>
        </xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute default="true" name="use-java-context" type="xs:boolean">
      <xs:annotation>
        <xs:documentation>
          <![CDATA[
            Specifies if a java:/ JNDI context should be used
           ]]>
```

```
        </xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="pool-name" type="xs:token" use="optional">
      <xs:annotation>
        <xs:documentation>
          <![CDATA[
            Specifies the pool name for the object
           ]]>
        </xs:documentation>
      </xs:annotation>
    </xs:attribute>
  </xs:attributeGroup>

  <xs:complexType name="admin-objectType">
    <xs:sequence>
                                    <xs:element      name="config-property"      type="config-
propertyType" minOccurs="0" maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Specifies an administration object configuration property.
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
    <xs:attributeGroup ref="common-attribute"></xs:attributeGroup>
  </xs:complexType>

  <xs:complexType name="timeoutType">
    <xs:sequence>
      <xs:element name="blocking-timeout-millis" type="xs:nonNegativeInteger" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
                The blocking-timeout-millis element indicates 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 30000 (30 seconds).
              ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="idle-timeout-minutes" type="xs:nonNegativeInteger" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              The idle-timeout-minutes elements indicates 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.
              ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
                                                          <xs:element          name="allocation-
retry" type="xs:nonNegativeInteger" minOccurs="0" maxOccurs="1">
```

```
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              The allocation retry element indicates the number of times that allocating
              a connection should be tried before throwing an exception. The default is
              0.
              ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
                                                <xs:element        name="allocation-retry-wait-
millis" type="xs:nonNegativeInteger" minOccurs="0" maxOccurs="1">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              The allocation retry wait millis element indicates the time in milliseconds
             to wait between retrying to allocate a connection. The default is 5000 (5 seconds).
              ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
                                                          <xs:element          name="xa-resource-
timeout" type="xs:nonNegativeInteger" minOccurs="0" maxOccurs="1">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
                 Passed to XAResource.setTransactionTimeout(). Default is zero which does not
 invoke the setter.
              Specified in seconds - e.g. 5 minutes
              <xa-resource-timeout>300</xa-resource-timeout>
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="validationType">
    <xs:sequence>
      <xs:element name="background-validation" type="xs:boolean" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              An element to specify that connections should be validated on a background
              thread versus being validated prior to use
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
     <xs:element name="background-validation-millis" type="xs:nonNegativeInteger" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              The background-validation-millis element specifies the amount of
              time, in millis, that background validation will run.
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
```

```
      <xs:element name="use-fast-fail" type="xs:boolean" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Whether fail a connection allocation on the first connection if it
              is invalid (true) or keep trying until the pool is exhausted of all potential
              connections (false). Default is false. e.g. <use-fast-fail>true</use-fast-fail>
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="connection-definitionsType">
    <xs:sequence>
                              <xs:element     name="connection-definition"     type="connection-
defintionType" minOccurs="1" maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Specifies a connection definition
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="connection-defintionType">
    <xs:sequence>
                                    <xs:element      name="config-property"      type="config-
propertyType" minOccurs="0" maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
             The config-property specifies managed connection factory configuration properties.
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:choice>
        <xs:element name="pool" type="poolType" minOccurs="0" maxOccurs="1">
          <xs:annotation>
            <xs:documentation>
                <![CDATA[
                  Specifies pooling settings
                 ]]>
            </xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="xa-pool" type="xa-poolType" minOccurs="0" maxOccurs="1">
          <xs:annotation>
            <xs:documentation>
                <![CDATA[
                  Specifies xa-pooling settings
                 ]]>
            </xs:documentation>
          </xs:annotation>
```

```
        </xs:element>
      </xs:choice>
      <xs:element name="security" type="securityType" minOccurs="0" maxOccurs="1">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Specifies security settings
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="timeout" type="timeoutType" minOccurs="0" maxOccurs="1">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Specifies timeout settings
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="validation" type="validationType" minOccurs="0" maxOccurs="1">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Specifies validation settings
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="recovery" type="recoverType" minOccurs="0" maxOccurs="1"></xs:element>
    </xs:sequence>
    <xs:attribute name="use-ccm" type="xs:boolean" default="true" use="optional">
      <xs:annotation>
        <xs:documentation>
          <![CDATA[
            Enable cached connection manager
           ]]>
        </xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attributeGroup ref="common-attribute"></xs:attributeGroup>
  </xs:complexType>

  <xs:complexType name="poolType">
    <xs:sequence>
      <xs:element name="min-pool-size" type="xs:nonNegativeInteger" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              The min-pool-size element indicates the minimum number of connections
              a pool should hold. These are not created until a Subject is known from a
              request for a connection. This default to 0. Ex: <min-pool-size>1</min-pool-size>
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="max-pool-size" type="xs:nonNegativeInteger" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
```

```
            <![CDATA[
              The max-pool-size element indicates the maximum number of connections
            for a pool. No more than max-pool-size connections will be created in each sub-pool.
              This defaults to 20.
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="prefill" type="xs:boolean" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Whether to attempt to prefill the connection pool.
              Default is false. e.g. <prefill>false</prefill>.
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="use-strict-min" type="xs:boolean" minOccurs="0" maxOccurs="1">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Specifies if the min-pool-size should be considered strictly.
              Default false
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="flush-strategy" type="xs:token" minOccurs="0" maxOccurs="1">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Specifies how the pool should be flush in case of an error.
              Valid values are: FailingConnectionOnly (default), IdleConnections, EntirePool
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="xa-poolType">
    <xs:complexContent>
      <xs:extension base="poolType">
        <xs:sequence>
          <xs:element name="is-same-rm-override" type="xs:boolean" minOccurs="0">
            <xs:annotation>
              <xs:documentation>
                <![CDATA[
                  The is-same-rm-override element allows one to unconditionally
                  set whether the javax.transaction.xa.XAResource.isSameRM(XAResource) returns
                  true or false. Ex: <is-same-rm-override>true</is-same-rm-override>
                 ]]>
              </xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="interleaving" type="boolean-presenceType" minOccurs="0">
            <xs:annotation>
              <xs:documentation>
```

```
                <![CDATA[
                  An element to enable interleaving for XA connection factories
                  Ex: <interleaving/>
                 ]]>
              </xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="no-tx-separate-pools" type="boolean-presenceType" minOccurs="0">
            <xs:annotation>
              <xs:documentation>
                <![CDATA[
                     Oracle does not like XA connections getting used both inside and outside
a JTA transaction.
                        To workaround the problem you can create separate sub-pools for the
different contexts
                  using <no-tx-separate-pools/>
                  Ex: <no-tx-separate-pools/>
                 ]]>
              </xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="pad-xid" type="xs:boolean" default="false" minOccurs="0">
            <xs:annotation>
              <xs:documentation>
                <![CDATA[
                   Should the Xid be padded
                   Ex: <pad-xid>true</pad-xid>
                 ]]>
              </xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="wrap-xa-resource" type="xs:boolean" default="true" minOccurs="0">
            <xs:annotation>
              <xs:documentation>
                <![CDATA[
                 Should the XAResource instances be wrapped in a org.jboss.tm.XAResourceWrapper
                   instance
                   Ex: <wrap-xa-resource>true</wrap-xa-resource>
                 ]]>
              </xs:documentation>
            </xs:annotation>
          </xs:element>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="securityType">
    <xs:sequence>
      <xs:choice>
        <xs:element name="application" type="boolean-presenceType" minOccurs="0" maxOccurs="1">
          <xs:annotation>
            <xs:documentation>
              <![CDATA[
                Indicates that app supplied parameters (such as from getConnection(user, pw))
                are used to distinguish connections in the pool.
                Ex:
                <application/>
              ]]>
```

```
            </xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="security-domain" type="xs:token" minOccurs="0" maxOccurs="1">
          <xs:annotation>
            <xs:documentation>
              <![CDATA[
                  Indicates Subject (from security domain) are used 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.
                Ex:
                <security-domain>HsqlDbRealm</security-domain>
              ]]>
            </xs:documentation>
          </xs:annotation>
        </xs:element>
                                                    <xs:element      name="security-domain-and-
application" type="xs:token" minOccurs="0" maxOccurs="1">
          <xs:annotation>
            <xs:documentation>
              <![CDATA[
                Indicates that either app supplied parameters (such as from
                getConnection(user, pw)) or Subject (from security domain) are used 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.

                Ex:
                <security-domain-and-application>HsqlDbRealm</security-domain-and-application>
              ]]>
            </xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:choice>
    </xs:sequence>
  </xs:complexType>


  <xs:complexType name="admin-objectsType">
    <xs:sequence>
                                        <xs:element      name="admin-object"      type="admin-
objectType" minOccurs="1" maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Specifies the setup for an admin object
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="bean-validation-groupsType">
    <xs:sequence>
```

```
                                                    <xs:element        name="bean-validation-
group" type="xs:token" minOccurs="1" maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Specifies the fully qualified class name for a bean validation group that
              should be used for validation
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="recoverType">
    <xs:sequence>
      <xs:element name="recover-credential" type="credentialType" minOccurs="0" maxOccurs="1">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Specifies the security options used when creating a connection during recovery.
                 Note: if this credential are not specified the security credential are used
 for recover too
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="recover-plugin" type="extensionType" minOccurs="0" maxOccurs="1">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Specifies the extension plugin used in spi (core.spi.xa)
                  which can be implemented by various plugins to provide better feedback to
 the XA recovery system.
              ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="no-recovery" type="xs:boolean" default="false" use="optional">
      <xs:annotation>
        <xs:documentation>
          <![CDATA[
            Specify if the xa-datasource should be excluded from recovery.
            Default false.
           ]]>
        </xs:documentation>
      </xs:annotation>
    </xs:attribute>
  </xs:complexType>
  <xs:complexType name="extensionType">
    <xs:sequence>
      <xs:element name="config-property" type="config-propertyType"></xs:element>
    </xs:sequence>
    <xs:attribute name="class-name" type="xs:token" use="required"></xs:attribute>
  </xs:complexType>
  <xs:complexType name="credentialType">
    <xs:sequence>
      <xs:element name="user-name" type="xs:token" minOccurs="0">
        <xs:annotation>
```

```
          <xs:documentation>
              <![CDATA[
                Specify the username used when creating a new connection.
                Ex: <user-name>sa</user-name>
               ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="password" type="xs:token" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
              <![CDATA[
                Specify the password used when creating a new connection.
                Ex: <password>sa-pass</password>
               ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="security-domain" type="xs:token" minOccurs="0" maxOccurs="1">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
                 Indicates Subject (from security domain) are used 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.
              Ex:
              <security-domain>HsqlDbRealm</security-domain>
            ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>


  <xs:element name="ironjacamar" type="ironjacamarType">
    <xs:annotation>
      <xs:documentation>
        <![CDATA[
          Specifies the fully qualified class name for a bean validation group that
          should be used for validation
         ]]>
      </xs:documentation>
    </xs:annotation>
  </xs:element>
</xs:schema>
```

A.5. Resource adapters 1.0

```
      <?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
      targetNamespace="http://www.jboss.org/ironjacamar/schema"   xmlns="http://www.jboss.org/
ironjacamar/schema">

  <xs:complexType name="boolean-presenceType"></xs:complexType>

  <xs:complexType name="config-propertyType" mixed="true">
    <xs:annotation>
      <xs:documentation>
        <![CDATA[
          Specifies an override for a config-property element in ra.xml or a @ConfigProperty
         ]]>
      </xs:documentation>
    </xs:annotation>
    <xs:simpleContent>
      <xs:extension base="xs:token">
        <xs:attribute use="required" name="name" type="xs:token">
          <xs:annotation>
            <xs:documentation>
              <![CDATA[
                Specifies the name of the config-property
               ]]>
            </xs:documentation>
          </xs:annotation>
        </xs:attribute>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>

  <xs:complexType name="resource-adapterType">
    <xs:sequence>
      <xs:element name="archive" type="xs:token" minOccurs="1" maxOccurs="1">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Specifies the resource adapter archive to be activated
              E.g. <archive>myra.rar</archive>
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
                        <xs:element    name="bean-validation-groups"    type="bean-validation-
groupsType" minOccurs="0" maxOccurs="1">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Specifies bean validation group that should be used
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="bootstrap-context" type="xs:token" minOccurs="0" maxOccurs="1">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Specifies the unique name of the bootstrap context that should be used
             ]]>
          </xs:documentation>
```

```
        </xs:annotation>
      </xs:element>
                                    <xs:element      name="config-property"      type="config-
propertyType" minOccurs="0" maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
               The config-property specifies resource adapter configuration properties.
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="transaction-support" type="transaction-supportType" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Specifies the transaction support level of the resource adapter
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
                           <xs:element     name="connection-definitions"     type="connection-
definitionsType" minOccurs="0" maxOccurs="1">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Specifies the connection definitions
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="admin-objects" type="admin-objectsType" minOccurs="0" maxOccurs="1">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Specifies the administration objects
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>

  <xs:simpleType name="transaction-supportType">
    <xs:annotation>
      <xs:documentation>
        <![CDATA[
          Define the type of transaction supported by this resource adapter.
          Valid values are: NoTransaction, LocalTransaction, XATransaction
         ]]>
      </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:token">
      <xs:enumeration value="NoTransaction" />
      <xs:enumeration value="LocalTransaction" />
      <xs:enumeration value="XATransaction" />
    </xs:restriction>
  </xs:simpleType>
```

```
  <xs:attributeGroup name="common-attribute">
    <xs:attribute name="class-name" type="xs:token" use="optional">
      <xs:annotation>
        <xs:documentation>
          <![CDATA[
            Specifies the the fully qualified class name of a managed connection factory
            or admin object
           ]]>
        </xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="jndi-name" type="xs:token" use="required">
      <xs:annotation>
        <xs:documentation>
          <![CDATA[
            Specifies the JNDI name
           ]]>
        </xs:documentation>
      </xs:annotation>
    </xs:attribute>

  <xs:attribute name="enabled" type="xs:boolean" default="true" form="unqualified" use="optional">
      <xs:annotation>
        <xs:documentation>
          <![CDATA[
            Should the object in question be activated
           ]]>
        </xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute default="true" name="use-java-context" type="xs:boolean">
      <xs:annotation>
        <xs:documentation>
          <![CDATA[
            Specifies if a java:/ JNDI context should be used
           ]]>
        </xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="pool-name" type="xs:token" use="optional">
      <xs:annotation>
        <xs:documentation>
          <![CDATA[
            Specifies the pool name for the object
           ]]>
        </xs:documentation>
      </xs:annotation>
    </xs:attribute>
  </xs:attributeGroup>

  <xs:complexType name="admin-objectType">
    <xs:sequence>
                                  <xs:element      name="config-property"      type="config-
propertyType" minOccurs="0" maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              The config-property specifies administration object configuration properties.
             ]]>
```

```
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
    <xs:attributeGroup ref="common-attribute"></xs:attributeGroup>
  </xs:complexType>

  <xs:complexType name="timeoutType">
    <xs:sequence>
      <xs:element name="blocking-timeout-millis" type="xs:nonNegativeInteger" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
                The blocking-timeout-millis element indicates 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 30000 (30 seconds).
              ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="idle-timeout-minutes" type="xs:nonNegativeInteger" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              The idle-timeout-minutes elements indicates 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.
              ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
                                                          <xs:element          name="allocation-
retry" type="xs:nonNegativeInteger" minOccurs="0" maxOccurs="1">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              The allocation retry element indicates the number of times that allocating
              a connection should be tried before throwing an exception. The default is
              0.
              ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
                                                 <xs:element        name="allocation-retry-wait-
millis" type="xs:nonNegativeInteger" minOccurs="0" maxOccurs="1">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              The allocation retry wait millis element indicates the time in milliseconds
              to wait between retrying to allocate a connection. The default is 5000 (5
              seconds).
              ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
```

```
                                                    <xs:element          name="xa-resource-
timeout" type="xs:nonNegativeInteger" minOccurs="0" maxOccurs="1">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
                 Passed to XAResource.setTransactionTimeout(). Default is zero which does not
 invoke the setter.
              Specified in seconds - e.g. 5 minutes
              <xa-resource-timeout>300</xa-resource-timeout>
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="validationType">
    <xs:sequence>
      <xs:element name="background-validation" type="xs:boolean" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              An element to specify that connections should be validated on a background
              thread versus being validated prior to use
              ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
     <xs:element name="background-validation-millis" type="xs:nonNegativeInteger" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
               The background-validation-millis element specifies the amount of
               time, in millis, that background validation will run.
              ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="use-fast-fail" type="xs:boolean" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
                Whether fail a connection allocation on the first connection if it
                is invalid (true) or keep trying until the pool is exhausted of all potential
                connections (false) default false. e.g. <use-fast-fail>true</use-fast-fail>
              ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>

  <xs:element name="resource-adapters" type="resource-adaptersType">
    <xs:annotation>
      <xs:documentation>
        <![CDATA[
          Specifies activation of resource adapters
         ]]>
      </xs:documentation>
```

```
    </xs:annotation>
  </xs:element>

  <xs:complexType name="resource-adaptersType">
    <xs:sequence>
                                  <xs:element      name="resource-adapter"      type="resource-
adapterType" minOccurs="1" maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Specifies activation of a resource adapter
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="connection-definitionsType">
    <xs:sequence>
                            <xs:element     name="connection-definition"     type="connection-
defintionType" minOccurs="1" maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Specifies a connection definition
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="connection-defintionType">
    <xs:sequence>
                                    <xs:element      name="config-property"      type="config-
propertyType" minOccurs="0" maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
             The config-property specifies managed connection factory configuration properties.
            ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:choice>
        <xs:element name="pool" type="poolType" minOccurs="0" maxOccurs="1">
          <xs:annotation>
            <xs:documentation>
                <![CDATA[
                  Specifies pooling settings
                 ]]>
            </xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="xa-pool" type="xa-poolType" minOccurs="0" maxOccurs="1">
          <xs:annotation>
            <xs:documentation>
                <![CDATA[
```

