geronimo-application.xml

{scrollbar}

top

The Geronimo deployment plan for an enterprise application is an XML document. It is defined by the geronimo-application-1.1.xsd schema, which can be found in the **schema/** subdirectory of the main Geronimo installation directory. The deployment plan for an enterprise application can be included in the application EAR, in which case it should be named META-INF/geronimo-application.xml or saved as a separate file and provided to the deploy tool when the application is deployed

The deployment plan should always use the Geronimo Application namespace. Additionally, it has a required attribute to identify its configuration name, and an optional attribute to select a parent configuration. A typical enterprise application deployment plan looks like this:

This article is organized into the following sections : -

Warning

1.0 Creating an Enterprise Application EAR

1.1 application.xml Format

2.0 The Enterprise Application Geronimo Deployment Plan Overview

There is a specific internal structure for the Enterprise Application deployment plan for Geronimo and follow a specific structure and order of elements.

The deployment plan should always use the Geronimo Application namespace. Additionally, it has a required attribute to identify its configuration name, and an optional attribute to select a parent configuration. A typical Geronimo enterprise application deployment plan can be implemented as follows.

META-INF/geronimo-application.xml

<?xml version="1.0" encoding="UTF-8"?> <application xmlns="http://geronimo.apache.org/xml/ns/j2ee/application-1.1" xmlns:sys="http://geronimo.apache.org/xml/ns/deployment-1.1" xmlns:security="http://geronimo.apache.org/xml/ns/security-1.1" inverseClassloading="false"> ... </application>

The attributes are defined below

xmlns The main namespace for the deployment plan, which should always be http://geronimo.apache.org/xml/ns/j2ee/application-1.1

xmlns:sys A secondary namespace, used to identify the common elements for third-party libraries and custom services. If present, this should always be set to http://geronimo.apache.org/xml/ns/deployment-1.1

xmlns:security A secondary namespace, used to identify the common elements for security role settings. If present, this should always be set to http://geronimo.apache.org/xml/ns/security-1.1

2.Class Path Settings for Enterprise Application Deployment Plan

import Refers to another configuration deployed in the server. That configuration will be added as a parent of this one (a configuration may have more than one parent). The main effect is that the class loader for the application will add the class loader for that configuration as a parent. Additionally, the parent configuration will be started before the application is started.

dependency Adds a third-party library to the class path for the application. Any common libraries used in this manner should be located in a subdirectory of the repository/ directory of the main Geronimo installation.

hidden-classes Lists packages or classes that may be in a parent class loader, but should not be exposed from there to the application. This is typically used when the application wants to use a different version of a library that one of its parent configurations (or Geronimo itself) uses. For example, Geronimo 1.0 uses Log4J 1.2.8. If the application wanted to use a newer version, it could include the newer version and then add org.apache.log4j to the list of hidden-classes so that the Log4J classes could not be loaded from a parent class loader.

non-overridable-classes Lists packages or classes that the application should always load from a parent class loader, and never load from its own class path.

filter Used to list classes or packages. The format is a comma-separated list of packages or fully-qualified class names (for example: javax.servlet,javax.ejb).

3.Specifying Imports and Dependencies in Class Path setting.

moduleId A unique name identifying this module. This name can be passed on the server command line to activate this application when the server starts.

groupId Identifies the parent configuration for this application (the value specified here should match the configId for that module or application). This can be used to make the application depend on another module such as a standalone EJB JAR or J2EE Connector or another entire application (or it should otherwise be omitted or set to the usual parent for J2EE modules, geronimo/j2ee-server/1.0/car).

application-name In the standard J2EE Management interface (JSR-77), an application will normally be identified by its configId. You can use this optional attribute to specify a different name for the application to use to identify itself and its children in the management interface. (This will be applied to all GBeans within the configuration, but the configuration GBean itself will still use the configId.) One way to use this is to deploy several database pool or JMS resource adapters in an EAR, but set the application-name to null to make all the resource adapters appear to be deployed at the top level of the server, making it easier to reference them from other applications.

inverseClassloading If it's set to true, the class loader for this application tries to load a class before checking whether the class is available from its parent class loader. If omitted or set to false, the normal (check parent first) class loader delegation behavior is used .

