

geronimo-connector-1.1.xsd

{scrollbar}top

Introduction

Geronimo includes a connector container supporting both in bound and out bound Resource adapters. A Connector RAR may include Geronimo-specific configuration information in addition to the standard J2EE deployment descriptor. This is the Geronimo specific XML Schema Documentation for the Connector 1.5 deployment descriptor. The deployment descriptor must be named "META-INF/ra.xml" in the connector's RAR file.

This article is organized into the following sections : -

- [#Geronimo Connector Schema Document](#)
- [#Schema Document Properties](#)
- [#Global Declarations](#)
 - [Element: connector](#)
- [#Global Definitions](#)
 - [Complex Type: adminobject-instanceType](#)
 - [Complex Type: adminobjectType](#)
 - [Complex Type: config-property-settingType](#)
 - [Complex Type: connection-definitionType](#)
 - [Complex Type: connectiondefinition-instanceType](#)
 - [Complex Type: connectionmanagerType](#)
 - [Complex Type: connectorType](#)
 - [Complex Type: descriptionType](#)
 - [Complex Type: emptyType](#)
 - [Complex Type: outbound-resourceadapterType](#)
 - [Complex Type: partitionedpoolType](#)
 - [Complex Type: resourceadapter-instanceType](#)
 - [Complex Type: resourceadapterType](#)
 - [Complex Type: singlepoolType](#)
 - [Complex Type: xatransactionType](#)
 - [Simple Type: fully-qualified-classType](#)
- [#Legend](#)
- [#Glossary](#)