```
                Specifies xa-pooling settings
               ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:choice>
    <xs:element name="security" type="securityType" minOccurs="0" maxOccurs="1">
      <xs:annotation>
        <xs:documentation>
          <![CDATA[
            Specifies security settings
           ]]>
        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="timeout" type="timeoutType" minOccurs="0" maxOccurs="1">
      <xs:annotation>
        <xs:documentation>
          <![CDATA[
            Specifies timeout settings
           ]]>
        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="validation" type="validationType" minOccurs="0" maxOccurs="1">
      <xs:annotation>
        <xs:documentation>
          <![CDATA[
            Specifies validation settings
           ]]>
        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="recovery" type="recoverType" minOccurs="0" maxOccurs="1"></xs:element>
  </xs:sequence>
  <xs:attribute name="use-ccm" type="xs:boolean" default="true" use="optional">
    <xs:annotation>
      <xs:documentation>
        <![CDATA[
          Enable cached connection manager
         ]]>
      </xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attributeGroup ref="common-attribute"></xs:attributeGroup>
</xs:complexType>

<xs:complexType name="poolType">
  <xs:sequence>
    <xs:element name="min-pool-size" type="xs:nonNegativeInteger" minOccurs="0">
      <xs:annotation>
        <xs:documentation>
          <![CDATA[
            The min-pool-size element indicates the minimum number of connections
            a pool should hold. These are not created until a Subject is known from a
            request for a connection. This default to 0. Ex: <min-pool-size>1</min-pool-size>
           ]]>
        </xs:documentation>
      </xs:annotation>
```

```
      </xs:element>
      <xs:element name="max-pool-size" type="xs:nonNegativeInteger" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              The max-pool-size element indicates the maximum number of connections
            for a pool. No more than max-pool-size connections will be created in each sub-pool.
              This defaults to 20.
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="prefill" type="xs:boolean" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Whether to attempt to prefill the connection pool. Default is false.
              e.g. <prefill>false</prefill>.
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="use-strict-min" type="xs:boolean" minOccurs="0" maxOccurs="1">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Define if the min-pool-size should be considered strict.
              Default false
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="flush-strategy" type="xs:token" minOccurs="0" maxOccurs="1">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Specifies how the pool should be flush in case of an error.
              Valid values are: FailingConnectionOnly (default), IdleConnections, EntirePool
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="xa-poolType">
    <xs:complexContent>
      <xs:extension base="poolType">
        <xs:sequence>
          <xs:element name="is-same-rm-override" type="xs:boolean" minOccurs="0">
            <xs:annotation>
              <xs:documentation>
                <![CDATA[
                  The is-same-rm-override element allows one to unconditionally
                  set whether the javax.transaction.xa.XAResource.isSameRM(XAResource) returns
                  true or false. Ex: <is-same-rm-override>true</is-same-rm-override>
                 ]]>
              </xs:documentation>
            </xs:annotation>
```

```
          </xs:element>
          <xs:element name="interleaving" type="boolean-presenceType" minOccurs="0">
            <xs:annotation>
              <xs:documentation>
                <![CDATA[
                  An element to enable interleaving for XA connection factories
                  Ex: <interleaving/>
                ]]>
              </xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="no-tx-separate-pools" type="boolean-presenceType" minOccurs="0">
            <xs:annotation>
              <xs:documentation>
                <![CDATA[
                     Oracle does not like XA connections getting used both inside and outside
a JTA transaction.
                       To workaround the problem you can create separate sub-pools for the
different contexts
                   using <no-tx-separate-pools/>
                   Ex: <no-tx-separate-pools/>
                 ]]>
              </xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="pad-xid" type="xs:boolean" default="false" minOccurs="0">
            <xs:annotation>
              <xs:documentation>
                <![CDATA[
                   Should the Xid be padded
                   Ex: <pad-xid>true</pad-xid>
                 ]]>
              </xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="wrap-xa-resource" type="xs:boolean" default="true" minOccurs="0">
            <xs:annotation>
              <xs:documentation>
                <![CDATA[
                 Should the XAResource instances be wrapped in a org.jboss.tm.XAResourceWrapper
                   instance
                   Ex: <wrap-xa-resource>true</wrap-xa-resource>
                 ]]>
              </xs:documentation>
            </xs:annotation>
          </xs:element>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="securityType">
    <xs:sequence>
      <xs:choice>
        <xs:element name="application" type="boolean-presenceType" minOccurs="0" maxOccurs="1">
          <xs:annotation>
            <xs:documentation>
              <![CDATA[
                Indicates that app supplied parameters (such as from getConnection(user, pw))
```

```
                are used to distinguish connections in the pool.
                Ex:
                <application/>
              ]]>
            </xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="security-domain" type="xs:token" minOccurs="0" maxOccurs="1">
          <xs:annotation>
            <xs:documentation>
              <![CDATA[
                  Indicates Subject (from security domain) are used 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.
                Ex:
                <security-domain>HsqlDbRealm</security-domain>
              ]]>
            </xs:documentation>
          </xs:annotation>
        </xs:element>
                                                  <xs:element      name="security-domain-and-
application" type="xs:token" minOccurs="0" maxOccurs="1">
          <xs:annotation>
            <xs:documentation>
              <![CDATA[
                Indicates that either app supplied parameters (such as from
                getConnection(user, pw)) or Subject (from security domain) are used 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.

                Ex:
                <security-domain-and-application>HsqlDbRealm</security-domain-and-application>
              ]]>
            </xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:choice>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="admin-objectsType">
    <xs:sequence>
                                       <xs:element      name="admin-object"      type="admin-
objectType" minOccurs="1" maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Specifies the setup for an admin object
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
```

```
  <xs:complexType name="bean-validation-groupsType">
    <xs:sequence>
                                                    <xs:element         name="bean-validation-
group" type="xs:token" minOccurs="1" maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[[
              Specifies the fully qualified class name for a bean validation group that
              should be used for validation
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="recoverType">
    <xs:sequence>
      <xs:element name="recover-credential" type="credentialType" minOccurs="0" maxOccurs="1">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[[
              Specifies the security options used when creating a connection during recovery.
                 Note: if this credential are not specified the security credential are used
 for recover too
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="recover-plugin" type="extensionType" minOccurs="0" maxOccurs="1">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[[
              Specifies the extension plugin used in spi (core.spi.xa)
                  which can be implemented by various plugins to provide better feedback to
 the XA recovery system.
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="no-recovery" type="xs:boolean" default="false" use="optional">
      <xs:annotation>
        <xs:documentation>
          <![CDATA[[
            Specify if the xa-datasource should be excluded from recovery.
            Default false.
           ]]>
        </xs:documentation>
      </xs:annotation>
    </xs:attribute>
  </xs:complexType>
  <xs:complexType name="extensionType">
    <xs:sequence>
      <xs:element name="config-property" type="config-propertyType"></xs:element>
    </xs:sequence>
    <xs:attribute name="class-name" type="xs:token" use="required"></xs:attribute>
  </xs:complexType>
  <xs:complexType name="credentialType">
```

```
    <xs:sequence>
      <xs:element name="user-name" type="xs:token" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
              <![CDATA[
                Specify the username used when creating a new connection.
                Ex: <user-name>sa</user-name>
               ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="password" type="xs:token" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
              <![CDATA[
                Specify the password used when creating a new connection.
                Ex: <password>sa-pass</password>
               ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="security-domain" type="xs:token" minOccurs="0" maxOccurs="1">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
                 Indicates Subject (from security domain) are used 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.
              Ex:
              <security-domain>HsqlDbRealm</security-domain>
            ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
   </xs:complexType>

</xs:schema>
```

A.6. Datasources 1.0

```
      <?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
           elementFormDefault="qualified"
           targetNamespace="http://www.jboss.org/ironjacamar/schema"
           xmlns="http://www.jboss.org/ironjacamar/schema">

  <xs:element name="datasources" type="datasourcesType">
    <xs:annotation>
      <xs:documentation>
```

```
        <![CDATA[
          The datasources element is the root of the JDBC datasource configuration
         ]]>
      </xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:complexType name="datasourcesType">
    <xs:sequence>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element name="datasource" type="datasourceType">
          <xs:annotation>
            <xs:documentation>
              <![CDATA[
                Specifies a non-XA datasource, using local transactions
               ]]>
            </xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="xa-datasource" type="xa-datasourceType">
          <xs:annotation>
            <xs:documentation>
              <![CDATA[
                Specifies a XA datasource
                ]]>
            </xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:choice>
      <xs:element name="drivers" type="driversType" maxOccurs="1" minOccurs="0"></xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="datasourceType" mixed="false">
    <xs:sequence>
      <xs:element name="connection-url" type="xs:token">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
          The JDBC driver connection URL Ex: <connection-url>jdbc:hsqldb:hsql://localhost:1701</
connection-url>
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="driver-class" type="xs:token" maxOccurs="1" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
                        The fully qualifed name of the JDBC driver class Ex: <driver-
class>org.hsqldb.jdbcDriver</driver-class>
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="datasource-class" type="xs:token" maxOccurs="1" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
                    The fully qualifed name of the JDBC datasource class Ex: <datasource-
class>org.h2.jdbcx.JdbcDataSource</datasource-class>
```

```
            ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="driver" type="xs:token" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              An unique reference to the classloader module which contains the JDBC driver
              The accepted format is driverName#majorVersion.minorVersion
            ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
                              <xs:element     name="connection-property"     type="connection-
propertyType" minOccurs="0" maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              The connection-property element allows you to pass in arbitrary connection
              properties to the 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. Ex:
              <connection-property name="char.encoding">UTF-8</connection-property>
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="new-connection-sql" type="xs:string" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Specify an SQL statement to execute whenever a connection is added
              to the connection pool.
              ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="transaction-isolation" type="transaction-isolationType" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Set java.sql.Connection transaction isolation level to use. The constants
              defined by transaction-isolation-values are the possible transaction isolation
              levels and include: TRANSACTION_READ_UNCOMMITTED TRANSACTION_READ_COMMITTED
              TRANSACTION_REPEATABLE_READ TRANSACTION_SERIALIZABLE TRANSACTION_NONE
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="url-delimiter" type="xs:token" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Specifies the delimeter for URLs in connection-url for HA datasources
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
```

```
      <xs:element name="url-selector-strategy-class-name" type="xs:token" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              A class that implements org.jboss.jca.adapters.jdbc.URLSelectorStrategy
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="pool" type="poolType" minOccurs="0" maxOccurs="1">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Specifies the pooling settings
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="security" type="dsSecurityType" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Specifies the security settings
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="validation" type="validationType" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Specifies the validation settings
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="timeout" type="timeoutType" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Specifies the time out settings
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="statement" type="statementType" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Specifies the statement settings
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="jta" type="xs:boolean" default="true" use="optional">
      <xs:annotation>
        <xs:documentation>
          <![CDATA[
```

```
            Enable JTA integration
           ]]>
        </xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attributeGroup ref="common-datasourceAttributes" />
  </xs:complexType>
  <xs:complexType name="xa-datasourceType">
    <xs:sequence>
                        <xs:element    name="xa-datasource-property"    type="xa-datasource-
propertyType" minOccurs="1" maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Specifies a property 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. Ex:
              <xa-datasource-property name="IfxWAITTIME">10</xa-datasource-property>
                 <xa-datasource-property name="IfxIFXHOST">myhost.mydomain.com</xa-datasource-
property>
              <xa-datasource-property name="PortNumber">1557</xa-datasource-property>
              <xa-datasource-property name="DatabaseName">mydb</xa-datasource-property>
              <xa-datasource-property name="ServerName">myserver</xa-datasource-property>
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="xa-datasource-class" type="xs:token" maxOccurs="1" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              The fully qualifed name of the javax.sql.XADataSource implementation
                 class. Ex: <xa-datasource-class>oracle.jdbc.xa.client.OracleXADataSource</xa-
datasource-class>
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="driver" type="xs:token" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              An unique reference to the classloader module which contains the JDBC driver
              The accepted format is driverName#majorVersion.minorVersion
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="url-delimiter" type="xs:token" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
               Specifies the delimeter for URLs in the connection url for HA datasources
              ]]>
          </xs:documentation>
```

```
      </xs:annotation>
    </xs:element>
    <xs:element name="url-selector-strategy-class-name" type="xs:token" minOccurs="0">
      <xs:annotation>
        <xs:documentation>
          <![CDATA[
             A class that implements org.jboss.jca.adapters.jdbc.URLSelectorStrategy
           ]]>
        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="new-connection-sql" type="xs:string" minOccurs="0">
      <xs:annotation>
        <xs:documentation>
          <![CDATA[
             Specifies an SQL statement to execute whenever a connection is added
             to the connection pool.
           ]]>
        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="transaction-isolation" type="transaction-isolationType" minOccurs="0">
      <xs:annotation>
        <xs:documentation>
          <![CDATA[
            Set java.sql.Connection transaction isolation level to use. The constants
            defined by transaction-isolation-values are the possible transaction isolation
            levels and include: TRANSACTION_READ_UNCOMMITTED TRANSACTION_READ_COMMITTED
            TRANSACTION_REPEATABLE_READ TRANSACTION_SERIALIZABLE TRANSACTION_NONE
           ]]>
        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="xa-pool" type="xa-poolType" minOccurs="0" maxOccurs="1">
      <xs:annotation>
        <xs:documentation>
          <![CDATA[
            Specifies the pooling settings
           ]]>
        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="security" type="dsSecurityType" minOccurs="0">
      <xs:annotation>
        <xs:documentation>
          <![CDATA[
            Specifies the security settings
           ]]>
        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="validation" type="validationType" minOccurs="0">
      <xs:annotation>
        <xs:documentation>
          <![CDATA[
            Specifies the validation settings
           ]]>
        </xs:documentation>
      </xs:annotation>
```

```
      </xs:element>
      <xs:element name="timeout" type="timeoutType" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Specifies the time out settings
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="statement" type="statementType" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Specifies the statement settings
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="recovery" type="recoverType" minOccurs="0" maxOccurs="1"></xs:element>
    </xs:sequence>
    <xs:attributeGroup ref="common-datasourceAttributes" />
  </xs:complexType>
  <xs:complexType name="boolean-presenceType" />
  <xs:attributeGroup name="common-datasourceAttributes">
    <xs:attribute name="jndi-name" type="xs:token" use="required">
      <xs:annotation>
        <xs:documentation>
          <![CDATA[
            Specifies the JNDI name for the datasource
           ]]>
        </xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="pool-name" type="xs:token" use="required">
      <xs:annotation>
        <xs:documentation>
          <![CDATA[
            Specifies the pool name for the datasource used for management
           ]]>
        </xs:documentation>
      </xs:annotation>
    </xs:attribute>

    <xs:attribute name="enabled" type="xs:boolean" default="true" form="unqualified" use="optional">
      <xs:annotation>
        <xs:documentation>
          <![CDATA[
            Specifies if the datasource should be enabled
           ]]>
        </xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute default="true" name="use-java-context" type="xs:boolean">
      <xs:annotation>
        <xs:documentation>
          <![CDATA[
            Setting this to false will bind the DataSource into global JNDI
            Ex: use-java-context="true"
```

```
            ]]>
        </xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute default="false" name="spy" type="xs:boolean">
      <xs:annotation>
        <xs:documentation>
          <![CDATA[
         Enable spy functionality on the JDBC layer - e.g. log all JDBC traffic to the datasource.
            Remember to enable the logging category (org.jboss.jdbc) too.
            Ex: spy="true"
           ]]>
        </xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute default="true" name="use-ccm" type="xs:boolean">
      <xs:annotation>
        <xs:documentation>
          <![CDATA[
            Enable the use of a cached connection manager
            Ex: use-ccm="true"
           ]]>
        </xs:documentation>
      </xs:annotation>
    </xs:attribute>
  </xs:attributeGroup>
  <xs:simpleType name="transaction-isolationType">
    <xs:annotation>
      <xs:documentation>
        <![CDATA[
           Define constants used as the possible transaction isolation levels in transaction-
isolation
                    type.  Include:  TRANSACTION_READ_UNCOMMITTED,  TRANSACTION_READ_COMMITTED,
 TRANSACTION_REPEATABLE_READ,
          TRANSACTION_SERIALIZABLE, TRANSACTION_NONE
         ]]>
      </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:token">
      <xs:enumeration value="TRANSACTION_READ_UNCOMMITTED" />
      <xs:enumeration value="TRANSACTION_READ_COMMITTED" />
      <xs:enumeration value="TRANSACTION_REPEATABLE_READ" />
      <xs:enumeration value="TRANSACTION_SERIALIZABLE" />
      <xs:enumeration value="TRANSACTION_NONE" />
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="xa-datasource-propertyType" mixed="true">
    <xs:attribute name="name" use="required" type="xs:token" />
  </xs:complexType>
  <xs:complexType name="connection-propertyType" mixed="true">
    <xs:attribute name="name" use="required" type="xs:token" />
  </xs:complexType>
  <xs:complexType name="validationType">
    <xs:sequence>
      <xs:element name="valid-connection-checker" type="extensionType" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              An org.jboss.jca.adapters.jdbc.ValidConnectionChecker that provides
```

```
            a SQLException isValidConnection(Connection e) method to validate is a connection
             is valid. An exception means the connection is destroyed. This overrides
             the check-valid-connection-sql when present. Ex:
                                                          <valid-connection-checker    class-
name="org.jboss.jca.adapters.jdbc.vendor.OracleValidConnectionChecker"/>
            ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>

      <xs:element name="check-valid-connection-sql" type="xs:string" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Specify an SQL statement to check validity of a pool connection. This
              may be called when managed connection is taken from pool for use.
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="validate-on-match" type="xs:boolean" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              The validate-on-match element indicates whether or not connection
              level validation should be done when a connection factory attempts to match
              a managed connection for a given set. This is typically exclusive to the
              use of background validation
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="background-validation" type="xs:boolean" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              An element to specify that connections should be validated on a background
              thread versus being validated prior to use
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    <xs:element name="background-validation-millis" type="xs:nonNegativeInteger" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              The background-validation-millis element specifies the amount of
              time, in millis, that background validation will run.
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="use-fast-fail" type="xs:boolean" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Whether fail a connection allocation on the first connection if it
              is invalid (true) or keep trying until the pool is exhausted of all potential
              connections (false) default false. e.g. <use-fast-fail>true</use-fast-fail>
```