3.Configuring Application Modules

The J2EE deployment descriptor (META-INF/application.xml) lists the modules in the application and gives some basic options for each, including:

- A context root for web application modules
- The ability to store the J2EE deployment descriptors for each module in the EAR (instead of packaged within each module)

The Geronimo deployment plan provides two different options for overriding the usual packaging of the module-level Geronimo deployment plans within the modules:

- The deployment plan for a module can be stored in the EAR
- The deployment plan for a module can be embedded in the deployment plan for the application

connector
Holds the location of a J2EE Connector module. Must match the same element in application.xml
ejb

Holds the location of an EJB module. Must match the same element in application.xml
java

Holds the location of an application client module. Must match the same element in application.xml
web

Holds the location of a web application module. Must match the web-uri in application.xml

There are two possibilities for identifying the deployment plan of the module:

alt-dd Specifies a location inside the EAR where the Geronimo deployment plan for this module can be found. Note that the alt-dd element in application.xml specifies an alternate location for the module's J2EE deployment descriptor, whereas the alt-dd here specifies an alternate location for the module's Geronimo deployment plan.

Any Module The module's entire Geronimo deployment plan may be embedded in the EAR deployment plan at this location. It should use the same elements and namespaces as normal. Note that this is commonly used by JSR-88 deployment tools (which must save a single deployment plan for the application and all its modules), but is used less frequently for handcrafted deployment plans.

3.0 Work out Geronimo Deployment plan

This section will be discussed the details of how Geronimo deployment plan for EAR has implemented for different type of applications accordingly.The Sample Application section of the Geronimo user guide has been heavily used to discusses the deployment plan in depth.

3.1.1 HelloWord EAR application

The article deployment plan level-1 gives an primary understand to the user in basics of Geronimo v1.1 specific deployment plan for an EAR and go head to work with complex applications.Further details about application can be find the original location as at http://cwiki.apache.org/GMOxDOC11/deployment-plans-level-1.html.Therefore here are deployment plans used in this sample application.

Geronimo requires that every EAR file has a standard META-INF/application.xml deployment descriptor. This may be configured for J2EE 1.2, J2EE 1.3, or J2EE 1.4, and should follow the appropriate XML format and following is the application.xml for the above application.

xmlsolidapplication.xml <?xml version="1.0" encoding="ISO-8859-1"?> <application> <display-name>HelloWorldEar</display-name> <module> <web> <web-uri>helloworld.war</web-uri> <context-root>/hello</context-root> </web> </module> </application>

This deployment plan define the war module in **<web><web-uri>helloworld.war<web-uri>** and the root context of it.The geronimo-application.xml is the geronimo specific plan for above application.

xmlsolidgeronimo-application.xml <application application-name="HelloWorldEar" xmlns="http://geronimo.apache.org/xml/ns/j2ee/application-1.1" xmlns:sec="http://geronimo.apache.org/xml/ns/security-1.1" xmlns:sys="http://geronimo.apache.org/xml/ns/deployment-1.1"> <sys:environment> <sys:moduleId> <sys:groupId>default</sys:groupId> <sys:artifactId>HelloWorldEar_HelloWorldEar</sys:artifactId> <sys:version>1-default</sys:version> <sys:type>car</sys:type> </sys:moduleId> <sys:dependencies/> <sys:hidden-classes/> <sys:non-overridable-classes/> </sys:environment> </application>

3.2 Sample Code for the Configuring modules*

This sample has been taken from the user guide, Sample application.http://cwiki.apache.org/GMOxDOC11/jms-and-mdb-sample-application.html.This is a sample deployment plan for A context root for web application module with a ejb application.