Geronimo Connector Schema Document

```
xmlsolidgeronimo-connector-1.1.xsd <?xml version="1.0" encoding="UTF-8"?> <!-- Copyright 2004 The Apache Software Foundation Licensed under the
Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0 Unless required by applicable law or agreed to in writing, software distributed under the License is
distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific
language governing permissions and limitations under the License. --> <xs:schema xmlns="http://www.w3.org/2001/XMLSchema" xmlns:ger="
http://geronimo.apache.org/xml/ns/j2ee/connector-1.1" xmlns:naming="http://geronimo.apache.org/xml/ns/naming-1.1" xmlns:sys="http://geronimo.apache.
org/xml/ns/deployment-1.1" targetNamespace="http://geronimo.apache.org/xml/ns/j2ee/connector-1.1" xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified"> <xs:import namespace="http://geronimo.apache.org/xml/ns/naming-1.1"
schemaLocation="geronimo-naming-1.1.xsd"/> <xs:import namespace="http://geronimo.apache.org/xml/ns/deployment-1.1" schemaLocation="geronimo-
module-1.1.xsd"/> <xs:import namespace="http://www.w3.org/XML/1998/namespace" schemaLocation="http://www.w3.org/2001/xml.xsd"/> <xs:
annotation> <xs:documentation> <![CDATA[ documents using this schema should start like: <connector xmlns="http://geronimo.apache.org/xml/ns/j2ee
/connector-1.1" version="1.5"> @(&#geronimo-connector_1_5.xsds ]> </xs:documentation> </xs:annotation> <!--
***** --> <xs:element name="connector" type="ger:connectorType"> <xs:annotation> <xs:documentation> The
connector element is the root element of the deployment descriptor for the resource adapter. This element includes <xs:documentation> </xs:annotation>
<xs:unique name="resourceadapter-instance-name-uniqueness"> <xs:annotation> <xs:documentation> The resourceadapter-instance element must have
a unique name. </xs:documentation> </xs:annotation> <xs:selector xpath="ger:resourceadapter/ger:resourceadapter-instance"/> <xs:field xpath="ger:
resourceadapter-name"/> </xs:unique> <xs:unique name="connectiondefinition-instance-name-uniqueness"> <xs:annotation> <xs:documentation> The
connectiondefinition-instance element must have a unique name. </xs:documentation> </xs:annotation> <xs:selector xpath="ger:resourceadapter/ger:
outbound-resourceadapter/ger:connection-definition/ ger:connectiondefinition-instance"/> <xs:field xpath="ger:name"/> </xs:unique> <xs:unique name="
admin-object-instance-name-uniqueness"> <xs:annotation> <xs:documentation> The admin-object-instance element must have a unique name. </xs:
documentation> </xs:annotation> <xs:selector xpath="ger:adminobject/ger:adminobject-instance"/> <xs:field xpath="ger:message-destination-name"/> <
/xs:unique> </xs:element> <!-- ***** --> <xs:complexType name="adminobjectType"> <xs:annotation> <xs:
documentation> reference back into standard dd </xs:documentation> </xs:annotation> <xs:sequence> <xs:element name="adminobject-interface" type="
ger:fully-qualified-classType"> <xs:annotation> <xs:documentation> believe it or not, there are no uniqueness constraints on admin object intf or class </xs:
documentation> </xs:annotation> </xs:element> <xs:element name="adminobject-class" type="ger:fully-qualified-classType"> <xs:annotation> <xs:
documentation>reference back into the standard dd</xs:documentation> </xs:annotation> </xs:element> <xs:element name="adminobject-instance" type="
ger:adminobject-instanceType" minOccurs="0" maxOccurs="unbounded"/> </xs:sequence> </xs:complexType> <xs:complexType name="adminobject-
instanceType"> <xs:sequence> <xs:element name="message-destination-name" type="xs:string"> <xs:annotation> <xs:documentation>unique name for
admin object instance. Used in object name of gbean for it. Matches message-destination-name in ejb-jar.xml assembly descriptor. </xs:documentation> <
/xs:annotation> </xs:element> <xs:element name="config-property-setting" type="ger:config-property-settingType" minOccurs="0" maxOccurs="
unbounded"/> </xs:sequence> </xs:complexType> <!-- ***** --> <xs:complexType name="config-property-
settingType"> <xs:annotation> <xs:documentation> The config-property-settingType contains a name-value pair for a config-property or required-config-
property </xs:documentation> </xs:annotation> <xs:simpleContent> <xs:extension base="xs:string"> <xs:attribute name="name" type="xs:string" use="
required"/> </xs:extension> </xs:simpleContent> </xs:complexType> <!-- ***** --> <xs:complexType name="
connection-definitionType"> <xs:annotation> <xs: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. The name element uniquely identifies this instance </xs:documentation> </xs:annotation> <xs:sequence> <xs:element name="connectionfactory-
interface" type="ger:fully-qualified-classType"> <xs:annotation> <xs:documentation> connectionfactory-interface is unique in ra.xml so is used to index into
ra.xml </xs:documentation> </xs:annotation> </xs:element> <xs:element name="connectiondefinition-instance" type="ger:connectiondefinition-
instanceType" minOccurs="0" maxOccurs="unbounded"/> </xs:sequence> </xs:complexType> <xs:complexType name="connectiondefinition-
```

```

instanceType"> <xs:sequence> <xs:element name="name" type="xs:string"/> <xs:element name="implemented-interface" type="ger:fully-qualified-
classType" minOccurs="0" maxOccurs="unbounded"/> <xs:annotation> <xs:documentation> Additional interfaces implemented by the connectionfactory,
used by the application </xs:documentation> </xs:annotation> </xs:element> <xs:element name="config-property-setting" type="ger:config-property-
settingType" minOccurs="0" maxOccurs="unbounded"/> <!-- <xs:choice>--> <xs:element name="connectionmanager" type="ger:connectionmanagerType"
/> </xs:element> <!-- <xs:element name="connectionmanager-ref" type="xs:string"/>--> <!-- </xs:choice>--> </xs:sequence> </xs:complexType> <!--
***** --> <xs:complexType name="connectionmanagerType"> <xs:annotation> <xs:documentation> The
ConnectionManager configuration. </xs:documentation> </xs:annotation> <xs:sequence> <xs:element name="container-managed-security" type="ger:
emptyType" minOccurs="0"/> <xs:choice minOccurs="0"> <xs:element name="no-transaction" type="ger:emptyType"/> <xs:element name="local-
transaction" type="ger:emptyType"/> <xs:element name="xa-transaction" type="ger:xatransactionType"/> <xs:element name="transaction-log" type="ger:
emptyType"/> </xs:choice> <xs:element name="no-pool" type="ger:emptyType"/> <xs:element name="single-pool" type="ger:singlepoolType"
/> <xs:element name="partitioned-pool" type="ger:partitionedpoolType"/> </xs:choice> </xs:sequence> </xs:complexType> <xs:complexType name="
xatransactionType"> <xs:sequence> <xs:element name="transaction-caching" type="ger:emptyType" minOccurs="0"/> <xs:element name="thread-
caching" type="ger:emptyType" minOccurs="0"/> </xs:sequence> </xs:complexType> <xs:complexType name="singlepoolType"> <xs:sequence> <xs:
element name="max-size" type="xs:int" minOccurs="0"/> <xs:element name="min-size" type="xs:int" minOccurs="0"/> <xs:element name="blocking-
timeout-milliseconds" type="xs:int" minOccurs="0"/> <xs:element name="idle-timeout-minutes" type="xs:int" minOccurs="0"/> <xs:choice> <xs:element
name="match-one" type="ger:emptyType"/> <xs:element name="match-all" type="ger:emptyType"/> <xs:element name="select-one-assume-match" type="
ger:emptyType"/> </xs:choice> </xs:sequence> </xs:complexType> <xs:complexType name="partitionedpoolType"> <complexContent> <extension
base="ger:singlepoolType"> <xs:sequence> <xs:element name="partition-by-subject" type="ger:emptyType" minOccurs="0"/> <xs:element name="
partition-by-connectionrequestinfo" type="ger:emptyType" minOccurs="0"/> </xs:sequence> </extension> </complexContent> </xs:complexType> <!--
***** --> <xs:complexType name="connectorType"> <xs:annotation> <xs:documentation> The connectorType
defines a resource adapter. </xs:documentation> </xs:annotation> <xs:sequence> <xs:element ref="sys:environment" minOccurs="0"/> <xs:element
name="resourceadapter" type="ger:resourceadapterType" maxOccurs="unbounded"/> <xs:element name="adminobject" type="ger:adminobjectType"
minOccurs="0" maxOccurs="unbounded"/> <xs:element ref="sys:gbbean" minOccurs="0" maxOccurs="unbounded"/> </xs:sequence> </xs:complexType>
<!-- ***** --> <xs:complexType name="emptyType"/> <!-- ***** --> <xs:
complexType name="outbound-resourceadapterType"> <xs:annotation> <xs: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. </xs:documentation> </xs:annotation> <xs:sequence> <xs:element name="connection-definition" type="ger:connection-
definitionType" maxOccurs="unbounded"/> </xs:sequence> </xs:complexType> <!-- ***** --> <xs:complexType
name="resourceadapterType"> <xs:sequence> <xs:element name="resourceadapter-instance" type="ger:resourceadapter-instanceType" minOccurs="0">
</xs:element> <xs:element name="outbound-resourceadapter" type="ger:outbound-resourceadapterType" minOccurs="0"> </xs:element> </xs:sequence>
</xs:complexType> <xs:complexType name="resourceadapter-instanceType"> <xs:annotation> <xs:documentation> The resourceadapter instance info is
put in a separate optional element to provide easier more consistent support for 1.0 adapters. </xs:documentation> </xs:annotation> <xs:sequence> <xs:
element name="resourceadapter-name" type="xs:string"/> <xs:element name="config-property-setting" type="ger:config-property-settingType"
minOccurs="0" maxOccurs="unbounded"/> <xs:element ref="naming:workmanager"/> </xs:sequence> </xs:complexType> <!-- types borrowed from the
j2ee schema. We might want to centralize these --> <xs:simpleType name="fully-qualified-classType"> <xs:restriction base="xs:string"/> </xs:simpleType>
<xs:complexType name="descriptionType"> <xs:simpleContent> <xs:extension base="xs:string"> <xs:attribute ref="xml:lang"/> </xs:extension> </xs:
simpleContent> </xs:complexType> </xs:schema>