```
            ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element minOccurs="0" name="stale-connection-checker" type="extensionType">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              An org.jboss.jca.adapters.jdbc.StaleConnectionChecker that provides
              a boolean isStaleConnection(SQLException e) method which if it it returns
                                                  true   will   wrap   the   exception   in   an
 org.jboss.jca.adapters.jdbc.StaleConnectionException
              which is a subclass of SQLException. Ex:
                                                                <stale-connection-checker    class-
name="org.jboss.jca.adapters.jdbc.vendor.OracleStaleConnectionChecker"/>
            ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="exception-sorter" type="extensionType" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              An org.jboss.jca.adapters.jdbc.ExceptionSorter that provides a
              boolean isExceptionFatal(SQLException e) method to validate is an exception
              should be broadcast to all javax.resource.spi.ConnectionEventListener as
              a connectionErrorOccurred message. Ex:
                                                                    <exception-sorter    class-
name="org.jboss.jca.adapters.jdbc.vendor.OracleExceptionSorter"/>
            ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="timeoutType">
    <xs:sequence>
      <xs:element name="blocking-timeout-millis" type="xs:nonNegativeInteger" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              The blocking-timeout-millis element indicates 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 30000 (30 seconds).
            ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="idle-timeout-minutes" type="xs:nonNegativeInteger" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              The idle-timeout-minutes elements indicates 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.
            ]]>
```

```
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="set-tx-query-timeout" type="boolean-presenceType" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Whether to set the query timeout based on the time remaining until
              transaction timeout, any configured query timeout will be used if there is
              no transaction. The default is false. e.g. <set-tx-query-timeout/>
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="query-timeout" type="xs:nonNegativeInteger" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Any configured query timeout in seconds The default is no timeout
              e.g. 5 minutes <query-timeout>300</query-timeout>
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="use-try-lock" type="xs:nonNegativeInteger" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Any configured timeout for internal locks on the resource adapter
                 objects in seconds The default is a 60 second timeout e.g. 5 minutes <use-
try-lock>300</use-try-lock>
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="allocation-retry" type="xs:nonNegativeInteger" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              The allocation retry element indicates the number of times that allocating
              a connection should be tried before throwing an exception. The default is 0.
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    <xs:element name="allocation-retry-wait-millis" type="xs:nonNegativeInteger" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              The allocation retry wait millis element indicates the time in milliseconds
            to wait between retrying to allocate a connection. The default is 5000 (5 seconds).
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="xa-resource-timeout" type="xs:token" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
```

```
              Passed to XAResource.setTransactionTimeout() Default is zero which
              does not invoke the setter. In seconds e.g. 5 minutes <xa-resource-timeout>300</
xa-resource-timeout>
            ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="track-statementsType">
    <xs:restriction base="xs:token">
      <xs:enumeration value="true" />
      <xs:enumeration value="false" />
      <xs:enumeration value="nowarn" />
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="statementType">
    <xs:sequence>
      <xs:element name="track-statements" type="track-statementsType" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Whether to check for unclosed statements when a connection is returned
              to the pool and result sets are closed when a statement is closed/return
             to the prepared statement cache. valid values are: false - do not track statements
              and results true - track statements and result sets and warn when they are
              not closed nowarn - track statements but do no warn about them being unclosed
              (the default) e.g. <track-statements>nowarn</track-statements>
            ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
                                            <xs:element       name="prepared-statement-cache-
size" type="xs:nonNegativeInteger" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              The number of prepared statements per connection in an LRU cache
            ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="share-prepared-statements" type="boolean-presenceType" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Whether to share prepare statements, i.e. whether asking for same
              statement twice without closing uses the same underlying prepared statement.
              The default is false. e.g. <share-prepared-statements/>
            ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="poolType">
    <xs:sequence>
      <xs:element name="min-pool-size" type="xs:nonNegativeInteger" minOccurs="0">
        <xs:annotation>
```

```
          <xs:documentation>
            <![CDATA[
              The min-pool-size element indicates the minimum number of connections
              a pool should hold. These are not created until a Subject is known from a
              request for a connection. This default to 0. Ex: <min-pool-size>1</min-pool-size>
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="max-pool-size" type="xs:nonNegativeInteger" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              The max-pool-size element indicates the maximum number of connections
              for a pool. No more connections will be created in each sub-pool.
              This defaults to 20.
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="prefill" type="xs:boolean" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Whether to attempt to prefill the connection pool. Empty element denotes
              a true value. e.g. <prefill>true</prefill>.
              Default is false
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="use-strict-min" type="xs:boolean" minOccurs="0" maxOccurs="1">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Define if the min-pool-size should be considered a strictly.
              Default false
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="flush-strategy" type="xs:token" minOccurs="0" maxOccurs="1">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Specifies how the pool should be flush in case of an error.
              Valid values are: FailingConnectionOnly (default), IdleConnections, EntirePool
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="xa-poolType">
    <xs:complexContent>
      <xs:extension base="poolType">
        <xs:sequence>
          <xs:element name="is-same-rm-override" type="xs:boolean" minOccurs="0">
            <xs:annotation>
```

```
            <xs:documentation>
              <![CDATA[
                The is-same-rm-override element allows one to unconditionally
                set whether the javax.transaction.xa.XAResource.isSameRM(XAResource) returns
                true or false. Ex: <is-same-rm-override>true</is-same-rm-override>
               ]]>
            </xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="interleaving" type="boolean-presenceType" minOccurs="0">
          <xs:annotation>
            <xs:documentation>
              <![CDATA[
                An element to enable interleaving for XA connection factories
                Ex: <interleaving/>
               ]]>
            </xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="no-tx-separate-pools" type="boolean-presenceType" minOccurs="0">
          <xs:annotation>
            <xs:documentation>
              <![CDATA[
                   Oracle does not like XA connections getting used both inside and outside
a JTA transaction.
                     To workaround the problem you can create separate sub-pools for the
different contexts
                using <no-tx-separate-pools/>
                Ex: <no-tx-separate-pools/>
               ]]>
            </xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="pad-xid" type="xs:boolean" default="false" minOccurs="0">
          <xs:annotation>
            <xs:documentation>
              <![CDATA[
                 Should the Xid be padded
                 Ex: <pad-xid>true</pad-xid>
               ]]>
            </xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="wrap-xa-resource" type="xs:boolean" default="true" minOccurs="0">
          <xs:annotation>
            <xs:documentation>
              <![CDATA[
               Should the XAResource instances be wrapped in a org.jboss.tm.XAResourceWrapper
                 instance
                 Ex: <wrap-xa-resource>true</wrap-xa-resource>
               ]]>
            </xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
<xs:complexType name="dsSecurityType">
```

```
    <xs:sequence>
      <xs:element name="user-name" type="xs:token" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
                Specify the username used when creating a new connection.
                Ex: <user-name>sa</user-name>
               ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="password" type="xs:token" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
                Specify the password used when creating a new connection.
                Ex: <password>sa-pass</password>
               ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="security-domain" type="xs:token" minOccurs="0" maxOccurs="1">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
                 Indicates Subject (from security domain) are used 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.
              Ex:
              <security-domain>HsqlDbRealm</security-domain>
            ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
         <xs:element name="reauth-plugin" type="extensionType" minOccurs="0" maxOccurs="1"></
xs:element>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="extensionType">
    <xs:sequence>
                                    <xs:element      name="config-property"      type="config-
propertyType" minOccurs="0" maxOccurs="unbounded"></xs:element>
    </xs:sequence>
    <xs:attribute name="class-name" type="xs:token" use="required"></xs:attribute>
  </xs:complexType>

  <xs:complexType name="config-propertyType" mixed="true">
    <xs:annotation>
      <xs:documentation>
        <![CDATA[
          Specifies a Java bean property value
         ]]>
      </xs:documentation>
    </xs:annotation>
    <xs:simpleContent>
```

```
      <xs:extension base="xs:token">
        <xs:attribute use="required" name="name" type="xs:token">
          <xs:annotation>
            <xs:documentation>
              <![CDATA[
                Specifies the name of the config-property
               ]]>
            </xs:documentation>
          </xs:annotation>
        </xs:attribute>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
  <xs:complexType name="recoverType">
    <xs:sequence>
      <xs:element name="recover-credential" type="dsSecurityType" minOccurs="0" maxOccurs="1">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Specifies the security options used when creating a connection during recovery.
                 Note: if this credential are not specified the security credential are used
 for recover too
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="recover-plugin" type="extensionType" minOccurs="0" maxOccurs="1">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Specifies the extension plugin used in spi (core.spi.xa)
                  which can be implemented by various plugins to provide better feedback to
 the XA recovery system.
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="no-recovery" type="xs:boolean" default="false" use="optional">
      <xs:annotation>
        <xs:documentation>
          <![CDATA[
            Specify if the xa-datasource should be excluded from recovery.
            Default false.
           ]]>
        </xs:documentation>
      </xs:annotation>
    </xs:attribute>
  </xs:complexType>

  <xs:complexType name="driverType">
    <xs:sequence>
      <xs:element name="driver-class" type="xs:token" maxOccurs="1" minOccurs="0">
      <xs:annotation>
          <xs:documentation>
            <![CDATA[
                        The fully qualifed name of the JDBC driver class Ex: <driver-
class>org.hsqldb.jdbcDriver</driver-class>
             ]]>
```

```
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="datasource-class" type="xs:token" maxOccurs="1" minOccurs="0">
      <xs:annotation>
          <xs:documentation>
           <![CDATA[
              The fully qualifed name of the javax.sql.DataSource implementation
              class.
             ]]>
          </xs:documentation>
        </xs:annotation></xs:element>
      <xs:element name="xa-datasource-class" type="xs:token" maxOccurs="1" minOccurs="0">
      <xs:annotation>
          <xs:documentation>
           <![CDATA[
              The fully qualifed name of the javax.sql.XADataSource implementation
                 class. Ex: <xa-datasource-class>oracle.jdbc.xa.client.OracleXADataSource</xa-
datasource-class>
             ]]>
          </xs:documentation>
        </xs:annotation></xs:element>
    </xs:sequence>
    <xs:attribute name="name" type="xs:token" use="required">
      <xs:annotation>
        <xs:documentation>
          <![CDATA[
            Specifies the symbolic name of this driver used to reference this driver
           ]]>
        </xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="module" type="xs:token" use="optional">
      <xs:annotation>
        <xs:documentation>
          <![CDATA[
            Specifies the name of AS7 module providing this driver.
            Thios tag is not used in IronJacamar standalone container.
           ]]>
        </xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="major-version" type="xs:int" use="optional">
      <xs:annotation>
        <xs:documentation>
          <![CDATA[
              Specifies the major version of this driver. If the major and minor version is
 obmitted the fist availabe
            Driver in module will be used.
           ]]>
        </xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="minor-verion" type="xs:int" use="optional">
    <xs:annotation>
        <xs:documentation>
          <![CDATA[
              Specifies the minor version of this driver. If the major and minor version is
 obmitted the fist availabe
```

```
            Driver in module will be used.
           ]]>
        </xs:documentation>
      </xs:annotation>
    </xs:attribute>
  </xs:complexType>

  <xs:complexType name="driversType">
    <xs:sequence>
         <xs:element  name="driver"  type="driverType"  maxOccurs="unbounded"  minOccurs="1"></
xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