xmlsolidgeronimo-application.xml <application xmlns="http://geronimo.apache.org/xml/ns/j2ee/application-1.1"> <dep:environment xmlns:dep="http://geronimo.apache.org/xml/ns/deployment-1.1"> <dep:moduleId> <dep:groupId>samples</dep:groupId> <dep:artifactId>Order</dep:artifactId> <dep:version>1.0</dep:version> <dep:type>car</dep:type> </dep:moduleId> <dep:dependencies/> <dep:hidden-classes/> <dep:non-overridable-classes/> </dep:environment> </application>

In this case, the EJB deployment plan is stored inside the EAR and it is compressed with Order.jar inside META-INF/openejb.xml and ejb-jar.xml . The Web application deployment plan is actually right there inside the EAR deployment plan,compressed with Orderweb.war/WEB-INF/web.xml and geronimo-web.xml.Also the EAR deployment plans both are directly stored right in side the Order.ear/META-INF/application.xml and geronimo-application.xml

xmlsolidapplication.xml <?xml version="1.0" encoding="UTF-8"?> <application 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/application_1_4.xsd" version="1.4"> <module> <ejb>OrderEjb.jar</ejb> </module> <module> <web> <web-uri>OrderWeb.war</web-uri> <context-root>/Order</context-root> </web> </module> </application>

3.3 Sample code for Adding a new module

This sample application uses a connector module in the deployment plan itself.This sample code for geronimo-application and the application.xml has been excerpt from user guide, sample application located at http://cwiki.apache.org/GMOxDOC11/ejb-sample-application.html

Therefore addition to configuring modules listed in application.xml, the Geronimo plan can add references to new modules, either in the EAR or located in the Geronimo repository.

xmlsolidgeronimo-application.xml <?xml version="1.0" encoding="UTF-8"?> <application xmlns="http://geronimo.apache.org/xml/ns/j2ee/application-1.1"> <dep:environment xmlns:dep="http://geronimo.apache.org/xml/ns/deployment-1.1"> <dep:moduleId> <dep:groupId>default</dep:groupId> <dep:artifactId>Bank</dep:artifactId> <dep:version>1.0</dep:version> <dep:type>car</dep:type> </dep:moduleId> <dep:dependencies/> <dep:hidden-classes/> <dep:non-overridable-classes/> </dep:environment> <module> <connector>tranql-connector-1.2.rar</connector> <alt-dd>BankPool.xml</alt-dd> </module> </application> xmlsolidapplication.xml <?xml version="1.0" encoding="UTF-8"?> <application 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/application_1_4.xsd" version="1.4"> <module> <ejb>BankEJB.jar</ejb> </module> <module> <web> <web-uri>BankWeb.war</web-uri> <context-root>/Bank</context-root> </web> </module> <module> <connector>tranql-connector-1.2.rar</connector> </module> </application>

geronimo-application.xml and application.xml define the main components of the EAR. Both EJB component and Web archive information are given in these files. Additionally, these two XML files define application scoped database connection pool with tranql-connector-1.2.rar and BankPool.xml.

In order to have the ext-module following there the elements need to be configured.

connector , **ejb**, **java** , **web**

These elements are same as described previously in the section of configuring modules.These identify the modules
(though they don't need to match anything in application.xml in this case).

internal-path

The internal path indicates that the module is packaged in the EAR, and a path specified here is relative to the root of the EAR.

external-path
The external path indicates that the module is located in the Geronimo repository. In fact, the value specified here is not a path at all, but is a URI formatted according to the standard repository URI syntax (e.g. tranql/tranql-connector/1.1/rar).

Any Other
If the Geronimo deployment plan for the module is not packaged in the module, it must be included here.

4. Conclusion

This simply gives a overview of EAR deployment plans for Geronimo.But it's nice to cover areas such as "Application-Wide Security Mapping" or type mapping which is not covered here.Your contribution to improve this document is grealty appreciated.