```

Schema Document Properties

Target Namespace	http://geronimo.apache.org/xml/ns/j2ee/connector-1.1 Geronimo Connector deployment descriptors must indicate the connector resource adapter schema by using the above Geronimo specific J2EE namespace.
Element and Attribute Namespaces	- Global element and attribute declarations belong to this schema's target namespace. - By default, local element declarations belong to this schema's target namespace. - By default, local attribute declarations have no namespace.
Schema Composition	This schema imports schema(s) from the following namespace(s): http://www.w3.org/XML/1998/namespace (at http://www.w3.org/2001/xml.xsd) http://geronimo.apache.org/xml/ns/naming-1.1 (at geronimo-naming-1.1.xsd) http://geronimo.apache.org/xml/ns/deployment-1.1 (at geronimo-module-1.1.xsd)
Documentation	documents using this schema should start as : <connectorxmlns="http://geronimo.apache.org/xml/ns/j2ee/connector-1.1" version="1.5"> @(#)geronimo-connector_1_5.xsds

Declared Namespaces

Prefix	Namespace
Default namespace	http://www.w3.org/2001/XMLSchema
xml	http://www.w3.org/XML/1998/namespace
ger	http://geronimo.apache.org/xml/ns/j2ee/connector-1.1
naming	http://geronimo.apache.org/xml/ns/naming-1.1
sys	http://geronimo.apache.org/xml/ns/deployment-1.1
xs	http://www.w3.org/2001/XMLSchema

xmlSolidSchema Component Representation <xs:schema targetNamespace="http://geronimo.apache.org/xml/ns/j2ee/connector-1.1"
elementFormDefault="qualified" attributeFormDefault="unqualified"> <xs:import namespace="http://geronimo.apache.org/xml/ns/naming-1.1"

```

schemaLocation="geronimo-naming-1.1.xsd"/> <xs:import namespace="http://geronimo.apache.org/xml/ns/deployment-1.1" schemaLocation="geronimo-
module-1.1.xsd"/> <xs:import namespace="http://www.w3.org/XML/1998/namespace" schemaLocation="http://www.w3.org/2001/xml.xsd"/> </xs:schema>

```

Global Declarations

Element : connectorconnector

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 resource adapter.