A.7. Datasources 1.1

```
      <?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
           elementFormDefault="qualified"
           targetNamespace="http://www.jboss.org/ironjacamar/schema"
           xmlns="http://www.jboss.org/ironjacamar/schema">

  <xs:element name="datasources" type="datasourcesType">
    <xs:annotation>
      <xs:documentation>
        <![CDATA[
          The datasources element is the root of the JDBC datasource configuration
         ]]>
      </xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:complexType name="datasourcesType">
    <xs:sequence>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element name="datasource" type="datasourceType">
          <xs:annotation>
            <xs:documentation>
              <![CDATA[
                Specifies a non-XA datasource, using local transactions
               ]]>
            </xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="xa-datasource" type="xa-datasourceType">
          <xs:annotation>
            <xs:documentation>
              <![CDATA[
                Specifies a XA datasource
                ]]>
            </xs:documentation>
          </xs:annotation>
        </xs:element>
```

```
      </xs:choice>
      <xs:element name="drivers" type="driversType" maxOccurs="1" minOccurs="0"></xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="datasourceType" mixed="false">
    <xs:sequence>
      <xs:element name="connection-url" type="xs:token">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
          The JDBC driver connection URL Ex: <connection-url>jdbc:hsqldb:hsql://localhost:1701</
connection-url>
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="driver-class" type="xs:token" maxOccurs="1" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
                        The fully qualifed name of the JDBC driver class Ex: <driver-
class>org.hsqldb.jdbcDriver</driver-class>
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="datasource-class" type="xs:token" maxOccurs="1" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
                    The fully qualifed name of the JDBC datasource class Ex: <datasource-
class>org.h2.jdbcx.JdbcDataSource</datasource-class>
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="driver" type="xs:token" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              An unique reference to the classloader module which contains the JDBC driver
              The accepted format is driverName#majorVersion.minorVersion
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
                              <xs:element     name="connection-property"     type="connection-
propertyType" minOccurs="0" maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              The connection-property element allows you to pass in arbitrary connection
              properties to the 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. Ex:
              <connection-property name="char.encoding">UTF-8</connection-property>
             ]]>
          </xs:documentation>
        </xs:annotation>
```

```
      </xs:element>
      <xs:element name="new-connection-sql" type="xs:string" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Specify an SQL statement to execute whenever a connection is added
              to the connection pool.
              ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="transaction-isolation" type="transaction-isolationType" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Set java.sql.Connection transaction isolation level to use. The constants
              defined by transaction-isolation-values are the possible transaction isolation
              levels and include: TRANSACTION_READ_UNCOMMITTED TRANSACTION_READ_COMMITTED
              TRANSACTION_REPEATABLE_READ TRANSACTION_SERIALIZABLE TRANSACTION_NONE
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="url-delimiter" type="xs:token" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Specifies the delimeter for URLs in connection-url for HA datasources
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="url-selector-strategy-class-name" type="xs:token" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              A class that implements org.jboss.jca.adapters.jdbc.URLSelectorStrategy
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="pool" type="poolType" minOccurs="0" maxOccurs="1">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Specifies the pooling settings
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="security" type="dsSecurityType" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Specifies the security settings
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
```

```
      <xs:element name="validation" type="validationType" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Specifies the validation settings
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="timeout" type="timeoutType" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Specifies the time out settings
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="statement" type="statementType" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Specifies the statement settings
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="jta" type="xs:boolean" default="true" use="optional">
      <xs:annotation>
        <xs:documentation>
          <![CDATA[
            Enable JTA integration
           ]]>
        </xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attributeGroup ref="common-datasourceAttributes" />
  </xs:complexType>
  <xs:complexType name="xa-datasourceType">
    <xs:sequence>
                          <xs:element     name="xa-datasource-property"     type="xa-datasource-
propertyType" minOccurs="1" maxOccurs="unbounded">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Specifies a property 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. Ex:
              <xa-datasource-property name="IfxWAITTIME">10</xa-datasource-property>
                 <xa-datasource-property name="IfxIFXHOST">myhost.mydomain.com</xa-datasource-
property>
              <xa-datasource-property name="PortNumber">1557</xa-datasource-property>
              <xa-datasource-property name="DatabaseName">mydb</xa-datasource-property>
              <xa-datasource-property name="ServerName">myserver</xa-datasource-property>
```

```
            ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="xa-datasource-class" type="xs:token" maxOccurs="1" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              The fully qualifed name of the javax.sql.XADataSource implementation
                 class. Ex: <xa-datasource-class>oracle.jdbc.xa.client.OracleXADataSource</xa-
datasource-class>
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="driver" type="xs:token" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              An unique reference to the classloader module which contains the JDBC driver
              The accepted format is driverName#majorVersion.minorVersion
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="url-delimiter" type="xs:token" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
               Specifies the delimeter for URLs in the connection url for HA datasources
              ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="url-selector-strategy-class-name" type="xs:token" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
               A class that implements org.jboss.jca.adapters.jdbc.URLSelectorStrategy
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="new-connection-sql" type="xs:string" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
               Specifies an SQL statement to execute whenever a connection is added
               to the connection pool.
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="transaction-isolation" type="transaction-isolationType" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Set java.sql.Connection transaction isolation level to use. The constants
              defined by transaction-isolation-values are the possible transaction isolation
```

```
            levels and include: TRANSACTION_READ_UNCOMMITTED TRANSACTION_READ_COMMITTED
            TRANSACTION_REPEATABLE_READ TRANSACTION_SERIALIZABLE TRANSACTION_NONE
          ]]>
        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="xa-pool" type="xa-poolType" minOccurs="0" maxOccurs="1">
      <xs:annotation>
        <xs:documentation>
          <![CDATA[
            Specifies the pooling settings
          ]]>
        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="security" type="dsSecurityType" minOccurs="0">
      <xs:annotation>
        <xs:documentation>
          <![CDATA[
            Specifies the security settings
          ]]>
        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="validation" type="validationType" minOccurs="0">
      <xs:annotation>
        <xs:documentation>
          <![CDATA[
            Specifies the validation settings
          ]]>
        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="timeout" type="timeoutType" minOccurs="0">
      <xs:annotation>
        <xs:documentation>
          <![CDATA[
            Specifies the time out settings
          ]]>
        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="statement" type="statementType" minOccurs="0">
      <xs:annotation>
        <xs:documentation>
          <![CDATA[
            Specifies the statement settings
          ]]>
        </xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="recovery" type="recoverType" minOccurs="0" maxOccurs="1"></xs:element>
  </xs:sequence>
  <xs:attributeGroup ref="common-datasourceAttributes" />
</xs:complexType>
<xs:complexType name="boolean-presenceType" />
<xs:attributeGroup name="common-datasourceAttributes">
  <xs:attribute name="jndi-name" type="xs:token" use="required">
    <xs:annotation>
```

```
        <xs:documentation>
          <![CDATA[
            Specifies the JNDI name for the datasource
          ]]>
        </xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="pool-name" type="xs:token" use="required">
      <xs:annotation>
        <xs:documentation>
          <![CDATA[
            Specifies the pool name for the datasource used for management
          ]]>
        </xs:documentation>
      </xs:annotation>
    </xs:attribute>

  <xs:attribute name="enabled" type="xs:boolean" default="true" form="unqualified" use="optional">
      <xs:annotation>
        <xs:documentation>
          <![CDATA[
            Specifies if the datasource should be enabled
          ]]>
        </xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute default="true" name="use-java-context" type="xs:boolean">
      <xs:annotation>
        <xs:documentation>
          <![CDATA[
            Setting this to false will bind the DataSource into global JNDI
            Ex: use-java-context="true"
          ]]>
        </xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute default="false" name="spy" type="xs:boolean">
      <xs:annotation>
        <xs:documentation>
          <![CDATA[
         Enable spy functionality on the JDBC layer - e.g. log all JDBC traffic to the datasource.
            Remember to enable the logging category (org.jboss.jdbc) too.
            Ex: spy="true"
          ]]>
        </xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute default="true" name="use-ccm" type="xs:boolean">
      <xs:annotation>
        <xs:documentation>
          <![CDATA[
            Enable the use of a cached connection manager
            Ex: use-ccm="true"
          ]]>
        </xs:documentation>
      </xs:annotation>
    </xs:attribute>
  </xs:attributeGroup>
  <xs:simpleType name="transaction-isolationType">
```

```
    <xs:annotation>
      <xs:documentation>
        <![CDATA[
           Define constants used as the possible transaction isolation levels in transaction-
isolation
                  type.  Include:  TRANSACTION_READ_UNCOMMITTED,  TRANSACTION_READ_COMMITTED,
 TRANSACTION_REPEATABLE_READ,
          TRANSACTION_SERIALIZABLE, TRANSACTION_NONE
         ]]>
      </xs:documentation>
    </xs:annotation>
    <xs:restriction base="xs:token">
      <xs:enumeration value="TRANSACTION_READ_UNCOMMITTED" />
      <xs:enumeration value="TRANSACTION_READ_COMMITTED" />
      <xs:enumeration value="TRANSACTION_REPEATABLE_READ" />
      <xs:enumeration value="TRANSACTION_SERIALIZABLE" />
      <xs:enumeration value="TRANSACTION_NONE" />
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="xa-datasource-propertyType" mixed="true">
    <xs:attribute name="name" use="required" type="xs:token" />
  </xs:complexType>
  <xs:complexType name="connection-propertyType" mixed="true">
    <xs:attribute name="name" use="required" type="xs:token" />
  </xs:complexType>
  <xs:complexType name="validationType">
    <xs:sequence>
      <xs:element name="valid-connection-checker" type="extensionType" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              An org.jboss.jca.adapters.jdbc.ValidConnectionChecker that provides
             a SQLException isValidConnection(Connection e) method to validate is a connection
              is valid. An exception means the connection is destroyed. This overrides
              the check-valid-connection-sql when present. Ex:
                                                                  <valid-connection-checker    class-
name="org.jboss.jca.adapters.jdbc.vendor.OracleValidConnectionChecker"/>
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>

      <xs:element name="check-valid-connection-sql" type="xs:string" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Specify an SQL statement to check validity of a pool connection. This
              may be called when managed connection is taken from pool for use.
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="validate-on-match" type="xs:boolean" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              The validate-on-match element indicates whether or not connection
              level validation should be done when a connection factory attempts to match
              a managed connection for a given set. This is typically exclusive to the
```

```
              use of background validation
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="background-validation" type="xs:boolean" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              An element to specify that connections should be validated on a background
              thread versus being validated prior to use
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    <xs:element name="background-validation-millis" type="xs:nonNegativeInteger" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              The background-validation-millis element specifies the amount of
              time, in millis, that background validation will run.
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="use-fast-fail" type="xs:boolean" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Whether fail a connection allocation on the first connection if it
              is invalid (true) or keep trying until the pool is exhausted of all potential
              connections (false) default false. e.g. <use-fast-fail>true</use-fast-fail>
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element minOccurs="0" name="stale-connection-checker" type="extensionType">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              An org.jboss.jca.adapters.jdbc.StaleConnectionChecker that provides
              a boolean isStaleConnection(SQLException e) method which if it it returns
                                                  true   will   wrap   the   exception   in   an
 org.jboss.jca.adapters.jdbc.StaleConnectionException
              which is a subclass of SQLException. Ex:
                                                                 <stale-connection-checker    class-
name="org.jboss.jca.adapters.jdbc.vendor.OracleStaleConnectionChecker"/>
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="exception-sorter" type="extensionType" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              An org.jboss.jca.adapters.jdbc.ExceptionSorter that provides a
              boolean isExceptionFatal(SQLException e) method to validate is an exception
              should be broadcast to all javax.resource.spi.ConnectionEventListener as
              a connectionErrorOccurred message. Ex:
```

```
                                                                    <exception-sorter     class-
name="org.jboss.jca.adapters.jdbc.vendor.OracleExceptionSorter"/>
            ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="timeoutType">
    <xs:sequence>
      <xs:element name="blocking-timeout-millis" type="xs:nonNegativeInteger" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              The blocking-timeout-millis element indicates 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 30000 (30 seconds).
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="idle-timeout-minutes" type="xs:nonNegativeInteger" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              The idle-timeout-minutes elements indicates 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.
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="set-tx-query-timeout" type="boolean-presenceType" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Whether to set the query timeout based on the time remaining until
              transaction timeout, any configured query timeout will be used if there is
              no transaction. The default is false. e.g. <set-tx-query-timeout/>
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="query-timeout" type="xs:nonNegativeInteger" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Any configured query timeout in seconds The default is no timeout
              e.g. 5 minutes <query-timeout>300</query-timeout>
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="use-try-lock" type="xs:nonNegativeInteger" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
```

```
            <![CDATA[
              Any configured timeout for internal locks on the resource adapter
                 objects in seconds The default is a 60 second timeout e.g. 5 minutes <use-
try-lock>300</use-try-lock>
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="allocation-retry" type="xs:nonNegativeInteger" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              The allocation retry element indicates the number of times that allocating
              a connection should be tried before throwing an exception. The default is 0.
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    <xs:element name="allocation-retry-wait-millis" type="xs:nonNegativeInteger" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              The allocation retry wait millis element indicates the time in milliseconds
             to wait between retrying to allocate a connection. The default is 5000 (5 seconds).
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="xa-resource-timeout" type="xs:token" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Passed to XAResource.setTransactionTimeout() Default is zero which
              does not invoke the setter. In seconds e.g. 5 minutes <xa-resource-timeout>300</
xa-resource-timeout>
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="track-statementsType">
    <xs:restriction base="xs:token">
      <xs:enumeration value="true" />
      <xs:enumeration value="false" />
      <xs:enumeration value="nowarn" />
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="statementType">
    <xs:sequence>
      <xs:element name="track-statements" type="track-statementsType" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Whether to check for unclosed statements when a connection is returned
              to the pool and result sets are closed when a statement is closed/return
             to the prepared statement cache. valid values are: false - do not track statements
              and results true - track statements and result sets and warn when they are
              not closed nowarn - track statements but do no warn about them being unclosed
```

```
              (the default) e.g. <track-statements>nowarn</track-statements>
            ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
                                              <xs:element       name="prepared-statement-cache-
size" type="xs:nonNegativeInteger" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              The number of prepared statements per connection in an LRU cache
            ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="share-prepared-statements" type="boolean-presenceType" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Whether to share prepare statements, i.e. whether asking for same
              statement twice without closing uses the same underlying prepared statement.
              The default is false. e.g. <share-prepared-statements/>
            ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="poolType">
    <xs:sequence>
      <xs:element name="min-pool-size" type="xs:nonNegativeInteger" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              The min-pool-size element indicates the minimum number of connections
              a pool should hold. These are not created until a Subject is known from a
              request for a connection. This default to 0. Ex: <min-pool-size>1</min-pool-size>
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="max-pool-size" type="xs:nonNegativeInteger" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              The max-pool-size element indicates the maximum number of connections
              for a pool. No more connections will be created in each sub-pool.
              This defaults to 20.
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="prefill" type="xs:boolean" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Whether to attempt to prefill the connection pool. Empty element denotes
              a true value. e.g. <prefill>true</prefill>.
              Default is false
```

```
            ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="use-strict-min" type="xs:boolean" minOccurs="0" maxOccurs="1">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Define if the min-pool-size should be considered a strictly.
              Default false
            ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="flush-strategy" type="xs:token" minOccurs="0" maxOccurs="1">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Specifies how the pool should be flush in case of an error.
              Valid values are: FailingConnectionOnly (default), IdleConnections, EntirePool
            ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
                              <xs:element     name="allow-multiple-users"     type="boolean-
presenceType" minOccurs="0" maxOccurs="1">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
           Specifies if multiple users will access the datasource through the getConnection(user,
 password)
              method and hence if the internal pool type should account for that
            ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="xa-poolType">
    <xs:complexContent>
      <xs:extension base="poolType">
        <xs:sequence>
          <xs:element name="is-same-rm-override" type="xs:boolean" minOccurs="0">
            <xs:annotation>
              <xs:documentation>
                <![CDATA[
                  The is-same-rm-override element allows one to unconditionally
                  set whether the javax.transaction.xa.XAResource.isSameRM(XAResource) returns
                  true or false. Ex: <is-same-rm-override>true</is-same-rm-override>
                ]]>
              </xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="interleaving" type="boolean-presenceType" minOccurs="0">
            <xs:annotation>
              <xs:documentation>
                <![CDATA[
                  An element to enable interleaving for XA connection factories
                  Ex: <interleaving/>
```

```
                ]]>
              </xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="no-tx-separate-pools" type="boolean-presenceType" minOccurs="0">
            <xs:annotation>
              <xs:documentation>
                <![CDATA[
                     Oracle does not like XA connections getting used both inside and outside
a JTA transaction.
                       To workaround the problem you can create separate sub-pools for the
different contexts
                  using <no-tx-separate-pools/>
                  Ex: <no-tx-separate-pools/>
                 ]]>
              </xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="pad-xid" type="xs:boolean" default="false" minOccurs="0">
            <xs:annotation>
              <xs:documentation>
                <![CDATA[
                   Should the Xid be padded
                   Ex: <pad-xid>true</pad-xid>
                 ]]>
              </xs:documentation>
            </xs:annotation>
          </xs:element>
          <xs:element name="wrap-xa-resource" type="xs:boolean" default="true" minOccurs="0">
            <xs:annotation>
              <xs:documentation>
                <![CDATA[
                 Should the XAResource instances be wrapped in a org.jboss.tm.XAResourceWrapper
                   instance
                   Ex: <wrap-xa-resource>true</wrap-xa-resource>
                 ]]>
              </xs:documentation>
            </xs:annotation>
          </xs:element>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="dsSecurityType">
    <xs:sequence>
      <xs:element name="user-name" type="xs:token" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
                Specify the username used when creating a new connection.
                Ex: <user-name>sa</user-name>
               ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="password" type="xs:token" minOccurs="0">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
```

```
                Specify the password used when creating a new connection.
                Ex: <password>sa-pass</password>
               ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="security-domain" type="xs:token" minOccurs="0" maxOccurs="1">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
                 Indicates Subject (from security domain) are used 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.
              Ex:
              <security-domain>HsqlDbRealm</security-domain>
            ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
         <xs:element name="reauth-plugin" type="extensionType" minOccurs="0" maxOccurs="1"></
xs:element>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="extensionType">
    <xs:sequence>
                                    <xs:element      name="config-property"      type="config-
propertyType" minOccurs="0" maxOccurs="unbounded"></xs:element>
    </xs:sequence>
    <xs:attribute name="class-name" type="xs:token" use="required"></xs:attribute>
  </xs:complexType>

  <xs:complexType name="config-propertyType" mixed="true">
    <xs:annotation>
      <xs:documentation>
        <![CDATA[
          Specifies a Java bean property value
         ]]>
      </xs:documentation>
    </xs:annotation>
    <xs:simpleContent>
      <xs:extension base="xs:token">
        <xs:attribute use="required" name="name" type="xs:token">
          <xs:annotation>
            <xs:documentation>
              <![CDATA[
                Specifies the name of the config-property
               ]]>
            </xs:documentation>
          </xs:annotation>
        </xs:attribute>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
  <xs:complexType name="recoverType">
    <xs:sequence>
```

```
      <xs:element name="recover-credential" type="dsSecurityType" minOccurs="0" maxOccurs="1">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Specifies the security options used when creating a connection during recovery.
                 Note: if this credential are not specified the security credential are used
 for recover too
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="recover-plugin" type="extensionType" minOccurs="0" maxOccurs="1">
        <xs:annotation>
          <xs:documentation>
            <![CDATA[
              Specifies the extension plugin used in spi (core.spi.xa)
                  which can be implemented by various plugins to provide better feedback to
 the XA recovery system.
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="no-recovery" type="xs:boolean" default="false" use="optional">
      <xs:annotation>
        <xs:documentation>
          <![CDATA[
            Specify if the xa-datasource should be excluded from recovery.
            Default false.
           ]]>
        </xs:documentation>
      </xs:annotation>
    </xs:attribute>
  </xs:complexType>

  <xs:complexType name="driverType">
    <xs:sequence>
      <xs:element name="driver-class" type="xs:token" maxOccurs="1" minOccurs="0">
      <xs:annotation>
          <xs:documentation>
            <![CDATA[
                        The fully qualifed name of the JDBC driver class Ex: <driver-
class>org.hsqldb.jdbcDriver</driver-class>
             ]]>
          </xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="datasource-class" type="xs:token" maxOccurs="1" minOccurs="0">
      <xs:annotation>
          <xs:documentation>
           <![CDATA[
              The fully qualifed name of the javax.sql.DataSource implementation
              class.
            ]]>
          </xs:documentation>
        </xs:annotation></xs:element>
      <xs:element name="xa-datasource-class" type="xs:token" maxOccurs="1" minOccurs="0">
      <xs:annotation>
          <xs:documentation>
```

```
          <![CDATA[
             The fully qualifed name of the javax.sql.XADataSource implementation
                class. Ex: <xa-datasource-class>oracle.jdbc.xa.client.OracleXADataSource</xa-
datasource-class>
            ]]>
         </xs:documentation>
       </xs:annotation></xs:element>
    </xs:sequence>
    <xs:attribute name="name" type="xs:token" use="required">
      <xs:annotation>
        <xs:documentation>
          <![CDATA[
            Specifies the symbolic name of this driver used to reference this driver
           ]]>
        </xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="module" type="xs:token" use="optional">
      <xs:annotation>
        <xs:documentation>
          <![CDATA[
            Specifies the name of AS7 module providing this driver.
            Thios tag is not used in IronJacamar standalone container.
           ]]>
        </xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="major-version" type="xs:int" use="optional">
      <xs:annotation>
        <xs:documentation>
          <![CDATA[
              Specifies the major version of this driver. If the major and minor version is
 obmitted the fist availabe
            Driver in module will be used.
           ]]>
        </xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="minor-verion" type="xs:int" use="optional">
    <xs:annotation>
        <xs:documentation>
          <![CDATA[
              Specifies the minor version of this driver. If the major and minor version is
 obmitted the fist availabe
            Driver in module will be used.
           ]]>
        </xs:documentation>
      </xs:annotation>
    </xs:attribute>
  </xs:complexType>

  <xs:complexType name="driversType">
    <xs:sequence>
          <xs:element  name="driver"  type="driverType"  maxOccurs="unbounded"  minOccurs="1"></
xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

Appendix B. Samples

B.1. HelloWorld example

B.1.1. Introduction

The HelloWorld resource adapter example shows a simple example of how to use and implement the interfaces in the Java EE Connector Architecture specification.

The HelloWorld examples exposes the HelloWorldConnection interface where developers can invoke the exposed methods.

The example shows how to build and test a resource adapter.

B.1.1.1. Setup

The build environment needs various libraries in order to being able to build and test the resource adapter. The setup is done by

```
cd doc/samples/helloworld
cp -R ../../../lib .
cp ../../../bin/ironjacamar-sjc.jar lib/
```

B.1.1.2. Building

Building the resource adapter is done by

```
ant
```

B.1.1.3. Testing

Testing the resource adapter is done by

```
ant test
```

B.1.2. HelloWorld Resource Adapter

```
/*
 * JBoss, Home of Professional Open Source.
 * Copyright 2010, Red Hat Middleware LLC, and individual contributors
 * as indicated by the @author tags. See the copyright.txt file in the
 * distribution for a full listing of individual contributors.
 *
 * This is free software; you can redistribute it and/or modify it
 * under the terms of the GNU Lesser General Public License as
 * published by the Free Software Foundation; either version 2.1 of
 * the License, or (at your option) any later version.
 *
 * This software is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public
 * License along with this software; if not, write to the Free
 * Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA
 * 02110-1301 USA, or see the FSF site: http://www.fsf.org.
 */
package org.jboss.jca.samples.helloworld;

import java.util.logging.Logger;

import javax.resource.ResourceException;
import javax.resource.spi.ActivationSpec;
import javax.resource.spi.BootstrapContext;
import javax.resource.spi.ConfigProperty;
import javax.resource.spi.Connector;
import javax.resource.spi.ResourceAdapter;
import javax.resource.spi.ResourceAdapterInternalException;
import javax.resource.spi.TransactionSupport;
import javax.resource.spi.endpoint.MessageEndpointFactory;

import javax.transaction.xa.XAResource;

/**
 * HelloWorldResourceAdapter
 *
 * @version $Revision: $
 */
@Connector(
   reauthenticationSupport = false,
   transactionSupport = TransactionSupport.TransactionSupportLevel.NoTransaction)
public class HelloWorldResourceAdapter implements ResourceAdapter
{
   /** The logger */
   private static Logger log = Logger.getLogger("HelloWorldResourceAdapter");

   /** Name property */
   @ConfigProperty(defaultValue = "AS 7", supportsDynamicUpdates = true)
   private String name;
```

```
   /**
    * Default constructor
    */
   public HelloWorldResourceAdapter()
   {
   }

   /**
    * Set name
    * @param name The value
    */
   public void setName(String name)
   {
      this.name = name;
   }

   /**
    * Get name
    * @return The value
    */
   public String getName()
   {
      return name;
   }

   /**
    * This is called during the activation of a message endpoint.
    *
    * @param endpointFactory A message endpoint factory instance.
    * @param spec An activation spec JavaBean instance.
    * @throws ResourceException generic exception
    */
   public void endpointActivation(MessageEndpointFactory endpointFactory,
                                  ActivationSpec spec) throws ResourceException
   {
   }

   /**
    * This is called when a message endpoint is deactivated.
    *
    * @param endpointFactory A message endpoint factory instance.
    * @param spec An activation spec JavaBean instance.
    */
   public void endpointDeactivation(MessageEndpointFactory endpointFactory,
                                    ActivationSpec spec)
   {
   }

