

geronimo-application-1.1.xsd

{scrollbar}top

Introduction

The following article describes the XML Schema for the geronimo-application.xsd deployment descriptor .In order to deploy an EAR application in Geronimo,there should be a Geronimo specific deployment plan for it called "geronimo-application.xml".The deployment descriptor must be named "META-INF/geronimo-application.xml" in the application's EAR file.

The article organized into the following sections : -

- [#Geronimo Application XML Schema Document](#)
- [Schema Document Properties](#)
- [Global Declarations](#)
- [Global Definitions](#)
 - [Complex Type: ext-moduleType](#)
 - [Complex Type: applicationType](#)
 - [Complex Type: moduleType](#)
 - [Complex Type: pathType](#)
 - [Complex Type: string](#)
- [#Legend](#)
- [#Glossary](#)

Geronimo Application XML Schema Document

```
xmlsolidgeronimo-application-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:geronimo="http://geronimo.apache.org/xml/ns/j2ee/application-
1.1" targetNamespace="http://geronimo.apache.org/xml/ns/j2ee/application-1.1" xmlns:security="http://geronimo.apache.org/xml/ns/security-1.1" xmlns:
sys="http://geronimo.apache.org/xml/ns/deployment-1.1" xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified" version="1.0"> <xs:import namespace="http://geronimo.apache.org/xml/ns/security-1.1" schemaLocation="geronimo-
security-1.1.xsd"/> <xs:import namespace="http://geronimo.apache.org/xml/ns/deployment-1.1" schemaLocation="geronimo-module-1.1.xsd"/> <xs:
element name="application" type="geronimo:applicationType"/> <xs:complexType name="applicationType"> <xs:sequence> <xs:element ref="sys:
environment"/> <xs:element name="module" type="geronimo:moduleType" minOccurs="0" maxOccurs="unbounded"/> <xs:element name="ext-module"
type="geronimo:ext-moduleType" minOccurs="0" maxOccurs="unbounded"/> <xs:element ref="security:security" minOccurs="0"/> <xs:element ref="sys:
gbean" minOccurs="0" maxOccurs="unbounded"/> </xs:sequence> <!--TODO application name should be settable using name-key in the environment.
Does this work? --> <xs:attribute name="application-name" type="xs:string" use="optional"/> </xs:complexType> <xs:complexType name="moduleType">
<xs:annotation> <xs:documentation> Mirrors the moduleType defined by application_1_4.xsd and adds an optional alt-dd element defining a Geronimo
specific deployment descriptor. </xs:documentation> </xs:annotation> <xs:sequence> <xs:choice> <xs:element name="connector" type="geronimo:
pathType"/> <xs:element name="ejb" type="geronimo:pathType"/> <xs:element name="java" type="geronimo:pathType"/> <xs:element name="web" type="
geronimo:pathType"/> </xs:choice> <xs:choice> <xs:element name="alt-dd" type="geronimo:pathType"> <xs:annotation> <xs:documentation> Specifies
an optional URI to the post-assembly version of the Geronimo specific deployment descriptor file for a particular J2EE module. </xs:documentation> </xs:
annotation> </xs:element> <xs:any namespace="##other" processContents="lax"/> </xs:choice> </xs:sequence> </xs:complexType> <xs:complexType
name="ext-moduleType"> <xs:annotation> <xs:documentation> Mirrors the moduleType defined by application_1_4.xsd and adds an optional alt-dd
element defining a Geronimo specific deployment descriptor. </xs:documentation> </xs:annotation> <xs:sequence> <xs:choice> <xs:annotation> <xs:
documentation> This is the name for the module, not the path inside the ear. </xs:documentation> </xs:annotation> <xs:element name="connector" type="
geronimo:pathType"/> <xs:element name="ejb" type="geronimo:pathType"/> <xs:element name="java" type="geronimo:pathType"/> <xs:element name="
web" type="geronimo:pathType"/> </xs:choice> <xs:choice> <xs:element name="internal-path" type="xs:token"/> <xs:element name="external-path" type="
xs:token"/> </xs:choice> <!--TODO consider an alt-spec-dd element--> <!--Do not allow vendor plan inside ear. If it is in the ear, this does not need to be
an external module --> <xs:any namespace="##other" processContents="lax"/> </xs:sequence> </xs:complexType> <xs:complexType name="pathType">
<xs:annotation> <xs:documentation> Mirrors j2ee:pathType. </xs:documentation> </xs:annotation> <xs:simpleContent> <xs:restriction base="geronimo:
string"/> </xs:simpleContent> </xs:complexType> <xs:complexType name="string"> <xs:annotation> <xs:documentation> Mirrors j2ee:string. </xs:
documentation> </xs:annotation> <xs:simpleContent> <xs:extension base="xs:token"> <xs:attribute name="id" type="xs:ID"/> </xs:extension> </xs:
simpleContent> </xs:complexType> </xs:schema>
```