Element:	connector
Name	connector
Type	ger:connectorType
Nilable	No
Abstract	No
Documentation	The connector element is the root element of the deployment descriptor for the resource adapter. This element includes general information - XML Instance Representation - Schema Component Representation Both of which are depicted below.

```

xmlsolidXML Instance Representation: element - connector <ger:connector moduleId=" xs:string [1]" groupId=" xs:string [0..1]" suppressDefaultParentId="
xs:boolean [0..1]"> <!-- Uniqueness Constraint - resourceadapter-instance-name-uniqueness Selector - ger:resourceadapter/ger:resourceadapter-instance
Field(s) - ger:resourceadapter-name --> <!-- Uniqueness Constraint - connectiondefinition-instance-name-uniqueness Selector - ger:resourceadapter/ger:
outbound-resourceadapter/ger:connection-definition/ ger:connectiondefinition-instance Field(s) - ger:message-destination-name --> <sys:import> ... </sys:
import> [0..*] <sys:dependency> ... </sys:dependency> [0..*] <ger:resourceadapter> ger:resourceadapterType </ger:resourceadapter> [1..*] <ger:
adminobject> ger:adminobjectType </ger:adminobject> [0..*] <sys:gbean> ... </sys:gbean> [0..*] </ger:connector> xmlsolidSchema Component
Representation: element - connector <element name="connector" type="ger:connectorType"> <xs:annotation> <xs:documentation> The connector
element is the root element of the deployment descriptor for the resource adapter. This element includes </xs:documentation> </xs:annotation> <xs:
unique name="resourceadapter-instance-name-uniqueness"> <xs:annotation> <xs:documentation> The resourceadapter-instance element must have a
unique name. </xs:documentation> </xs:annotation> <xs:selector xpath="ger:resourceadapter/ger:resourceadapter-instance"> <xs:field xpath="ger:
resourceadapter-name"> </xs:unique> <xs:unique name="connectiondefinition-instance-name-uniqueness"> <xs:annotation> <xs:documentation> The
connectiondefinition-instance element must have a unique name. </xs:documentation> </xs:annotation> <xs:selector xpath="ger:resourceadapter/ger:
outbound-resourceadapter/ger:connection-definition/ ger:connectiondefinition-instance"> <xs:field xpath="ger:name"> </xs:unique> <xs:unique name="
admin-object-instance-name-uniqueness"> <xs:annotation> <xs:documentation> The admin-object-instance element must have a unique name. </xs:
documentation> </xs:annotation> <xs:selector xpath="ger:adminobject/ger:adminobject-instance"> <xs:field xpath="ger:message-destination-name"> <
/xs:unique> </element>

```

Global Definitions

Complex Type : adminobject-instanceTypeinstanceType

Complex Type:	adminobject-instanceType
Super-types:	None
Sub-types:	None
Name	adminobject-instanceType
Abstract	no

```

xmlsolidXML Instance Representation: adminobject-instanceType <...> <ger:message-destination-name> xs:string </ger:message-destination-name> [1] ?
<ger:config-property-setting> ger:config-property-settingType </ger:config-property-setting> [0..*] </...> xmlsolidSchema Component Representation:
adminobject-instanceType <complexType name="adminobject-instanceType"> <sequence> <element name="message-destination-name" type="xs:string"
> <annotation> <!-- <documentation>unique name for admin object instance.Used in object name of gbean for it. Matches message-destination-name in
ejb-jar.xml assembly descriptor. </documentation>--> </annotation> </element>

```

Complex Type: adminobjectTypeadminobjectType

The adminobjectType specifies information about an administered object. Administered objects are specific to a messaging style or message provider in Geronimo connector xml schema.

Complex Type:	adminobjectType
Super-types:	None
Sub-types:	None

Name	adminobjectType
------	-----------------

Abstract	no
Documentation	reference back into standard dd

xmlsolidXML Instance Representation : adminobjectType <...> <ger:adminobject-interface> ger:fully-qualified-classType </ger:adminobject-interface> [1] ? <ger:adminobject-class> ger:fully-qualified-classType </ger:adminobject-class> [1] ? <ger:adminobject-instance> ger:adminobject-instanceType </ger:adminobject-instance> [0..*] </...> xmlsolidSchema Component Representation: adminobjectType <complexType name="adminobjectType"> <sequence> <element name="adminobject-interface" type=" ger:fully-qualified-classType "/> <element name="adminobject-class" type=" ger:fully-qualified-classType " /> <element name="adminobject-instance" type=" ger:adminobject-instanceType " minOccurs="0" maxOccurs="unbounded"/> </sequence> </complexType>