   /**
    * This is called when a resource adapter instance is bootstrapped.
    *
    * @param ctx A bootstrap context containing references
    * @throws ResourceAdapterInternalException indicates bootstrap failure.
    */
   public void start(BootstrapContext ctx)
      throws ResourceAdapterInternalException
   {
   }
```

```
   /**
    * This is called when a resource adapter instance is undeployed or
    * during application server shutdown.
    */
   public void stop()
   {
   }

   /**
    * This method is called by the application server during crash recovery.
    *
    * @param specs an array of ActivationSpec JavaBeans
    * @throws ResourceException generic exception
    * @return an array of XAResource objects
    */
   public XAResource[] getXAResources(ActivationSpec[] specs)
      throws ResourceException
   {
      return null;
   }

   /**
    * Returns a hash code value for the object.
    * @return A hash code value for this object.
    */
   @Override
   public int hashCode()
   {
      int result = 17;
      if (name != null)
         result += 31 * result + 7 * name.hashCode();
      else
         result += 31 * result + 7;
      return result;
   }

   /**
    * Indicates whether some other object is equal to this one.
    * @param other The reference object with which to compare.
    * @return true If this object is the same as the obj argument, false otherwise.
    */
   @Override
   public boolean equals(Object other)
   {
      if (other == null)
         return false;
      if (other == this)
         return true;
      if (!(other instanceof HelloWorldResourceAdapter))
         return false;
      HelloWorldResourceAdapter obj = (HelloWorldResourceAdapter)other;
      boolean result = true;
      if (result)
      {
         if (name == null)
            result = obj.getName() == null;
         else
            result = name.equals(obj.getName());
      }
```

```
      return result;
   }
}
```

B.1.3. HelloWorld Managed Connection Factory

```
/*
 * JBoss, Home of Professional Open Source.
 * Copyright 2010, Red Hat Middleware LLC, and individual contributors
 * as indicated by the @author tags. See the copyright.txt file in the
 * distribution for a full listing of individual contributors.
 *
 * This is free software; you can redistribute it and/or modify it
 * under the terms of the GNU Lesser General Public License as
 * published by the Free Software Foundation; either version 2.1 of
 * the License, or (at your option) any later version.
 *
 * This software is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public
 * License along with this software; if not, write to the Free
 * Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA
 * 02110-1301 USA, or see the FSF site: http://www.fsf.org.
 */
package org.jboss.jca.samples.helloworld;

import java.io.PrintWriter;
import java.util.Iterator;
import java.util.Set;
import java.util.logging.Logger;

import javax.resource.ResourceException;
import javax.resource.spi.ConnectionDefinition;
import javax.resource.spi.ConnectionManager;
import javax.resource.spi.ConnectionRequestInfo;
import javax.resource.spi.ManagedConnection;
import javax.resource.spi.ManagedConnectionFactory;
import javax.resource.spi.ResourceAdapter;
import javax.resource.spi.ResourceAdapterAssociation;

import javax.security.auth.Subject;

/**
 * HelloWorldManagedConnectionFactory
 *
 * @version $Revision: $
 */
@ConnectionDefinition(connectionFactory = HelloWorldConnectionFactory.class,
   connectionFactoryImpl = HelloWorldConnectionFactoryImpl.class,
   connection = HelloWorldConnection.class,
```

```
   connectionImpl = HelloWorldConnectionImpl.class)
public class HelloWorldManagedConnectionFactory
   implements ManagedConnectionFactory, ResourceAdapterAssociation
{

   /** The serialVersionUID */
   private static final long serialVersionUID = 1L;

   /** The logger */
   private static Logger log = Logger.getLogger("HelloWorldManagedConnectionFactory");

   /** The resource adapter */
   private ResourceAdapter ra;

   /** The logwriter */
   private PrintWriter logwriter;

   /**
    * Default constructor
    */
   public HelloWorldManagedConnectionFactory()
   {
      this.ra = null;
      this.logwriter = null;
   }

   /**
    * Creates a Connection Factory instance.
    *
    * @return EIS-specific Connection Factory instance or
    *         javax.resource.cci.ConnectionFactory instance
    * @throws ResourceException Generic exception
    */
   public Object createConnectionFactory() throws ResourceException
   {
          throw new ResourceException("This resource adapter doesn't support non-managed
 environments");
   }

   /**
    * Creates a Connection Factory instance.
    *
    * @param cxManager ConnectionManager to be associated with created EIS
    *        connection factory instance
    * @return EIS-specific Connection Factory instance or
    *        javax.resource.cci.ConnectionFactory instance
    * @throws ResourceException Generic exception
    */
   public Object createConnectionFactory(ConnectionManager cxManager) throws ResourceException
   {
      return new HelloWorldConnectionFactoryImpl(this, cxManager);
   }

   /**
    * Creates a new physical connection to the underlying EIS resource manager.
    *
    * @param subject Caller's security information
    * @param cxRequestInfo Additional resource adapter specific connection
    *        request information
```

```
     * @throws ResourceException generic exception
     * @return ManagedConnection instance
     */
    public ManagedConnection createManagedConnection(Subject subject,
                                                     ConnectionRequestInfo cxRequestInfo)
       throws ResourceException
    {
       return new HelloWorldManagedConnection(this);
    }

    /**
     * Returns a matched connection from the candidate set of connections.
     *
     * @param connectionSet Candidate connection set
     * @param subject Caller's security information
     * @param cxRequestInfo Additional resource adapter specific connection request information
     * @throws ResourceException generic exception
     * @return ManagedConnection if resource adapter finds an acceptable match otherwise null
     */
    public ManagedConnection matchManagedConnections(Set connectionSet,
                                        Subject subject, ConnectionRequestInfo cxRequestInfo)
       throws ResourceException
    {
       ManagedConnection result = null;

       Iterator it = connectionSet.iterator();
       while (result == null && it.hasNext())
       {
          ManagedConnection mc = (ManagedConnection)it.next();
          if (mc instanceof HelloWorldManagedConnection)
          {
             HelloWorldManagedConnection hwmc = (HelloWorldManagedConnection)mc;
             result = hwmc;
          }
       }

       return result;
    }

    /**
     * Get the log writer for this ManagedConnectionFactory instance.
     *
     * @return PrintWriter
     * @throws ResourceException generic exception
     */
    public PrintWriter getLogWriter() throws ResourceException
    {
       return logwriter;
    }

    /**
     * Set the log writer for this ManagedConnectionFactory instance.
     *
     * @param out PrintWriter - an out stream for error logging and tracing
     * @throws ResourceException generic exception
     */
    public void setLogWriter(PrintWriter out) throws ResourceException
    {
       logwriter = out;
```

```
   }

   /**
    * Get the resource adapter
    *
    * @return The handle
    */
   public ResourceAdapter getResourceAdapter()
   {
      return ra;
   }

   /**
    * Set the resource adapter
    *
    * @param ra The handle
    */
   public void setResourceAdapter(ResourceAdapter ra)
   {
      this.ra = ra;
   }

   /**
    * Returns a hash code value for the object.
    * @return A hash code value for this object.
    */
   @Override
   public int hashCode()
   {
      int result = 17;
      return result;
   }

   /**
    * Indicates whether some other object is equal to this one.
    * @param other The reference object with which to compare.
    * @return true If this object is the same as the obj argument, false otherwise.
    */
   @Override
   public boolean equals(Object other)
   {
      if (other == null)
         return false;
      if (other == this)
         return true;
      if (!(other instanceof HelloWorldManagedConnectionFactory))
         return false;
      HelloWorldManagedConnectionFactory obj = (HelloWorldManagedConnectionFactory)other;
      boolean result = true;
      return result;
   }


}
```

B.1.4. HelloWorld Managed Connection

```
/*
 * JBoss, Home of Professional Open Source.
 * Copyright 2010, Red Hat Middleware LLC, and individual contributors
 * as indicated by the @author tags. See the copyright.txt file in the
 * distribution for a full listing of individual contributors.
 *
 * This is free software; you can redistribute it and/or modify it
 * under the terms of the GNU Lesser General Public License as
 * published by the Free Software Foundation; either version 2.1 of
 * the License, or (at your option) any later version.
 *
 * This software is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public
 * License along with this software; if not, write to the Free
 * Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA
 * 02110-1301 USA, or see the FSF site: http://www.fsf.org.
 */
package org.jboss.jca.samples.helloworld;

import java.io.PrintWriter;
import java.util.ArrayList;
import java.util.List;
import java.util.logging.Logger;

import javax.resource.NotSupportedException;
import javax.resource.ResourceException;
import javax.resource.spi.ConnectionEvent;
import javax.resource.spi.ConnectionEventListener;
import javax.resource.spi.ConnectionRequestInfo;
import javax.resource.spi.LocalTransaction;
import javax.resource.spi.ManagedConnection;
import javax.resource.spi.ManagedConnectionMetaData;

import javax.security.auth.Subject;
import javax.transaction.xa.XAResource;

/**
 * HelloWorldManagedConnection
 *
 * @version $Revision: $
 */
public class HelloWorldManagedConnection implements ManagedConnection
{
   /** The logger */
   private static Logger log = Logger.getLogger("HelloWorldManagedConnection");

   /** MCF */
   private HelloWorldManagedConnectionFactory mcf;

   /** Log writer */
```

```
    private PrintWriter logWriter;

    /** Listeners */
    private List<ConnectionEventListener> listeners;

    /** Connection */
    private Object connection;

    /**
     * default constructor
     * @param mcf mcf
     */
    public HelloWorldManagedConnection(HelloWorldManagedConnectionFactory mcf)
    {
       this.mcf = mcf;
       this.logWriter = null;
       this.listeners = new ArrayList<ConnectionEventListener>(1);
       this.connection = null;
    }

    /**
     * Creates a new connection handle for the underlying physical connection
     * represented by the ManagedConnection instance.
     *
     * @param subject Security context as JAAS subject
     * @param cxRequestInfo ConnectionRequestInfo instance
     * @return generic Object instance representing the connection handle.
     * @throws ResourceException generic exception if operation fails
     */
    public Object getConnection(Subject subject,
                                ConnectionRequestInfo cxRequestInfo)
       throws ResourceException
    {
       connection = new HelloWorldConnectionImpl(this, mcf);

       return connection;
    }

    /**
     * Used by the container to change the association of an
     * application-level connection handle with a ManagedConneciton instance.
     *
     * @param connection Application-level connection handle
     * @throws ResourceException generic exception if operation fails
     */
    public void associateConnection(Object connection) throws ResourceException
    {
       this.connection = connection;
    }

    /**
     * Application server calls this method to force any cleanup on
     * the ManagedConnection instance.
     *
     * @throws ResourceException generic exception if operation fails
     */
    public void cleanup() throws ResourceException
    {
    }
```

```
/**
 * Destroys the physical connection to the underlying resource manager.
 *
 * @throws ResourceException generic exception if operation fails
 */
public void destroy() throws ResourceException
{
   this.connection = null;
}

/**
 * Adds a connection event listener to the ManagedConnection instance.
 *
 * @param listener A new ConnectionEventListener to be registered
 */
public void addConnectionEventListener(ConnectionEventListener listener)
{
   if (listener == null)
      throw new IllegalArgumentException("Listener is null");

   listeners.add(listener);
}

/**
 * Removes an already registered connection event listener
 * from the ManagedConnection instance.
 *
 * @param listener Already registered connection event listener to be removed
 */
public void removeConnectionEventListener(ConnectionEventListener listener)
{
   if (listener == null)
      throw new IllegalArgumentException("Listener is null");

   listeners.remove(listener);
}

/**
 * Gets the log writer for this ManagedConnection instance.
 *
 * @return Character ourput stream associated with this
 *         Managed-Connection instance
 * @throws ResourceException generic exception if operation fails
 */
public PrintWriter getLogWriter() throws ResourceException
{
   return logWriter;
}

/**
 * Sets the log writer for this ManagedConnection instance.
 *
 * @param out Character Output stream to be associated
 * @throws ResourceException generic exception if operation fails
 */
public void setLogWriter(PrintWriter out) throws ResourceException
{
   this.logWriter = out;
```

```
    }

    /**
     * Returns an <code>javax.resource.spi.LocalTransaction</code> instance.
     *
     * @return LocalTransaction instance
     * @throws ResourceException generic exception if operation fails
     */
    public LocalTransaction getLocalTransaction() throws ResourceException
    {
       throw new NotSupportedException("LocalTransaction not supported");
    }

    /**
     * Returns an <code>javax.transaction.xa.XAresource</code> instance.
     *
     * @return XAResource instance
     * @throws ResourceException generic exception if operation fails
     */
    public XAResource getXAResource() throws ResourceException
    {
       throw new NotSupportedException("GetXAResource not supported");
    }

    /**
     * Gets the metadata information for this connection's underlying
     * EIS resource manager instance.
     *
     * @return ManagedConnectionMetaData instance
     * @throws ResourceException generic exception if operation fails
     */
    public ManagedConnectionMetaData getMetaData() throws ResourceException
    {
       return new HelloWorldManagedConnectionMetaData();
    }

    /**
     * Call helloWorld
     * @param name String name
     * @return String helloworld
     */
    String helloWorld(String name)
    {
       return "Hello World, " + name + " !";
    }

    /**
     * Close handle
     * @param handle The handle
     */
    void closeHandle(HelloWorldConnection handle)
    {
       ConnectionEvent event = new ConnectionEvent(this, ConnectionEvent.CONNECTION_CLOSED);
       event.setConnectionHandle(handle);

       for (ConnectionEventListener cel : listeners)
       {
          cel.connectionClosed(event);
       }
```

```
   }
}
```

B.1.5. HelloWorld Connection Factory

```
/*
 * JBoss, Home of Professional Open Source.
 * Copyright 2010, Red Hat Middleware LLC, and individual contributors
 * as indicated by the @author tags. See the copyright.txt file in the
 * distribution for a full listing of individual contributors.
 *
 * This is free software; you can redistribute it and/or modify it
 * under the terms of the GNU Lesser General Public License as
 * published by the Free Software Foundation; either version 2.1 of
 * the License, or (at your option) any later version.
 *
 * This software is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public
 * License along with this software; if not, write to the Free
 * Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA
 * 02110-1301 USA, or see the FSF site: http://www.fsf.org.
 */
package org.jboss.jca.samples.helloworld;

import java.io.Serializable;

import javax.resource.Referenceable;
import javax.resource.ResourceException;

/**
 * HelloWorldConnectionFactory
 *
 * @version $Revision: $
 */
public interface HelloWorldConnectionFactory extends Serializable, Referenceable
{
   /**
    * Get connection from factory
    *
    * @return HelloWorldConnection instance
    * @exception ResourceException Thrown if a connection can't be obtained
    */
   public HelloWorldConnection getConnection() throws ResourceException;

}
```

B.1.6. HelloWorld Connection Factory Implementation

```
/*
 * JBoss, Home of Professional Open Source.
 * Copyright 2010, Red Hat Middleware LLC, and individual contributors
 * as indicated by the @author tags. See the copyright.txt file in the
 * distribution for a full listing of individual contributors.
 *
 * This is free software; you can redistribute it and/or modify it
 * under the terms of the GNU Lesser General Public License as
 * published by the Free Software Foundation; either version 2.1 of
 * the License, or (at your option) any later version.
 *
 * This software is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public
 * License along with this software; if not, write to the Free
 * Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA
 * 02110-1301 USA, or see the FSF site: http://www.fsf.org.
 */
package org.jboss.jca.samples.helloworld;

import javax.naming.NamingException;
import javax.naming.Reference;

import javax.resource.ResourceException;
import javax.resource.spi.ConnectionManager;

/**
 * HelloWorldConnectionFactoryImpl
 *
 * @version $Revision: $
 */
public class HelloWorldConnectionFactoryImpl implements HelloWorldConnectionFactory
{
   /** The serialVersionUID */
   private static final long serialVersionUID = 1L;

   private Reference reference;

   private HelloWorldManagedConnectionFactory mcf;
   private ConnectionManager connectionManager;

   /**
    * Default constructor
    * @param mcf ManagedConnectionFactory
    * @param cxManager ConnectionManager
    */
   public HelloWorldConnectionFactoryImpl(HelloWorldManagedConnectionFactory mcf,
                                          ConnectionManager cxManager)
   {
      this.mcf = mcf;
      this.connectionManager = cxManager;
```

```
   }

   /**
    * Get connection from factory
    *
    * @return HelloWorldConnection instance
    * @exception ResourceException Thrown if a connection can't be obtained
    */
   @Override
   public HelloWorldConnection getConnection() throws ResourceException
   {
      return (HelloWorldConnection)connectionManager.allocateConnection(mcf, null);
   }

   /**
    * Get the Reference instance.
    *
    * @return Reference instance
    * @exception NamingException Thrown if a reference can't be obtained
    */
   @Override
   public Reference getReference() throws NamingException
   {
      return reference;
   }

   /**
    * Set the Reference instance.
    *
    * @param reference A Reference instance
    */
   @Override
   public void setReference(Reference reference)
   {
      this.reference = reference;
   }
}
```

B.1.7. HelloWorld Connection

```
/*
 * JBoss, Home of Professional Open Source.
 * Copyright 2010, Red Hat Middleware LLC, and individual contributors
 * as indicated by the @author tags. See the copyright.txt file in the
 * distribution for a full listing of individual contributors.
 *
 * This is free software; you can redistribute it and/or modify it
 * under the terms of the GNU Lesser General Public License as
 * published by the Free Software Foundation; either version 2.1 of
 * the License, or (at your option) any later version.
 *
 * This software is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
```

```
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public
 * License along with this software; if not, write to the Free
 * Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA
 * 02110-1301 USA, or see the FSF site: http://www.fsf.org.
 */
package org.jboss.jca.samples.helloworld;

/**
 * HelloWorldConnection
 *
 * @version $Revision: $
 */
public interface HelloWorldConnection
{
   /**
    * HelloWorld
    * @return String
    */
   public String helloWorld();

   /**
    * HelloWorld
    * @param name A name
    * @return String
    */
   public String helloWorld(String name);

   /**
    * Close
    */
   public void close();
}
```

B.1.8. HelloWorld Connection Implementation

```
/*
 * JBoss, Home of Professional Open Source.
 * Copyright 2010, Red Hat Middleware LLC, and individual contributors
 * as indicated by the @author tags. See the copyright.txt file in the
 * distribution for a full listing of individual contributors.
 *
 * This is free software; you can redistribute it and/or modify it
 * under the terms of the GNU Lesser General Public License as
 * published by the Free Software Foundation; either version 2.1 of
 * the License, or (at your option) any later version.
 *
 * This software is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
 * Lesser General Public License for more details.
```