Schema Document Properties schema

Target Namespace	http://geronimo.apache.org/xml/ns/j2ee/application-1.1 As all the other application deployment descriptors must indicate the application schema by using the J2EE namespace the geronimo-application also uses this J2EE namespace
Version	1.0 - The version indicates the published version of the schema
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://geronimo.apache.org/xml/ns/security-1.1(at geronimo-security-1.1.xsd) http://geronimo.apache.org/xml/ns/deployment-1.1(at geronimo-module-1.1.xsd)
--------------------	---

Declared Namespaces

Prefix	Namespace	Description
xml	http://www.w3.org/XML/1998/namespace	The namespace whose name is http://www.w3.org/XML/1998/namespace is bound by definition to the prefix xml: according to Namespaces in XML, W3C Recommendation 14 Jan 1999
Geronimo	http://geronimo.apache.org/xml/ns/j2ee/application-1.1	This is the target name space defined for Geronimo EAR applications
security	http://geronimo.apache.org/xml/ns/security-1.1	This schema describes Common security elements used by other plans in Security Mapping module.
sys	http://geronimo.apache.org/xml/ns/deployment-1.1	This schema Used to deploy new services in Geronimo in a standalone plan, and also contains common elements used by many other plans and the module type is Server Plans & Common Elements
xs	http://www.w3.org/2001/XMLSchema	This document describes the XML Schema namespace. It also contains a directory of links to these related resources, using Resource Directory Description Language http://www.rddl.org/

xmlsolidSchema Component Representation `xs:schema xmlns:geronimo="http://geronimo.apache.org/xml/ns/j2ee/application-1.1" targetNamespace="http://geronimo.apache.org/xml/ns/j2ee/application-1.1" xmlns:security="http://geronimo.apache.org/xml/ns/security-1.1" xmlns:sys="http://geronimo.apache.org/xml/ns/deployment-1.1" xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="unqualified" version="1.0"> <xs:import namespace="http://geronimo.apache.org/xml/ns/security-1.1" schemaLocation="geronimo-security-1.1.xsd"/> <xs:import namespace="http://geronimo.apache.org/xml/ns/deployment-1.1" schemaLocation="geronimo-module-1.1.xsd"/> </xs:schema>`

Global Declarations global

This section highlights the application element declaration that is available for use in the content models throughout the schema.

Element: Application

Name	Application
Type	N/A
Geronimo	ApplicationType
Nilable	No
Abstract	No
Documentation	The application element is the root element of a J2EE application deployment descriptor in Geronimo

xmlsolidXML Instance Representation `<geronimo:application> <dep:moduleId="xs:string[0..1]"> <dep:groupId="xs:string[1]"> <dep:artifactId="xs:string[1]"> <dep:version="xs:string[1]"> <dep:type="xs:string[1]"> <sys:import> ... </sys:import> [0..*] <sys:dependency> ... </sys:dependency> [0..*] <geronimo:module> geronimo:moduleType </geronimo:module> [0..*] <geronimo:ext-module> geronimo:ext-moduleType </geronimo:ext-module> [0..*] <security:security> ... </security:security> [0..1] <sys:gbean> ... </sys:gbean> [0..*] </geronimo:application>` xmlsolidSchema Component Representation `<xs:element name="application" type="geronimo:applicationType"/>`

Global Definitionsdefinitions

[This section highlights the complex types (an XML element that contains other elements and/or attributes) which can be referenced within the same schema or by other schemas.]