Complex Type: config-property-settingTypesettingType

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

Complex Type:	config-property-settingType
Super-types:	xs:string < config-property-settingType (by extension)
Sub-types:	None

Name	config-property-settingType
Abstract	no
Documentation	The config-property-settingType contains a name-value pair for a config-property or required-config-property

xmlsolidXML Instance Representation: config-property-settingType <... name=" xs:string [0..1]"> xs:string </...> xmlsolidSchema Component Representation: config-property-settingType <complexType name="config-property-settingType"> <simpleContent> <extension base=" xs:string "> <attribute name="name" type=" xs:string "/> </extension> </simpleContent> </complexType>

Complex Type: connection-definitionTypedefinitionType

Complex Type:	connection-definitionType
Super-types:	None
Sub-types:	None

Name	connection-definitionType
Abstract	no
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. The name element uniquely identifies this instance

xmlsolidXML Instance Representation: connection-definitionType <...> <ger:connectionfactory-interface> ger:fully-qualified-classType </ger:connectionfactory-interface> [1] ? <ger:connectiondefinition-instance> ger:connectiondefinition-instanceType </ger:connectiondefinition-instance> [0..*] </...> xmlsolidSchema Component Representation: connection-definitionType <complexType name="connection-definitionType"> <sequence> <element name="connectionfactory-interface" type=" ger:fully-qualified-classType "/> <element name="connectiondefinition-instance" type=" ger:connectiondefinition-instanceType " minOccurs="0" maxOccurs="unbounded"/> </sequence> </complexType>

Complex Type: connectiondefinition-instanceTypedefinitioninstanceType

Complex Type:	connectiondefinition-instanceType
Super-types:	None
Sub-types:	None

Name	connectiondefinition-instanceType
Abstract	No

xmlsolidXML Instance Representation: connectiondefinition-instanceType <...> <ger:name> xs:string </ger:name> [1] <ger:implemented-interface> ger:fully-qualified-classType </ger:implemented-interface> [0..*] ? <ger:config-property-setting> ger:config-property-settingType </ger:config-property-setting> [0..*] <ger:connectionmanager> ger:connectionmanagerType </ger:connectionmanager> [1] </...> xmlsolidSchema Component Representation: connectiondefinition-instanceType <complexType name="connectiondefinition-instanceType"> <sequence> <element name="name" type=" xs:string "/> <element name="implemented-interface" type=" ger:fully-qualified-classType " minOccurs="0" maxOccurs="unbounded"/> <element name="config-property-setting" type=" ger:config-property-settingType " minOccurs="0" maxOccurs="unbounded"/> <-- <xs:choice>--> <element name="connectionmanager" type=" ger:connectionmanagerType "/> <-- <xs:element name="connectionmanager-ref" type="xs:string"/>--> <-- </xs:choice>--> </sequence> </complexType>

Complex Type: connectionmangerTypeconnectionmanagerType

Complex Type:	connectionmanagerType
Super-types:	None
Sub-types:	None

Name	connectionmanagerType
Abstract	no
Documentation	The ConnectionManager configuration

xmlsolidXML Instance Representation: connectionmanagerType <...> <ger:container-managed-security> ger:emptyType </ger:container-managed-security> [0..1] Start Choice [0..1] <ger:no-transaction> ger:emptyType </ger:no-transaction> [1] <ger:local-transaction> ger:emptyType </ger:local-transaction> [1] <ger:xa-transaction> ger:xatransactionType </ger:xa-transaction> [1] <ger:transaction-log> ger:emptyType </ger:transaction-log> [1] End Choice Start Choice [1] <ger:no-pool> ger:emptyType </ger:no-pool> [1] <ger:single-pool> ger:singlepoolType </ger:single-pool> [1] <ger:partitioned-pool> ger:partitionedpoolType </ger:partitioned-pool> [1] End Choice </...> xmlsolidSchema Component Representation: connectionmanagerType <complexType name="connectionmanagerType"> <sequence> <element name="container-managed-security" type=" ger:emptyType " minOccurs="0"/> <choice minOccurs="0"> <element name="no-transaction" type=" ger:emptyType "/> <element name="local-transaction" type=" ger:emptyType "/> <element name="xa-transaction" type=" ger:xatransactionType "/> <element name="transaction-log" type=" ger:emptyType "/> </choice> <choice> <element name="no-pool" type=" ger:emptyType "/> <element name="single-pool" type=" ger:singlepoolType "/> <element name="partitioned-pool" type=" ger:partitionedpoolType "/> </choice> </sequence> </complexType>