```
 *
 * You should have received a copy of the GNU Lesser General Public
 * License along with this software; if not, write to the Free
 * Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA
 * 02110-1301 USA, or see the FSF site: http://www.fsf.org.
 */
package org.jboss.jca.samples.helloworld;

import java.util.logging.Logger;

/**
 * HelloWorldConnectionImpl
 *
 * @version $Revision: $
 */
public class HelloWorldConnectionImpl implements HelloWorldConnection
{
   /** The logger */
   private static Logger log = Logger.getLogger("HelloWorldConnectionImpl");

   /** ManagedConnection */
   private HelloWorldManagedConnection mc;

   /** ManagedConnectionFactory */
   private HelloWorldManagedConnectionFactory mcf;

   /**
    * Default constructor
    * @param mc HelloWorldManagedConnection
    * @param mcf HelloWorldManagedConnectionFactory
    */
   public HelloWorldConnectionImpl(HelloWorldManagedConnection mc,
                                   HelloWorldManagedConnectionFactory mcf)
   {
      this.mc = mc;
      this.mcf = mcf;
   }

   /**
    * Call helloWorld
    * @return String helloworld
    */
   public String helloWorld()
   {
      return helloWorld(((HelloWorldResourceAdapter)mcf.getResourceAdapter()).getName());
   }

   /**
    * Call helloWorld
    * @param name String name
    * @return String helloworld
    */
   public String helloWorld(String name)
   {
      return mc.helloWorld(name);
   }

   /**
    * Close
```

```
    */
   public void close()
   {
      mc.closeHandle(this);
   }
}
```

B.1.9. HelloWorld Managed Connection MetaData

```
/*
 * JBoss, Home of Professional Open Source.
 * Copyright 2010, Red Hat Middleware LLC, and individual contributors
 * as indicated by the @author tags. See the copyright.txt file in the
 * distribution for a full listing of individual contributors.
 *
 * This is free software; you can redistribute it and/or modify it
 * under the terms of the GNU Lesser General Public License as
 * published by the Free Software Foundation; either version 2.1 of
 * the License, or (at your option) any later version.
 *
 * This software is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public
 * License along with this software; if not, write to the Free
 * Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA
 * 02110-1301 USA, or see the FSF site: http://www.fsf.org.
 */
package org.jboss.jca.samples.helloworld;

import javax.resource.ResourceException;

import javax.resource.spi.ManagedConnectionMetaData;

/**
 * HelloWorldManagedConnectionMetaData
 *
 * @version $Revision: $
 */
public class HelloWorldManagedConnectionMetaData implements ManagedConnectionMetaData
{
   /**
    * Default constructor
    */
   public HelloWorldManagedConnectionMetaData()
   {
   }

   /**
    * Returns Product name of the underlying EIS instance connected
    * through the ManagedConnection.
```

```
     *
     * @return Product name of the EIS instance
     * @throws ResourceException Thrown if an error occurs
     */
    @Override
    public String getEISProductName() throws ResourceException
    {
       return "HelloWorld Resource Adapter";
    }

    /**
     * Returns Product version of the underlying EIS instance connected
     * through the ManagedConnection.
     *
     * @return Product version of the EIS instance
     * @throws ResourceException Thrown if an error occurs
     */
    @Override
    public String getEISProductVersion() throws ResourceException
    {
       return "1.0";
    }

    /**
     * Returns maximum limit on number of active concurrent connections
     *
     * @return Maximum limit for number of active concurrent connections
     * @throws ResourceException Thrown if an error occurs
     */
    @Override
    public int getMaxConnections() throws ResourceException
    {
       return 0;
    }

    /**
     * Returns name of the user associated with the ManagedConnection instance
     *
     * @return Name of the user
     * @throws ResourceException Thrown if an error occurs
     */
    @Override
    public String getUserName() throws ResourceException
    {
       return null;
    }
}
```

B.1.10. HelloWorld ironjacamar.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<ironjacamar>
  <connection-definitions>
```

```
    <connection-definition
     class-name="org.jboss.jca.samples.helloworld.HelloWorldManagedConnectionFactory"
     jndi-name="java:/eis/HelloWorld"/>
  </connection-definitions>
</ironjacamar>
```

B.1.11. HelloWorld Connection Test Case

```
/*
 * JBoss, Home of Professional Open Source.
 * Copyright 2010, Red Hat Middleware LLC, and individual contributors
 * as indicated by the @author tags. See the copyright.txt file in the
 * distribution for a full listing of individual contributors.
 *
 * This is free software; you can redistribute it and/or modify it
 * under the terms of the GNU Lesser General Public License as
 * published by the Free Software Foundation; either version 2.1 of
 * the License, or (at your option) any later version.
 *
 * This software is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public
 * License along with this software; if not, write to the Free
 * Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA
 * 02110-1301 USA, or see the FSF site: http://www.fsf.org.
 */
package org.jboss.jca.samples.helloworld;

import java.util.UUID;
import java.util.logging.Logger;

import javax.annotation.Resource;

import org.jboss.arquillian.container.test.api.Deployment;
import org.jboss.arquillian.junit.Arquillian;

import org.jboss.shrinkwrap.api.ShrinkWrap;
import org.jboss.shrinkwrap.api.spec.JavaArchive;
import org.jboss.shrinkwrap.api.spec.ResourceAdapterArchive;

import org.junit.Test;
import org.junit.runner.RunWith;
import static org.junit.Assert.*;


/**
 * ConnectorTestCase
 *
 * @version $Revision: $
 */
```

```
@RunWith(Arquillian.class)
public class ConnectorTestCase
{
   private static Logger log = Logger.getLogger("ConnectorTestCase");

   private static String deploymentName = "ConnectorTestCase";

   /**
    * Define the deployment
    *
    * @return The deployment archive
    */
   @Deployment
   public static ResourceAdapterArchive createDeployment()
   {
      ResourceAdapterArchive raa =
         ShrinkWrap.create(ResourceAdapterArchive.class, deploymentName + ".rar");
      JavaArchive ja = ShrinkWrap.create(JavaArchive.class,
         UUID.randomUUID().toString() + ".jar");
      ja.addClasses(HelloWorldResourceAdapter.class,
         HelloWorldManagedConnectionFactory.class,
         HelloWorldManagedConnection.class,
         HelloWorldManagedConnectionMetaData.class,
         HelloWorldConnectionFactory.class,
         HelloWorldConnectionFactoryImpl.class,
         HelloWorldConnection.class,
         HelloWorldConnectionImpl.class);
      raa.addAsLibrary(ja);
      raa.addAsManifestResource("META-INF/ironjacamar.xml", "ironjacamar.xml");

      return raa;
   }

   /** resource */
   @Resource(mappedName = "java:/eis/HelloWorld")
   private HelloWorldConnectionFactory connectionFactory;

   /**
    * Test helloWorld
    *
    * @exception Throwable Thrown if case of an error
    */
   @Test
   public void testHelloWorldNoArgs() throws Throwable
   {
      assertNotNull(connectionFactory);
      HelloWorldConnection connection = connectionFactory.getConnection();
      assertNotNull(connection);
      String result = connection.helloWorld();
      connection.close();
   }

   /**
    * Test helloWorld
    *
    * @exception Throwable Thrown if case of an error
    */
   @Test
   public void testHelloWorldNameString() throws Throwable
```

```
   {
      assertNotNull(connectionFactory);
      HelloWorldConnection connection = connectionFactory.getConnection();
      assertNotNull(connection);
      String result = connection.helloWorld(null);
      connection.close();
   }
}
```

B.1.12. HelloWorld Ant build.xml

```
<!--
/*
 * JBoss, Home of Professional Open Source.
 * Copyright 2010, Red Hat Middleware LLC, and individual contributors
 * as indicated by the @author tags. See the copyright.txt file in the
 * distribution for a full listing of individual contributors.
 *
 * This is free software; you can redistribute it and/or modify it
 * under the terms of the GNU Lesser General Public License as
 * published by the Free Software Foundation; either version 2.1 of
 * the License, or (at your option) any later version.
 *
 * This software is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public
 * License along with this software; if not, write to the Free
 * Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA
 * 02110-1301 USA, or see the FSF site: http://www.fsf.org.
 */
-->

<project name="helloworld" basedir="." default="rar">

  <!-- ================================
       Properties
       ================================ -->
  <property name="build.dir" value="${basedir}/build" />
  <property name="target.dir" value="${basedir}/target" />
  <property name="lib.dir" value="${basedir}/lib" />

  <property name="javac.debug" value="on" />
  <property name="javac.deprecation" value="on" />
  <property name="javac.optimize" value="off" />

  <property name="junit.printsummary" value="yes" />
  <property name="junit.haltonerror" value="no" />
  <property name="junit.haltonfailure" value="no" />
  <property name="junit.fork" value="yes" />
  <property name="junit.timeout" value="60000" />
```

```
<property name="junit.jvm" value="" />
<property name="junit.jvm.options" value="-Xms128m -Xmx512m -XX:MaxPermSize=256m" />
<property name="junit.batchtest.haltonerror" value="no" />
<property name="junit.batchtest.haltonfailure" value="no" />
<property name="junit.batchtest.fork" value="yes" />

<path id="lib.path.id">
  <fileset dir="${lib.dir}">
    <include name="**/*.jar"/>
  </fileset>
</path>

<path id="test.lib.path.id">
  <fileset dir="${lib.dir}">
    <include name="**/*.jar"/>
  </fileset>
  <fileset dir="${build.dir}">
    <include name="**/*.jar"/>
  </fileset>
</path>

<!-- ================================
     Target: init
     ================================ -->
<target name="init">
  <mkdir dir="${lib.dir}" />
</target>

<!-- ================================
     Target: compile
     ================================ -->
<target name="compile" depends="init">
  <mkdir dir="${build.dir}" />

  <javac srcdir="${basedir}/src/main/java"
         destdir="${build.dir}"
         classpathref="lib.path.id"
         debug="${javac.debug}"
         deprecation="${javac.deprecation}"
         optimize="${javac.optimize}">
  </javac>
</target>

<!-- ================================
     Target: rar
     ================================ -->
<target name="rar" depends="compile">
  <mkdir dir="${target.dir}" />
  <mkdir dir="${basedir}/src/main/resources" />
  <jar destfile="${build.dir}/helloworld.jar"
       basedir="${build.dir}"
       includes="**/*.class"/>
  <jar destfile="${target.dir}/helloworld.rar">
    <fileset dir="${basedir}/src/main/resources" includes="META-INF/*"/>
    <fileset dir="${build.dir}" includes="**/*.jar"/>
  </jar>
</target>
```

```
  <!-- ================================
       Target: prepare-test
       ================================ -->
  <target name="prepare-test" depends="init">
    <mkdir dir="${build.dir}/test" />

    <javac srcdir="src/test"
           destdir="${build.dir}/test"
           classpathref="test.lib.path.id"
           debug="${javac.debug}"
           deprecation="${javac.deprecation}"
           optimize="${javac.optimize}">
      <compilerarg value="-Xlint"/>
    </javac>

    <copy todir="${build.dir}/test">
      <fileset dir="src/main/resources"/>
      <fileset dir="src/test/resources"/>
    </copy>
  </target>

  <!-- ================================
       Target: test
       ================================ -->
  <target name="test" depends="rar, prepare-test">
    <mkdir dir="${basedir}/reports"/>

    <junit dir="src/test"
           printsummary="${junit.printsummary}"
           haltonerror="${junit.haltonerror}"
           haltonfailure="${junit.haltonfailure}"
           fork="${junit.fork}"
           timeout="${junit.timeout}">

      <jvmarg line="${junit.jvm.options}"/>
      <sysproperty key="archives.dir" value="${target.dir}"/>
      <sysproperty key="reports.dir" value="${basedir}/reports"/>
      <sysproperty key="java.util.logging.manager" value="org.jboss.logmanager.LogManager"/>
      <sysproperty key="log4j.defaultInitOverride" value="true"/>
      <sysproperty key="org.jboss.logging.Logger.pluginClass"
                   value="org.jboss.logging.logmanager.LoggerPluginImpl"/>
      <sysproperty key="test.dir" value="${build.dir}/test"/>
      <sysproperty key="xb.builder.useUnorderedSequence" value="true"/>

      <classpath>
        <fileset dir="${lib.dir}" includes="**/*.jar" />
        <fileset dir="${build.dir}" includes="*.jar" />
        <pathelement location="${build.dir}/test"/>
      </classpath>

      <formatter type="plain"/>
      <formatter type="xml"/>

      <batchtest todir="${basedir}/reports"
                 haltonerror="${junit.batchtest.haltonerror}"
                 haltonfailure="${junit.batchtest.haltonfailure}"
                 fork="${junit.batchtest.fork}">

        <fileset dir="${build.dir}/test">
```

```
          <include name="**/*TestCase.class"/>
        </fileset>
      </batchtest>

    </junit>

  </target>

  <!-- ================================
       Target: docs
       ================================ -->
  <target name="docs" depends="compile">
    <mkdir dir="${target.dir}/docs"/>
    <javadoc packagenames="*"
             sourcepath="src/main/java"
             destdir="${target.dir}/docs"
             classpathref="lib.path.id">
    </javadoc>
  </target>

  <!-- ================================
       Target: clean
       ================================ -->
  <target name="clean">
    <delete>
      <fileset dir="${basedir}" defaultexcludes="no">
        <include name="**/*~"/>
        <include name="**/*.bak"/>
      </fileset>
    </delete>
    <delete dir="${build.dir}"/>
    <delete dir="${target.dir}"/>
    <delete dir="${basedir}/reports"/>
  </target>

  <!-- ================================
       Target: dist-clean
       ================================ -->
  <target name="dist-clean" depends="init,clean">
    <delete includeemptydirs="true">
      <fileset dir="${lib.dir}" includes="**/*"/>
    </delete>
  </target>

</project>
```

Appendix C. Datasources

The datasource schema can found at *http://www.jboss.org/ironjacamar/schema/datasources_1_0.xsd*.

C.1. PosgreSQL

```
      <?xml version="1.0" encoding="UTF-8"?>
<!-- See http://www.jboss.org/community/wiki/Multiple1PC for information about datasource -->
<datasources xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                      xsi:noNamespaceSchemaLocation="http://www.jboss.org/ironjacamar/schema/
datasources_1_0.xsd">

  <datasource jndi-name="PostgresDS" pool-name="PostgresDS">
    <connection-url>jdbc:postgresql://[servername]:[port]/[database name]</connection-url>
    <driver-class>org.postgresql.Driver</driver-class>
    <security>
      <user-name>x</user-name>
      <password>y</password>
    </security>
    <validation>
                                                    <valid-connection-checker         class-
name="org.jboss.jca.adapters.jdbc.extensions.postgres.PostgreSQLValidConnectionChecker"></
valid-connection-checker>
                                                          <exception-sorter          class-
name="org.jboss.jca.adapters.jdbc.extensions.postgres.PostgreSQLExceptionSorter"></exception-
sorter>
    </validation>
  </datasource>

</datasources>
```

C.2. PosgreSQL XA

```
      <?xml version="1.0" encoding="UTF-8"?>

<datasources xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                      xsi:noNamespaceSchemaLocation="http://www.jboss.org/ironjacamar/schema/
datasources_1_0.xsd">

  <xa-datasource jndi-name="PostgresDS" pool-name="PostgresDS">
    <xa-datasource-property name="ServerName">server_name</xa-datasource-property>
    <xa-datasource-property name="PortNumber">5432</xa-datasource-property>
    <xa-datasource-property name="DatabaseName">database_name</xa-datasource-property>
    <xa-datasource-property name="User">user</xa-datasource-property>
    <xa-datasource-property name="Password">password</xa-datasource-property>
    <xa-datasource-class>org.postgresql.xa.PGXADataSource</xa-datasource-class>
```

```
    <validation>
                                                    <valid-connection-checker         class-
name="org.jboss.jca.adapters.jdbc.extensions.postgres.PostgreSQLValidConnectionChecker"></
valid-connection-checker>
                                                            <exception-sorter         class-
name="org.jboss.jca.adapters.jdbc.extensions.postgres.PostgreSQLExceptionSorter"></exception-
sorter>
    </validation>
  </xa-datasource>

</datasources>
```

C.3. MySQL

```
      <?xml version="1.0" encoding="UTF-8"?>
<!-- See http://www.jboss.org/community/wiki/Multiple1PC for information about datasource -->

<datasources xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                      xsi:noNamespaceSchemaLocation="http://www.jboss.org/ironjacamar/schema/
datasources_1_0.xsd">

  <datasource jndi-name="MySqlDS" pool-name="MySqlDS">
    <connection-url>jdbc:mysql://mysql-hostname:3306/jbossdb</connection-url>
    <driver-class>com.mysql.jdbc.Driver</driver-class>
    <security>
      <user-name>x</user-name>
      <password>y</password>
    </security>
    <validation>
                                                    <valid-connection-checker         class-
name="org.jboss.jca.adapters.jdbc.extensions.mysql.MySQLValidConnectionChecker"></valid-
connection-checker>
                                                            <exception-sorter         class-
name="org.jboss.jca.adapters.jdbc.extensions.mysql.MySQLExceptionSorter"></exception-sorter>
    </validation>
  </datasource>

</datasources>
```

C.4. MySQL XA

```
      <?xml version="1.0" encoding="UTF-8"?>

<datasources xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                      xsi:noNamespaceSchemaLocation="http://www.jboss.org/ironjacamar/schema/
datasources_1_0.xsd">
```

```
  <xa-datasource jndi-name="MysqlXADS" pool-name="MysqlXADS">
    <xa-datasource-property name="ServerName">server_name</xa-datasource-property>
    <xa-datasource-property name="DatabaseName">database_name</xa-datasource-property>
    <xa-datasource-property name="User">user</xa-datasource-property>
    <xa-datasource-property name="Password">password</xa-datasource-property>
    <xa-datasource-class>com.mysql.jdbc.jdbc2.optional.MysqlXADataSource</xa-datasource-class>
    <validation>
                                                    <valid-connection-checker         class-
name="org.jboss.jca.adapters.jdbc.extensions.mysql.MySQLValidConnectionChecker"></valid-
connection-checker>
                                                          <exception-sorter          class-
name="org.jboss.jca.adapters.jdbc.extensions.mysql.MySQLExceptionSorter"></exception-sorter>
    </validation>
  </xa-datasource>

</datasources>
```