Complex Type: applicationTypeapplicationType

Super-types	None
Sub-types	None
Name	applicationType
Abstract	no

Documentation	The applicationType defines the structure of the application in geronimo. The application deployment descriptor must have one module element for each J2EE module in the application package. A module element is defined by moduleType definition
---------------	---

xmlsolidXML Instance Representation: applicationType <... <sys:environment>...</sys:environment> application-name=" xs:string [0..1]"> <sys:import> ... </sys:import> [0..*] <sys:dependency> ... </sys:dependency> [0..*] <geronimo:module> geronimo:moduleType </geronimo:module> [0..*] <geronimo:ext-module> geronimo:ext-moduleType </geronimo:ext-module> [0..*] <security:security> ... </security:security> [0..1] <sys:gbean> ... </sys:gbean> [0..*] </...> xmlsolidSchema Component Representation: applicationType <xs:complexType name="applicationType"> <xs:sequence> <xs:element ref="sys:environment"/> <xs:element name="module" type="geronimo:moduleType" minOccurs="0" maxOccurs="unbounded"/> <xs:element name="ext-module" type="geronimo:ext-moduleType" minOccurs="0" maxOccurs="unbounded"/> <xs:element ref="security:security" minOccurs="0"/> <xs:element ref="sys:gbean" minOccurs="0" maxOccurs="unbounded"/> </xs:sequence> <!--TODO application name should be settable using name-key in the environment. Does this work? --> <xs:attribute name="application-name" type="xs:string" use="optional"/> </xs:complexType>

Complex Type: ext-moduleTypeext-moduleType

Super-types	None
Sub-types	None
Name	ext-moduleType
Abstract	no
Documentation	Mirrors the moduleType defined by application_1_4.xsd and adds an optional alt-dd element defining a Geronimo specific deploy descriptors.

xmlsolidXML Instance Representation: ext-ModuleType <xs:complexType name="ext-moduleType"> <xs:annotation> <xs:documentation> Mirrors the moduleType defined by application_1_4.xsd and adds an optional alt-dd element defining a Geronimo specific deployment descriptor. </xs:documentation> </xs:annotation> <xs:sequence> <xs:choice> <xs:annotation> <xs:documentation> This is the name for the module, not the path inside the ear. </xs:documentation> </xs:annotation> <xs:element name="connector" type="geronimo:pathType"/> <xs:element name="ejb" type="geronimo:pathType"/> <xs:element name="java" type="geronimo:pathType"/> <xs:element name="web" type="geronimo:pathType"/> </xs:choice> <xs:choice> <xs:element name="internal-path" type="xs:token"/> <xs:element name="external-path" type="xs:token"/> </xs:choice> <!--TODO consider an alt-spec-dd element--> <!--Do not allow vendor plan inside ear. If it is in the ear, this does not need to be an external module --> <xs:any namespace="##other" processContents="lax"/> </xs:sequence> </xs:complexType> xmlsolidSchema Component Representation: ext-ModuleType <xs:complexType name="pathType"> <xs:annotation> <xs:documentation> Mirrors j2ee:pathType. </xs:documentation> </xs:annotation> <xs:simpleContent> <xs:restriction base="geronimo:string"/> </xs:simpleContent> </xs:complexType>

Complex Type: stringstring

Super-types	xs:token < string (by extension)
Sub-types	None
Name	string
Abstract	no
Documentation	Mirrors j2ee:string.

xmlsolidXML Instance Representation: string <... id=" xs:ID [0..1]"> xs:token </...> xmlsolidSchema Component Representation: string <xs:complexType name="string"> <xs:simpleContent> <xs:extension base="xs:token"> <xs:attribute name="id" type="xs:ID"/> </xs:extension> </xs:simpleContent> </xs:complexType>

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/#cidentity-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/#cidentity-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.

Nilable (Applies to element declarations). If an element declaration is nilable, 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 Substitutions (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.