Complex Type: connectorTypeconnectorType

The connectorType defines a resource adapter

Complex Type:	connectorType
Super-types:	None
Sub-types:	None

Name	connectorType
Abstract	no
Documentation	The connectorType defines a resource adapter

xmlsolidXML Instance Representation: connectorType <... <sys:environment>..</sys:enviornment> <sys:import> ... </sys:import> [0..*] <sys:dependency> ... </sys:dependency> [0..*] <ger:resourceadapter> ger:resourceadapterType </ger:resourceadapter> [1..*] <ger:adminobject> ger:adminobjectType </ger:adminobject> [0..*] <sys:gbean> ... </sys:gbean> [0..*] </...> xmlsolidSchema Component Representation: connectorType <complexType name="connectorType"> <sequence> <element ref="sys:environment" minOccurs="0"/> <element name="resourceadapter" type="ger:resourceadapterType" maxOccurs="unbounded"/> <element name="adminobject" type="ger:adminobjectType" minOccurs="0" maxOccurs="unbounded"/> <element ref="sys:gbean" minOccurs="0" maxOccurs="unbounded"/> </sequence> </complexType>

Complex Type: descriptionTypedescriptionType

Complex Type:	descriptionType
Super-types:	xs:string < descriptionType (by extension)
Sub-types:	None

Name	descriptionType
Abstract	no

xmlsolidXML Instance Representation: descriptionType <... xml:lang="[0..1]"> xs:string </...> xmlsolidSchema Component Representation: descriptionType <complexType name="descriptionType"> <simpleContent> <extension base=" xs:string "> <attribute ref="xml:lang"/> </extension> </simpleContent> </complexType>

Complex Type: emptyTypeemptyType

Complex Type:	emptyType
Super-types:	None
Sub-types:	None

Name	emptyType
------	-----------

Abstract	no
----------	----

xmlsolidXML Instance Representation: emptyType <.../> xmlsolidSchema Component Representation: emptyType <complexType name="emptyType"/>

Complex Type: outbound-resourceadapterTyperesourceadapterType

Complex Type:	outbound-resourceadapterType
Super-types:	None
Sub-types:	None

Name	outbound-resourceadapterType
Abstract	no
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 security permissions required. 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. In this case, the application server ignores the security part of the system contracts.

xmlsolidXML Instance Representation: outbound-resourceadapterType <...> <ger:connection-definition> ger:connection-definitionType </ger:connection-definition> [1..*] </...> xmlsolidSchema Component Representation: outbound-resourceadapterType <complexType name="outbound-resourceadapterType"> <sequence> <element name="connection-definition" type=" ger:connection-definitionType " maxOccurs="unbounded"/> </sequence> </complexType>

Complex Type: partitionedpoolTypepartitionedpoolType

Complex Type:	partitionedpoolType
Super-types:	singlepoolType < partitionedpoolType (by extension)
Sub-types:	None

Name	partitionedpoolType
Abstract	No

xmlsolidXML Instance Representation: partitionedpoolType <...> <ger:max-size> xs:int </ger:max-size> [0..1] <ger:min-size> xs:int </ger:min-size> [0..1] <ger:blocking-timeout-milliseconds> xs:int </ger:blocking-timeout-milliseconds> [0..1] <ger:idle-timeout-minutes> xs:int </ger:idle-timeout-minutes> [0..1] Start Choice [1] <ger:match-one> ger:emptyType </ger:match-one> [1] <ger:match-all> ger:emptyType </ger:match-all> [1] <ger:select-one-assume-match> ger:emptyType </ger:select-one-assume-match> [1] End Choice <ger:partition-by-subject> ger:emptyType </ger:partition-by-subject> [0..1] <ger:partition-by-connectionrequestinfo> ger:emptyType </ger:partition-by-connectionrequestinfo> [0..1] </...> xmlsolidSchema Component Representation: partitionedpoolType <xs:complexType name="partitionedpoolType"> <complexContent> <extension base="ger:singlepoolType"> <xs:sequence> <xs:element name="partition-by-subject" type="ger:emptyType" minOccurs="0"/> <xs:element name="partition-by-connectionrequestinfo" type="ger:emptyType" minOccurs="0"/> </xs:sequence> </extension> </complexContent>

Complex Type: resourceadapter-instanceTypeadapterinstanceType

Complex Type:	resourceadapter-instanceType
Super-types:	None
Sub-types:	None

Name	resourceadapter-instanceType
Abstract	no
Documentation	The resourceadapter instance info is put in a separate optional element to provide easier more consistent support for 1.0 adapters.