C.5. H2

```
      <?xml version="1.0" encoding="UTF-8"?>

<datasources xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                      xsi:noNamespaceSchemaLocation="http://www.jboss.org/ironjacamar/schema/
datasources_1_0.xsd">

  <datasource jndi-name="java:/H2DS" pool-name="H2DS">
    <connection-url>jdbc:h2:mem:test;DB_CLOSE_DELAY=-1</connection-url>
    <driver-class>org.h2.Driver</driver-class>
    <security>
      <user-name>sa</user-name>
      <password>sa</password>
    </security>
  </datasource>

</datasources>
```

C.6. H2 XA

```
      <?xml version="1.0" encoding="UTF-8"?>

<datasources xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                      xsi:noNamespaceSchemaLocation="http://www.jboss.org/ironjacamar/schema/
datasources_1_0.xsd">

  <xa-datasource jndi-name="java:/H2XADS" pool-name="H2XADS">
```

```
    <xa-datasource-class>org.h2.jdbcx.JdbcDataSource</xa-datasource-class>
    <xa-datasource-property name="URL">jdbc:h2:mem:test</xa-datasource-property>
    <security>
      <!-- Have to defined as a primary property - otherwise it won't work -->
      <user-name>sa</user-name>
      <!-- Have to defined as a primary property - otherwise it won't work -->
      <password>sa</password>
    </security>
  </xa-datasource>

</datasources>
```

C.7. Derby

```
      <?xml version="1.0" encoding="UTF-8"?>
<!-- See http://www.jboss.org/community/wiki/Multiple1PC for information about datasource -->
<datasources xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                      xsi:noNamespaceSchemaLocation="http://www.jboss.org/ironjacamar/schema/
datasources_1_0.xsd">

  <datasource jndi-name="DerbyDS" pool-name="DerbyDS">
       <connection-url>jdbc:derby:${ironjacamar.home}${/}data${/}derby${/}localDB;create=true</
connection-url>
    <driver-class>org.apache.derby.jdbc.EmbeddedDriver</driver-class>
    <pool>
      <min-pool-size>5</min-pool-size>
      <max-pool-size>20</max-pool-size>
    </pool>
    <security>
      <user-name>sa</user-name>
      <password></password>
    </security>
    <timeout>
      <idle-timeout-minutes>5</idle-timeout-minutes>
    </timeout>
    <statement>
      <track-statements>true</track-statements>
    </statement>
  </datasource>

</datasources>
```

C.8. Derby XA

```
      <?xml version="1.0" encoding="UTF-8"?>
```

```
<datasources xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                     xsi:noNamespaceSchemaLocation="http://www.jboss.org/ironjacamar/schema/
datasources_1_0.xsd">

  <xa-datasource jndi-name="XADerbyDS" pool-name="XADerbyDS">
    <xa-datasource-property name="DatabaseName">derby/default</xa-datasource-property>
    <xa-datasource-property name="User">sa</xa-datasource-property>
    <xa-datasource-property name="Password"></xa-datasource-property>
    <xa-datasource-property name="CreateDatabase">create</xa-datasource-property>
    <xa-datasource-class>org.apache.derby.jdbc.EmbeddedXADataSource</xa-datasource-class>

    <xa-pool>
      <is-same-rm-override>false</is-same-rm-override>
      <!-- Uncomment to enable interleaving <interleaving/> -->
    </xa-pool>
  </xa-datasource>

</datasources>
```

C.9. Oracle

```
      <?xml version="1.0" encoding="UTF-8"?>
<!-- See http://www.jboss.org/community/wiki/Multiple1PC for information about datasource -->
<datasources xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                     xsi:noNamespaceSchemaLocation="http://www.jboss.org/ironjacamar/schema/
datasources_1_0.xsd">
  <datasource jndi-name="OracleDS" pool-name="OracleDS">
    <!--
      Here are a couple of the possible OCI configurations. For more information,
        see http://otn.oracle.com/docs/products/oracle9i/doc_library/release2/java.920/a96654/
toc.htm
      <connection-url>jdbc:oracle:oci:@youroracle-tns-name</connection-url> or
     <connection-url>jdbc:oracle:oci:@(description=(address=(host=youroraclehost)(protocol=tcp)
(port=1521))(connect_data=(SERVICE_NAME=yourservicename)))</connection-url>
      Clearly, its better to have TNS set up properly.
    -->
    <connection-url>jdbc:oracle:thin:@youroraclehost:1521:yoursid</connection-url>
    <driver-class>oracle.jdbc.driver.OracleDriver</driver-class>
    <security>
      <user-name>x</user-name>
      <password>y</password>
    </security>
    <validation>
                                                    <valid-connection-checker         class-
name="org.jboss.jca.adapters.jdbc.extensions.oracle.OracleValidConnectionChecker"></valid-
connection-checker>
                                                    <stale-connection-checker         class-
name="org.jboss.jca.adapters.jdbc.extensions.oracle.OracleStaleConnectionChecker"></stale-
connection-checker>
                                                              <exception-sorter          class-
name="org.jboss.jca.adapters.jdbc.extensions.oracle.OracleExceptionSorter"></exception-sorter>
    </validation>
  </datasource>
```

```
</datasources>
```

C.10. Oracle XA

```
      <?xml version="1.0" encoding="UTF-8"?>
<!-- ATTENTION: DO NOT FORGET TO SET Pad=true IN transaction.xml -->
<datasources xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                      xsi:noNamespaceSchemaLocation="http://www.jboss.org/ironjacamar/schema/
datasources_1_0.xsd">

  <xa-datasource jndi-name="XAOracleDS" pool-name="XAOracleDS">
    <xa-datasource-property name="URL">jdbc:oracle:oci8:@tc</xa-datasource-property>
    <xa-datasource-property name="User">scott</xa-datasource-property>
    <xa-datasource-property name="Password">tiger</xa-datasource-property>
    <xa-datasource-class>oracle.jdbc.xa.client.OracleXADataSource</xa-datasource-class>
    <xa-pool>
      <is-same-rm-override>false</is-same-rm-override>
      <!-- Uncomment to enable interleaving <interleaving/> -->
      <no-tx-separate-pools />
    </xa-pool>
    <validation>
                                                    <valid-connection-checker         class-
name="org.jboss.jca.adapters.jdbc.extensions.oracle.OracleValidConnectionChecker"></valid-
connection-checker>
                                                    <stale-connection-checker         class-
name="org.jboss.jca.adapters.jdbc.extensions.oracle.OracleStaleConnectionChecker"></stale-
connection-checker>
                                                              <exception-sorter          class-
name="org.jboss.jca.adapters.jdbc.extensions.oracle.OracleExceptionSorter"></exception-sorter>
    </validation>
  </xa-datasource>

</datasources>
```

C.11. Microsoft SQLServer

```
      <?xml version="1.0" encoding="UTF-8"?>
<!-- See http://www.jboss.org/community/wiki/Multiple1PC for information about datasource -->
<datasources xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                      xsi:noNamespaceSchemaLocation="http://www.jboss.org/ironjacamard/schema/
atasources_1_0.xsd">

  <datasource jndi-name="MSSQLDS" pool-name="MSSQLDS">
           <connection-url>jdbc:microsoft:sqlserver://localhost:1433;DatabaseName=MyDatabase</
connection-url>
```

```
    <driver-class>com.microsoft.sqlserver.jdbc.SQLServerDriver</driver-class>
    <security>
      <user-name>x</user-name>
      <password>y</password>
    </security>
    <validation>
                                                    <valid-connection-checker         class-
name="org.jboss.jca.adapters.jdbc.extensions.mssql.MSSQLValidConnectionChecker"></valid-
connection-checker>
    </validation>
  </datasource>

</datasources>
```

C.12. Microsoft SQLServer XA

```
      <?xml version="1.0" encoding="UTF-8"?>

<datasources xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                      xsi:noNamespaceSchemaLocation="http://www.jboss.org/ironjacamar/schema/
datasources_1_0.xsd">

  <xa-datasource jndi-name="MSSQLXADS" pool-name="MSSQLXADS">
    <xa-datasource-property name="ServerName">myserver</xa-datasource-property>
    <xa-datasource-property name="DatabaseName">mydatabase</xa-datasource-property>
    <xa-datasource-property name="SelectMethod">cursor</xa-datasource-property>
    <xa-datasource-property name="User">myuser</xa-datasource-property>
    <xa-datasource-property name="Password">mypassword</xa-datasource-property>
       <xa-datasource-class>com.microsoft.sqlserver.jdbc.SQLServerXADataSource</xa-datasource-
class>
    <xa-pool>
      <is-same-rm-override>false</is-same-rm-override>
      <!-- Uncomment to enable interleaving <interleaving/> -->
    </xa-pool>
    <validation>
                                                    <valid-connection-checker         class-
name="org.jboss.jca.adapters.jdbc.extensions.mssql.MSSQLValidConnectionChecker"></valid-
connection-checker>
    </validation>
  </xa-datasource>

</datasources>
```

C.13. IBM DB2

```
      <?xml version="1.0" encoding="UTF-8"?>
```

```
<datasources xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                     xsi:noNamespaceSchemaLocation="http://www.jboss.org/ironjacamar/schema/
datasources_1_0.xsd">

  <datasource jndi-name="DB2DS" pool-name="DB2DS">
    <!--
      DB2 Universal Driver Note connection URL is in form of
      jdbc:db2://host:port:dbname

      Default port for Type 4 driver is 50000

     Note, host and port must be specified if using Type 4 driver. And be forewarned, no native
     XA support is provided with Type 4; you must set a DB property calling for Type 2 to get XA

      <driver-class>com.ibm.db2.jcc.DB2Driver</driver-class>
      <connection-url>jdbc:db2://[hostname]:[port]/databasename"</connection-url>

      Please see http://www-128.ibm.com/developerworks/db2/library/techarticle/dm-0512kokkat/
      or the DB2 JDBC application developers manual.
     -->
    <connection-url>jdbc:db2:yourdatabase</connection-url>
    <driver-class>COM.ibm.db2.jdbc.app.DB2Driver</driver-class>
    <pool>
      <min-pool-size>0</min-pool-size>
      <max-pool-size>50</max-pool-size>
    </pool>
    <security>
      <user-name>x</user-name>
      <password>y</password>
    </security>
    <validation>
                                                    <valid-connection-checker         class-
name="org.jboss.jca.adapters.jdbc.extensions.db2.DB2ValidConnectionChecker"></valid-
connection-checker>
                                                    <stale-connection-checker         class-
name="org.jboss.jca.adapters.jdbc.extensions.db2.DB2StaleConnectionChecker"></stale-
connection-checker>
                                                              <exception-sorter          class-
name="org.jboss.jca.adapters.jdbc.extensions.db2.DB2ExceptionSorter"></exception-sorter>
    </validation>
  </datasource>

</datasources>
```

C.14. IBM DB2 XA

```
      <?xml version="1.0" encoding="UTF-8"?>
<datasources xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                     xsi:noNamespaceSchemaLocation="http://www.jboss.org//ironjacamar/schema/
datasources_1_0.xsd">

  <xa-datasource jndi-name="DB2XADS" pool-name="DB2XADS">
    <xa-datasource-property name="DatabaseName">yout_database_name</xa-datasource-property>
```

```
    <xa-datasource-property name="User">your_user</xa-datasource-property>
    <xa-datasource-property name="Password">your_password</xa-datasource-property>
    <xa-datasource-class>COM.ibm.db2.jdbc.DB2XADataSource</xa-datasource-class>
    <xa-pool>
      <is-same-rm-override>false</is-same-rm-override>
      <!-- Uncomment to enable interleaving <interleaving/> -->
    </xa-pool>
    <validation>
                                                    <valid-connection-checker          class-
name="org.jboss.jca.adapters.jdbc.extensions.db2.DB2ValidConnectionChecker"></valid-
connection-checker>
                                                    <stale-connection-checker          class-
name="org.jboss.jca.adapters.jdbc.extensions.db2.DB2StaleConnectionChecker"></stale-
connection-checker>
                                                              <exception-sorter          class-
name="org.jboss.jca.adapters.jdbc.extensions.db2.DB2ExceptionSorter"></exception-sorter>
    </validation>
  </xa-datasource>

</datasources>
```

Appendix D. Logging codes

D.1. Core: 000000 - 009999

Table D.1. Logging codes for core

Code	Level	Description
100	INFO	Closing a connection for you. Please close them yourself
102	INFO	Throwable trying to close a connection for you, please close it yourself
103	INFO	Could not find a close method on alleged connection object. Please close your own connections
151	EXCEPTION	Some connections were not closed, see the log for the allocation stacktraces
201	ERROR	SecurityContext setup failed
202	ERROR	SecurityContext setup failed since CallbackSecurity was null
251	EXCEPTION	SecurityContext setup failed
252	EXCEPTION	SecurityContext setup failed since CallbackSecurity was null
253	EXCEPTION	Work is null
254	EXCEPTION	StartTimeout is negative
255	EXCEPTION	Interrupted while requesting permit
256	EXCEPTION	Work execution context must be null because work instance implements WorkContextProviderStartTimeout is negative
257	EXCEPTION	Run method is synchronized
258	EXCEPTION	Release method is synchronized
259	EXCEPTION	Unsupported WorkContext class
260	EXCEPTION	Duplicate TransactionWorkContext class
261	EXCEPTION	Duplicate SecurityWorkContext class
262	EXCEPTION	Duplicate HintWorkContext class
263	EXCEPTION	WorkManager is shutting down
264	EXCEPTION	SecurityContext setup failed since CallbackSecurity::Domain was empty
265	EXCEPTION	ResourceAdapterAssociation failed
301	INFO	Registered a null handle for managed connection
302	INFO	Unregistered handle that was not registered
303	INFO	Unregistered a null handle for managed connection

Code	Level	Description
305	WARN	Connection error occured
306	WARN	Unknown connection error occured
307	WARN	Notified of error on a different managed connection
311	INFO	Throwable from unregister connection
312	ERROR	Error while closing connection handle
313	ERROR	There is something wrong with the pooling
351	EXCEPTION	Not correct type
352	EXCEPTION	Failure to delist resource
353	EXCEPTION	Error in delist
354	EXCEPTION	Unfinished local transaction - error getting local transaction
355	EXCEPTION	Unfinished local transaction but managed connection does not provide a local transaction
356	EXCEPTION	Failed to enlist
401	WARN	Error during tidy up connection
402	WARN	ResourceException in returning connection
403	WARN	Reconnecting a connection handle that still has a managed connection
404	WARN	Unchecked throwable in managedConnectionDisconnected()
451	EXCEPTION	The connection manager is shutdown
452	EXCEPTION	Method getManagedConnection retry wait was interrupted
453	EXCEPTION	Unable to get managed connection
454	EXCEPTION	You are trying to use a connection factory that has been shut down: ManagedConnectionFactory is null
455	EXCEPTION	Wrong ManagedConnectionFactory sent to allocateConnection
456	EXCEPTION	Unchecked throwable in ManagedConnection.getConnection()
457	EXCEPTION	Unchecked throwable in managedConnectionReconnected()
458	EXCEPTION	This method is not supported
459	EXCEPTION	Transaction is not active
460	EXCEPTION	Error checking for a transaction
461	EXCEPTION	Could not enlist in transaction on entering meta-aware object
462	EXCEPTION	Could not delist resource, probably a transaction rollback
463	EXCEPTION	Unable to set XAResource transaction timeout
501	WARN	Thread is not the enlisting thread
502	WARN	Transaction error in beforeCompletion

Code	Level	Description
503	WARN	Transaction error in afterCompletion
601	INFO	ConnectionValidator has been interrupted
602	WARN	ConnectionValidator ignored unexpected runtime exception
603	WARN	ConnectionValidator ignored unexpected error
604	WARN	Throwable while attempting to get a new connection
605	WARN	Destroying connection that could not be successfully matched
606	WARN	Throwable while trying to match managed connection, destroying connection
607	WARN	ResourceException cleaning up managed connection
608	WARN	Destroying returned connection, maximum pool size exceeded
609	WARN	Attempt to return connection twice
610	WARN	Unable to fill pool
611	WARN	Warning: Background validation was specified with a non compliant ManagedConnectionFactory interface
612	WARN	Destroying connection that could not be successfully matched
613	WARN	Throwable while trying to match managed connection, destroying connection
614	ERROR	Exception during createSubject()
615	WARN	Destroying active connection in pool
651	EXCEPTION	Unable to get managed connection pool
652	EXCEPTION	Unable to obtain lock
653	EXCEPTION	The pool has been shutdown
654	EXCEPTION	Interrupted while requesting connection
655	EXCEPTION	No managed connections available within configured blocking timeout
656	EXCEPTION	This should never happen
657	EXCEPTION	Interrupted while requesting permit
658	EXCEPTION	Unexpected throwable while trying to create a connection
701	WARN	Exception during unbind
751	EXCEPTION	Deployment failed
851	EXCEPTION	Resource adapter instance not active
852	EXCEPTION	Validation exception
853	EXCEPTION	The activation spec class is no longer available
854	EXCEPTION	The resource adapter is no longer available
855	EXCEPTION	Key isn't registered

Code	Level	Description
856	EXCEPTION	Unable to lookup resource adapter in MDR
901	WARN	Error during connection close
902	ERROR	Error during inflow crash recovery
903	ERROR	Error creating Subject for crash recovery
904	WARN	No security domain defined for crash recovery
905	WARN	Subject for crash recovery was null
906	ERROR	Error during crash recovery
951	EXCEPTION	Error during connection close
1001	WARN	No users.properties were found
1002	ERROR	Error while loading users.properties
1003	WARN	No roles.properties were found
1004	ERROR	Error while loading roles.properties
1005	WARN	No callback.properties were found
1006	ERROR	Error while loading callback.properties
1011	WARN	Prepare called on a local tx. Use of local transactions on a JTA transaction with more than one branch may result in inconsistent data in some cases of failure
1151	EXCEPTION	Trying to start a new transaction when old is not complete
1152	EXCEPTION	Trying to start a new transaction with wrong flags
1153	EXCEPTION	Error trying to start local transaction
1154	EXCEPTION	Throwable trying to start local transaction
1155	EXCEPTION	Wrong xid in commit
1156	EXCEPTION	Could not commit local transaction
1157	EXCEPTION	Forget not supported in local transaction
1158	EXCEPTION	No recovery for LocalTransaction only resource manager
1159	EXCEPTION	Wrong xid in rollback
1160	EXCEPTION	Could not rollback local transaction

D.2. Common: 010000 - 019999

Table D.2. Logging codes for common

Code	Level	Description
10001	ERROR	Parsing error of ra.xml file
10002	ERROR	Parsing error of ironjacamar.xml file

Code	Level	Description
10003	ERROR	No @Connector was found and no definition in the ra.xml metadata either
10004	ERROR	More than one @Connector was found but the correct one wasn't defined in the ra.xml metadata
10051	EXCEPTION	AnnotationRepository reference is null
10052	EXCEPTION	No @Connector defined
10053	EXCEPTION	More than @Connector defined
10054	EXCEPTION	More than one @ConnectionDefinitions defined
10055	EXCEPTION	Unknown annotation
10056	EXCEPTION	Element isn't a valid boolean
10057	EXCEPTION	Attribute isn't a valid boolean
10058	EXCEPTION	Element isn't a valid number
10059	EXCEPTION	Invalid flush strategy
10060	EXCEPTION	Unexpected end tag
10061	EXCEPTION	Unexpected element
10062	EXCEPTION	Reached end of xml document unexpectedly
10063	EXCEPTION	Mandatory class-name attribute missing
10064	EXCEPTION	Unexpected attribute
10065	EXCEPTION	Missing mandatory jndi-name attribute
10066	EXCEPTION	You cannot define more than one pool or xa-pool in same connection-definition
10067	EXCEPTION	Element cannot be set without an xa-pool
10068	EXCEPTION	Missing required attribute
10069	EXCEPTION	Missing required element
10070	EXCEPTION	Invalid negative value
10071	EXCEPTION	Tag is not valid
10072	EXCEPTION	Tag cannot be undefined
10073	EXCEPTION	Invalid <security> configuration
10074	EXCEPTION	The resource adapter metadata must be defined
10075	EXCEPTION	The resource adapter metadata must contain either an outbound or inbound configuration
10076	EXCEPTION	Tag must be defined
10077	EXCEPTION	Wrong annotation type

D.3. Deployers: 020000 - 029999

Table D.3. Logging codes for deployers

Code	Level	Description
20001	INFO	Required license terms
20002	INFO	Deployed
20003	WARN	Failure during validation report generation
20004	WARN	Only one connection definition found with a mismatch in class-name
20005	WARN	Only one admin object found with a mismatch in class-name
20006	ERROR	ConnectionFactory is null
20007	ERROR	Exception during createSubject()
20008	WARN	Invalid config property
20009	WARN	Invalid connection definition
20010	ERROR	Connection definition with missing class-name
20011	ERROR	Admin object with missing class-name
20012	WARN	Admin object not bound
20013	WARN	Connection factory not bound
20014	INFO	Admin object not spec compliant
20015	INFO	Connection factory not spec compliant
20016	WARN	Missing <recovery> element. XA recovery disabled
20051	EXCEPTION	Unable to start
20052	EXCEPTION	Unable to associate
20053	EXCEPTION	ManagedConnectionFactory must be defined in class-name
20054	EXCEPTION	AdminObject must be defined in class-name
20055	EXCEPTION	Failed to bind admin object
20056	EXCEPTION	Deployment failed
20057	EXCEPTION	Invalid ManagedConnectionFactory class
20058	EXCEPTION	Invalid ActivationSpec class
20059	EXCEPTION	Invalid ResourceAdapter class
20060	EXCEPTION	Unable to inject
20061	EXCEPTION	Invalid required work context
20062	EXCEPTION	Invalid connection factory interface
20063	EXCEPTION	Invalid connection factory implementation
20064	EXCEPTION	Invalid connection interface
20065	EXCEPTION	Invalid connection implementation

Code	Level	Description
20066	EXCEPTION	Connection factory implementation doesn't implement interface
20067	EXCEPTION	Connection implementation doesn't implement interface

Appendix E. Licenses

All licenses can be found in the `doc/licenses` directory.

E.1. GNU Lesser General Public License 2.1

GNU LESSER GENERAL PUBLIC LICENSE

Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc. 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts as the successor of the GNU Library Public License, version 2, hence the version number 2.1.]

E.1.1. Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages--typically libraries--of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method:

1. we copyright the library, and
2. we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the *Lesser* General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a “work based on the library” and a “work that uses the library”.

The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

E.1.2. Terms and Conditions for Copying, Distribution and Modification

E.1.2.1. Section 0

This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called “this License”). Each licensee is addressed as “you”.

A “library” means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The “Library”, below, refers to any such software library or work which has been distributed under these terms. A “work based on the Library” means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term “modification”.)

“Source code” for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

E.1.2.2. Section 1

You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

E.1.2.3. Section 2

You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of *Section 1* above, provided that you also meet all of these conditions:

a. The modified work must itself be a software library.

b. You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c. You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d. If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

   (For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, *Subsection 2d* requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

E.1.2.4. Section 3

You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

E.1.2.5. Section 4

You may copy and distribute the Library (or a portion or derivative of it, under *Section 2*) in object code or executable form under the terms of *Sections 1* and *2* above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of *Sections 1* and *2* above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

E.1.2.6. Section 5

A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a “work that uses the Library”. Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a “work that uses the Library” with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a “work that uses the library”. The executable is therefore covered by this License. *Section 6* states terms for distribution of such executables.

When a “work that uses the Library” uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under *Section 6*.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of *Section 6*. Any executables containing that work also fall under *Section 6*, whether or not they are linked directly with the Library itself.

E.1.2.7. Section 6

As an exception to the Sections above, you may also combine or link a “work that uses the Library” with the Library to produce a work containing portions of the Library, and distribute that work under

terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a. Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under *Sections 1* and *2* above); and, if the work is an executable linked with the Library, with the complete machine-readable “work that uses the Library”, as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b. Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c. Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in *Subsection 6a*, above, for a charge no more than the cost of performing this distribution.

d. If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e. Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the “work that uses the Library” must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

E.1.2.8. Section 7

You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined

library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a. Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b. Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

E.1.2.9. Section 8

You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

E.1.2.10. Section 9

You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

E.1.2.11. Section 10

Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

E.1.2.12. Section 11

If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

E.1.2.13. Section 12

If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

E.1.2.14. Section 13

The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and “any later version”, you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

E.1.2.15. Section 14

If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

E.1.2.16. NO WARRANTY Section 15

BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT

WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

E.1.2.17. Section 16

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

E.1.3. How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the “copyright” line and a pointer to where the full notice is found.

<one line to give the library's name and a brief idea of what it does.> Copyright (C) <year> <name of author>

This library is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a “copyright disclaimer” for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the library `Frob' (a library for tweaking knobs) written by James Random Hacker.

<signature of Ty Coon>, 1 April 1990 Ty Coon, President of Vice

That's all there is to it!

E.2. Creative Commons Attribution–Share Alike 3.0 Unported License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. TO THE EXTENT THIS LICENSE MAY BE CONSIDERED TO BE A CONTRACT, THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

E.2.1. Definitions

a. *"Adaptation"* means a work based upon the Work, or upon the Work and other pre-existing works, such as a translation, adaptation, derivative work, arrangement of music or other alterations of a literary or artistic work, or phonogram or performance and includes cinematographic adaptations or any other form in which the Work may be recast, transformed, or adapted including in any form recognizably derived from the original, except that a work that constitutes a Collection will not be considered an Adaptation for the purpose of this License. For the avoidance of doubt, where the Work is a musical work, performance or phonogram, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered an Adaptation for the purpose of this License.

b. *"Collection"* means a collection of literary or artistic works, such as encyclopedias and anthologies, or performances, phonograms or broadcasts, or other works or subject matter other than works listed in Section 1(f) below, which, by reason of the selection and arrangement of their contents, constitute intellectual creations, in which the Work is included in its entirety in unmodified form along with one or more other contributions, each constituting separate and independent works in themselves, which together are assembled into a collective whole. A work that constitutes a Collection will not be considered an Adaptation (as defined below) for the purposes of this License.

c. *"Creative Commons Compatible License"* means a license that is listed at http://creativecommons.org/compatiblelicenses that has been approved by Creative Commons as being essentially equivalent to this License, including, at a minimum, because that license: (i) contains terms that have the same purpose, meaning and effect as the License Elements of this License; and, (ii) explicitly permits the relicensing of adaptations of works made available under that license under this License or a Creative Commons jurisdiction license with the same License Elements as this License.

d. *"Distribute"* means to make available to the public the original and copies of the Work or Adaptation, as appropriate, through sale or other transfer of ownership.

e. *"License Elements"* means the following high-level license attributes as selected by Licensor and indicated in the title of this License: Attribution, ShareAlike.

f. *"Licensor"* means the individual, individuals, entity or entities that offer(s) the Work under the terms of this License.

g. *"Original Author"* means, in the case of a literary or artistic work, the individual, individuals, entity or entities who created the Work or if no individual or entity can be identified, the publisher; and in addition (i) in the case of a performance the actors, singers, musicians, dancers, and other persons who act, sing, deliver, declaim, play in, interpret or otherwise perform literary or artistic works or expressions of folklore; (ii) in the case of a phonogram the producer being the person or legal entity who first fixes the sounds of a performance or other sounds; and, (iii) in the case of broadcasts, the organization that transmits the broadcast.

h. *"Work"* means the literary and/or artistic work offered under the terms of this License including without limitation any production in the literary, scientific and artistic domain, whatever may be the mode or form of its expression including digital form, such as a book, pamphlet and other writing; a lecture, address, sermon or other work of the same nature; a dramatic or dramatico-musical work; a choreographic work or entertainment in dumb show; a musical composition with or without words; a cinematographic work to which are assimilated works expressed by a process analogous to cinematography; a work of drawing, painting, architecture, sculpture, engraving or lithography; a photographic work to which are assimilated works expressed by a process analogous to photography; a work of applied art; an illustration, map, plan, sketch or three-dimensional work relative to geography, topography, architecture or science; a performance; a broadcast; a phonogram; a compilation of data to the extent it is protected as a copyrightable work; or a work performed by a variety or circus performer to the extent it is not otherwise considered a literary or artistic work.

i. *"You"* means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

j. *"Publicly Perform"* means to perform public recitations of the Work and to communicate to the public those public recitations, by any means or process, including by wire or wireless means or public digital performances; to make available to the public Works in such a way that members of the public may access these Works from a place and at a place individually chosen by them;

to perform the Work to the public by any means or process and the communication to the public of the performances of the Work, including by public digital performance; to broadcast and rebroadcast the Work by any means including signs, sounds or images.

k. *"Reproduce"* means to make copies of the Work by any means including without limitation by sound or visual recordings and the right of fixation and reproducing fixations of the Work, including storage of a protected performance or phonogram in digital form or other electronic medium.

E.2.2. Fair Dealing Rights

Nothing in this License is intended to reduce, limit, or restrict any uses free from copyright or rights arising from limitations or exceptions that are provided for in connection with the copyright protection under copyright law or other applicable laws.

E.2.3. License Grant

Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

a. to Reproduce the Work, to incorporate the Work into one or more Collections, and to Reproduce the Work as incorporated in the Collections;

b. to create and Reproduce Adaptations provided that any such Adaptation, including any translation in any medium, takes reasonable steps to clearly label, demarcate or otherwise identify that changes were made to the original Work. For example, a translation could be marked "The original work was translated from English to Spanish," or a modification could indicate "The original work has been modified.";

c. to Distribute and Publicly Perform the Work including as incorporated in Collections; and,

d. to Distribute and Publicly Perform Adaptations.

e. For the avoidance of doubt:

   i. *Non-waivable Compulsory License Schemes.* In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme cannot be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License;

   ii. *Waivable Compulsory License Schemes.* In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme can be waived, the Licensor waives the exclusive right to collect such royalties for any exercise by You of the rights granted under this License; and,

   iii. *Voluntary License Schemes.* The Licensor waives the right to collect royalties, whether individually or, in the event that the Licensor is a member of a collecting society that

administers voluntary licensing schemes, via that society, from any exercise by You of the rights granted under this License.

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. Subject to Section 8(f), all rights not expressly granted by Licensor are hereby reserved.

E.2.4. Restrictions

The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

a. You may Distribute or Publicly Perform the Work only under the terms of this License. You must include a copy of, or the Uniform Resource Identifier (URI) for, this License with every copy of the Work You Distribute or Publicly Perform. You may not offer or impose any terms on the Work that restrict the terms of this License or the ability of the recipient of the Work to exercise the rights granted to that recipient under the terms of the License. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties with every copy of the Work You Distribute or Publicly Perform. When You Distribute or Publicly Perform the Work, You may not impose any effective technological measures on the Work that restrict the ability of a recipient of the Work from You to exercise the rights granted to that recipient under the terms of the License. This Section 4(a) applies to the Work as incorporated in a Collection, but this does not require the Collection apart from the Work itself to be made subject to the terms of this License. If You create a Collection, upon notice from any Licensor You must, to the extent practicable, remove from the Collection any credit as required by Section 4(c), as requested. If You create an Adaptation, upon notice from any Licensor You must, to the extent practicable, remove from the Adaptation any credit as required by Section 4(c), as requested.

b. You may Distribute or Publicly Perform an Adaptation only under the terms of: (i) this License; (ii) a later version of this License with the same License Elements as this License; (iii) a Creative Commons jurisdiction license (either this or a later license version) that contains the same License Elements as this License (e.g., Attribution-ShareAlike 3.0 US)); (iv) a Creative Commons Compatible License. If you license the Adaptation under one of the licenses mentioned in (iv), you must comply with the terms of that license. If you license the Adaptation under the terms of any of the licenses mentioned in (i), (ii) or (iii) (the "Applicable License"), you must comply with the terms of the Applicable License generally and the following provisions: (I) You must include a copy of, or the URI for, the Applicable License with every copy of each Adaptation You Distribute or Publicly Perform; (II) You may not offer or impose any terms on the Adaptation that restrict the terms of the Applicable License or the ability of the recipient of the Adaptation to exercise the rights granted to that recipient under the terms of the Applicable License; (III) You must keep intact all notices that refer to the Applicable License and to the disclaimer of warranties with every copy of the Work as included in the Adaptation You Distribute or Publicly Perform; (IV) when You Distribute or Publicly Perform the Adaptation, You

may not impose any effective technological measures on the Adaptation that restrict the ability of a recipient of the Adaptation from You to exercise the rights granted to that recipient under the terms of the Applicable License. This Section 4(b) applies to the Adaptation as incorporated in a Collection, but this does not require the Collection apart from the Adaptation itself to be made subject to the terms of the Applicable License.

c. If You Distribute, or Publicly Perform the Work or any Adaptations or Collections, You must, unless a request has been made pursuant to Section 4(a), keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or if the Original Author and/or Licensor designate another party or parties (e.g., a sponsor institute, publishing entity, journal) for attribution ("Attribution Parties") in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; (ii) the title of the Work if supplied; (iii) to the extent reasonably practicable, the URI, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and (iv) , consistent with Section 3(b), in the case of an Adaptation, a credit identifying the use of the Work in the Adaptation (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). The credit required by this Section 4(c) may be implemented in any reasonable manner; provided, however, that in the case of a Adaptation or Collection, at a minimum such credit will appear, if a credit for all contributing authors of the Adaptation or Collection appears, then as part of these credits and in a manner at least as prominent as the credits for the other contributing authors. For the avoidance of doubt, You may only use the credit required by this Section for the purpose of attribution in the manner set out above and, by exercising Your rights under this License, You may not implicitly or explicitly assert or imply any connection with, sponsorship or endorsement by the Original Author, Licensor and/or Attribution Parties, as appropriate, of You or Your use of the Work, without the separate, express prior written permission of the Original Author, Licensor and/or Attribution Parties.

d. Except as otherwise agreed in writing by the Licensor or as may be otherwise permitted by applicable law, if You Reproduce, Distribute or Publicly Perform the Work either by itself or as part of any Adaptations or Collections, You must not distort, mutilate, modify or take other derogatory action in relation to the Work which would be prejudicial to the Original Author's honor or reputation. Licensor agrees that in those jurisdictions (e.g. Japan), in which any exercise of the right granted in Section 3(b) of this License (the right to make Adaptations) would be deemed to be a distortion, mutilation, modification or other derogatory action prejudicial to the Original Author's honor and reputation, the Licensor will waive or not assert, as appropriate, this Section, to the fullest extent permitted by the applicable national law, to enable You to reasonably exercise Your right under Section 3(b) of this License (right to make Adaptations) but not otherwise.

E.2.5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE,

INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTIBILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

E.2.6. Termination

a. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Adaptations or Collections from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.

b. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

E.2.7. Miscellaneous

a. Each time You Distribute or Publicly Perform the Work or a Collection, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.

b. Each time You Distribute or Publicly Perform an Adaptation, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.

c. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

d. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

e. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

f. The rights granted under, and the subject matter referenced, in this License were drafted utilizing the terminology of the Berne Convention for the Protection of Literary and Artistic Works (as amended on September 28, 1979), the Rome Convention of 1961, the WIPO Copyright Treaty of 1996, the WIPO Performances and Phonograms Treaty of 1996 and the Universal Copyright Convention (as revised on July 24, 1971). These rights and subject matter take effect in the relevant jurisdiction in which the License terms are sought to be enforced according to the corresponding provisions of the implementation of those treaty provisions in the applicable national law. If the standard suite of rights granted under applicable copyright law includes additional rights not granted under this License, such additional rights are deemed to be included in the License; this License is not intended to restrict the license of any rights under applicable law.

E.3. Apache License, Version 2.0

Apache license

Version 2.0, January 2004

http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

E.3.1. Definitions

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

E.3.2. Grant of Copyright License

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

E.3.3. Grant of Patent License

Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

E.3.4. Redistribution

You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

a. You must give any other recipients of the Work or Derivative Works a copy of this License; and

b. You must cause any modified files to carry prominent notices stating that You changed the files; and

c. You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

d. If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

E.3.5. Submission of Contributions

Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

E.3.6. Trademarks

This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

E.3.7. Disclaimer of Warranty

Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

E.3.8. Limitation of Liability

In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

E.3.9. Accepting Warranty or Additional Liability

While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.