xmlsolidXML Instance Representation: resourceadapter-instanceType <...> <ger:resourceadapter-name> xs:string </ger:resourceadapter-name> [1] <ger:config-property-setting> ger:config-property-settingType </ger:config-property-setting> [0..*] <naming:workmanager> ... </naming:workmanager> [1] </...> x
mlsolidSchema Component Representation: resourceadapter-instanceType <complexType name="resourceadapter-instanceType"> <sequence> <element name="resourceadapter-name" type=" xs:string "/> <element name="config-property-setting" type=" ger:config-property-settingType " minOccurs="0" maxOccurs="unbounded"/> <element ref=" naming:workmanager "/> </sequence> </complexType>

Complex Type: resourceadapterTyperesourceadapterType1

Complex Type:	resourceadapterType
---------------	---------------------

Super-types:	None
Sub-types:	None

Name	resourceadapterType
Abstract	no

xmlsolidXML Instance Representation: resourceadapterType <...> <ger:resourceadapter-instance> ger:resourceadapter-instanceType </ger:resourceadapter-instance> [0..1] <ger:outbound-resourceadapter> ger:outbound-resourceadapterType </ger:outbound-resourceadapter> [0..1] </...> xmlsolidSchema Component Representation: resourceadapterType <complexType name="resourceadapterType"> <sequence> <element name="resourceadapter-instance" type=" ger:resourceadapter-instanceType " minOccurs="0"/> <element name="outbound-resourceadapter" type=" ger:outbound-resourceadapterType " minOccurs="0"/> </sequence> </complexType>

Complex Type: singlepoolTypesinglepoolType

Complex Type:	singlepoolType
Super-types:	None
Sub-types:	*partitionedpoolType (by extension)

Name	singlepoolType
Abstract	no

xmlsolidXML Instance Representation: singlepoolType <...> <ger:max-size> xs:int </ger:max-size> [0..1] <ger:min-size> xs:int </ger:min-size> [0..1] <ger:blocking-timeout-milliseconds> xs:int </ger:blocking-timeout-milliseconds> [0..1] <ger:idle-timeout-minutes> xs:int </ger:idle-timeout-minutes> [0..1] Start Choice [1] <ger:match-one> ger:emptyType </ger:match-one> [1] <ger:match-all> ger:emptyType </ger:match-all> [1] <ger:select-one-assume-match> ger:emptyType </ger:select-one-assume-match> [1] End Choice </...> xmlsolidSchema Component Representation: singlepoolType <complexType name="singlepoolType"> <sequence> <element name="max-size" type=" xs:int " minOccurs="0"/> <element name="min-size" type=" xs:int " minOccurs="0"/> <element name="blocking-timeout-milliseconds" type=" xs:int " minOccurs="0"/> <element name="idle-timeout-minutes" type=" xs:int " minOccurs="0"/> <choice> <element name="match-one" type=" ger:emptyType "/> <element name="match-all" type=" ger:emptyType "/> <element name="select-one-assume-match" type=" ger:emptyType "/> </choice> </sequence> </complexType>

Complex Type: xatransactionTypextransactionType

Complex Type:	xatransactionType
Super-types:	None
Sub-types:	None

Name	xatransactionType
Abstract	no

xmlsolidXML Instance Representation: xatransactionType <...> <ger:transaction-caching> ger:emptyType </ger:transaction-caching> [0..1] <ger:thread-caching> ger:emptyType </ger:thread-caching> [0..1] </...> xmlsolidSchema Component Representation: xatransactionType <complexType name="xatransactionType"> <element name="transaction-caching" type="ger:emptyType" minOccurs="0"/> <element name="thread-caching" type="ger:emptyType" minOccurs="0"/> </complexType>

Simple Type: fully-qualifies-classTypeclassType

Simple Type:	fully-qualified-classType
Super-types:	xs:string < fully-qualified-classType (by restriction)
Sub-types:	None

Name	fully-qualified-classType
Content	Base XSD Type: string

xmlsolidSchema Component Representation: fully-qualifies-classType <simpleType name="fully-qualified-classType"> <restriction base="xs:string"/> </simpleType>

Legend

Complex Type:	AusAddress
Schema Component Type	

Schema Component Name	
-----------------------	--

Super-types:	Address < AusAddress (by extension)
Sub-types:	QLDAddress (by restriction)

If this schema component is a type definition, its type hierarchy is shown in a gray-bordered box.

Name	AusAddress
Abstract	no

The table above displays the properties of this schema component.

xmlsolidXML Instance Representation <... country="Australia" > <unitNo> string </unitNo> [0..1] <houseNo> string </houseNo> [1] <street> string </street> [1] Start Choice [1] <city> string </city> [1] <town> string </town> [1] End Choice <state> AusStates </state> [1] <postcode> string <<pattern = [1-9][0-9]{3}>> </postcode> [1] ? </...>

The XML Instance Representation table above shows the schema component's content as an XML instance.

- The minimum and maximum occurrence of elements and attributes are provided in square brackets, e.g. 0..1.
- Model group information are shown in gray, e.g. Start Choice ... End Choice.
- For type derivations, the elements and attributes that have been added to or changed from the base type's content are shown in bold.
- If an element/attribute has a fixed value, the fixed value is shown in green, e.g. country="Australia".
- Otherwise, the type of the element/attribute is displayed.
 - If the element/attribute's type is in the schema, a link is provided to it.
 - For local simple type definitions, the constraints are displayed in angle brackets, e.g. <pattern = [1-9][0-9]{3}/>.
- If a local element/attribute has documentation, it will be displayed in a window that pops up when the question mark inside the attribute or next to the element is clicked, e.g. <postcode>.

xmlsolidSchema Component Representation <complexType name="AusAddress"> <complexContent> <extension base=" Address "> <sequence> <element name="state" type=" AusStates "/> <element name="postcode"> <simpleType> <restriction base=" string "> <pattern value="[1-9][0-9]{3}"/> </restriction> </simpleType> </element> </sequence> <attribute name="country" type=" string " fixed="Australia"/> </extension> </complexContent> </complexType>

The Schema Component Representation table above displays the underlying XML representation of the schema component. (Annotations are not shown.)

Glossary

Abstract (Applies to complex type definitions and element declarations). An abstract element or complex type cannot be used to validate an element instance. If there is a reference to an abstract element, only element declarations that can substitute the abstract element can be used to validate the instance. For references to abstract type definitions, only derived types can be used.

All Model Group Child elements can be provided in any order in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-all>.

Choice Model Group Only one from the list of child elements and model groups can be provided in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-choice>.

Collapse Whitespace Policy Replace tab, line feed, and carriage return characters with space character (Unicode character 32). Then, collapse contiguous sequences of space characters into single space character, and remove leading and trailing space characters.

Disallowed Substitutions (Applies to element declarations). If substitution is specified, then substitution group members cannot be used in place of the given element declaration to validate element instances. If derivation methods, e.g. extension, restriction, are specified, then the given element declaration will not validate element instances that have types derived from the element declaration's type using the specified derivation methods. Normally, element instances can override their declaration's type by specifying an xsi:type attribute.

Key Constraint Like Uniqueness Constraint, but additionally requires that the specified value(s) must be provided. See: http://www.w3.org/TR/xmlschema-1/#cldentity-constraint_Definitions.

Key Reference Constraint Ensures that the specified value(s) must match value(s) from a Key Constraint or Uniqueness Constraint. See: http://www.w3.org/TR/xmlschema-1/#cldentity-constraint_Definitions.

Model Group Groups together element content, specifying the order in which the element content can occur and the number of times the group of element content may be repeated. See: http://www.w3.org/TR/xmlschema-1/#Model_Groups.

Nillable (Applies to element declarations). If an element declaration is nillable, instances can use the xsi:nil attribute. The xsi:nil attribute is the boolean attribute, nil, from the <http://www.w3.org/2001/XMLSchema-instance> namespace. If an element instance has an xsi:nil attribute set to true, it can be left empty, even though its element declaration may have required content.

Notation A notation is used to identify the format of a piece of data. Values of elements and attributes that are of type, NOTATION, must come from the names of declared notations. See: http://www.w3.org/TR/xmlschema-1/#cNotation_Declarations.

Preserve Whitespace Policy Preserve whitespaces exactly as they appear in instances.

Prohibited Derivations (Applies to type definitions). Derivation methods that cannot be used to create sub-types from a given type definition.

Prohibited Substitution (Applies to complex type definitions). Prevents sub-types that have been derived using the specified derivation methods from validating element instances in place of the given type definition.

Replace Whitespace Policy Replace tab, line feed, and carriage return characters with space character (Unicode character 32).

Sequence Model Group Child elements and model groups must be provided in the specified order in instances. See: <http://www.w3.org/TR/xmlschema-1/#element-sequence>.

Substitution Group Elements that are members of a substitution group can be used wherever the head element of the substitution group is referenced.

Substitution Group Exclusions (Applies to element declarations). Prohibits element declarations from nominating themselves as being able to substitute a given element declaration, if they have types that are derived from the original element's type using the specified derivation methods.

Target Namespace The target namespace identifies the namespace that components in this schema belongs to. If no target namespace is provided, then the schema components do not belong to any namespace.

Uniqueness Constraint Ensures uniqueness of an element/attribute value, or a combination of values, within a specified scope. See: http://www.w3.org/TR/xmlschema-1/#cidentity-constraint_Definitions.