


WebSphere v5 Administration, Network Deployment Edition

Loading Java Classes



Class Loader

- A class loader is a Java class that loads compiled Java byte code of other classes.
 - Depending on the capability of the loader, the byte code can be loaded from file system, ZIP/JAR file or from the network.
 - Usually, one can configure the source location a class loader looks for a class.
- WebSphere uses multiple class loaders.
 - When the application code refers to a class that is not already loaded, WebSphere automatically uses the right class loader to load the class.
- Understanding the relation between these loaders is key to:
 - Appropriately package an enterprise application.
 - Share common JAR files between applications.
 - Allow different applications to use different versions of the same class.
 - Solve ClassNotFoundException problems.

© Web Age Solutions, Inc. 2003 33

Source Location

The CLASSPATH environment variable is a common way to specify the source JAR, ZIP and file system directories that contain classes. But, different class loaders may have different mechanisms for this. This will be discussed in details later.


Correct Class Loader?

What among the many class loaders used in WebSphere is the correct one in a given situation? This depends on the thread and the J2EE module the application code is running on. For example, if a Servlet application code needs to load a class, WebSphere will use the class loader specifically associated with the web module of the Servlet.



Class Loader Hierarchy

- Various class loaders used by WebSphere follow a strict parent-child hierarchy.
 - A child can have a single parent only.
- The following rules apply:
 - Delegation – By default, when a class loader needs to load a class, it first asks its parent to load it. If none of the parents in the hierarchy chain can load the class, the class loader attempts to load it.
 - We can configure the rule and have a class loader attempt to load the class first before delegating the process to its parent.
 - If a class loader fails to load a class, it never asks its children to load it.
 - Visibility – A class loaded by a class loader can use only the classes loaded by that class loader or any of its parents.



WebSphere Class Loaders

- System class loader – Loads classes from:
 - The JDK bootstrap directory (usually, jre/lib).
 - JDK extension directory (usually, jre/lib/ext).
 - The JAR, ZIP and directories specified in the CLASSPATH environment variable.
- WebSphere Extension class loader – Loads classes from the JAR and ZIP files located in a directory listed in the ws.ext.dirs system property.
- Application (EAR) class loader – Loads classes from:
 - The EJB module JAR files of a EAR.
 - JAR files referred to by any EJB module in the META-INF/MANIFEST.MF file within that module.
- Web module (WAR) class loader – Loads classes from:
 - Loose class files (*.class) in the WEB-INF/classes folder of the module.
 - JAR files in the WEB-INF/lib folder of the module.
 - Any other JAR files referred to by the web module in the META-INF/MANIFEST.MF file.

© Web Age Solutions, Inc. 2003 35

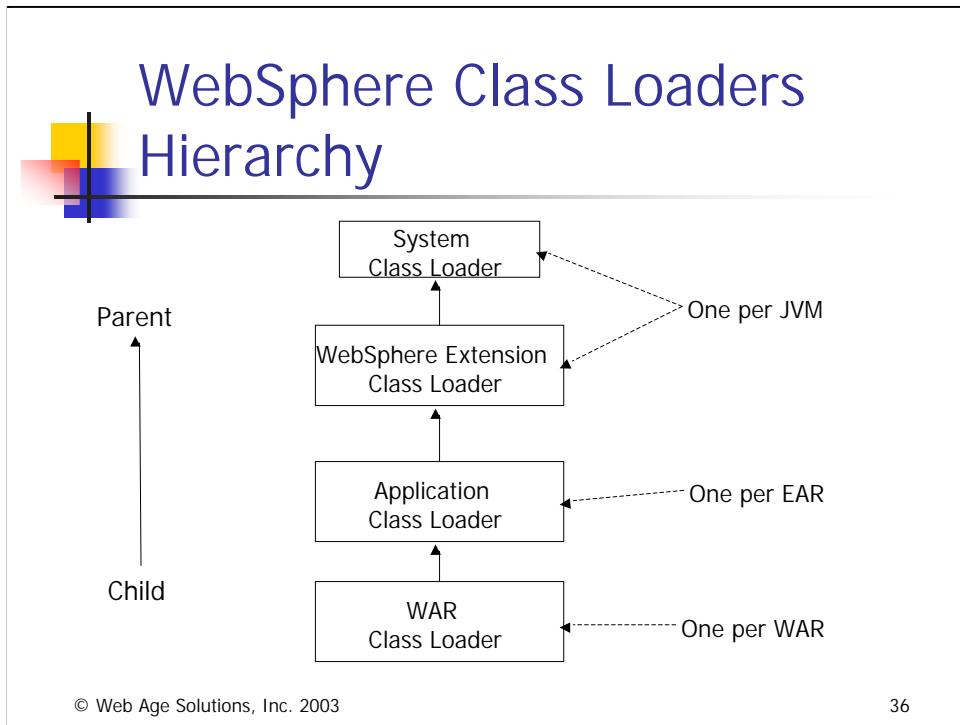
The `ws.ext.dirs` system parameter is specified as a command line argument of JVM when in the application server startup script – `startServer.bat(sh)`.

Reference to other JAR files can be specified in a JAR file's META-INF/MANIFEST.MF file as follows:

Manifest-Version: 1.0

Class-Path: log4j-1.2.8.jar xml4j.jar

The JAR file names are path names relative to the root of the EAR file.





Class Loading Isolation Options

- Complete Isolation
- Medium Isolation
- Minimal Isolation



Complete Isolation

- Each web module uses a different class loader
- Two web modules in the same EAR can use separate versions of the same class.



Medium Isolation

- Enterprise applications are isolated from one another
- All the web modules within an application share a class loader
 - Web modules in a EAR can not load different versions of the same class
 - If indeed two web modules contain different versions of the same class, resolve the race condition by setting web module start weights – a lower start weight means that module (and its versions of the classes) load first



Minimal Isolation

- Each WebSphere JVM has a global class loader
- Enterprise applications share classes
 - As with web modules, the application start weight determines whose version of a class loads – the first one wins



Which option to use?

- Greater isolation tends to be better
 - These applications and modules are usually built in isolation, so they should be deployed in isolation
 - Using a global class loader may save memory, but the potential confusion may not be worth it
- So use the 'Complete Isolation' level unless you have an application that just doesn't work with this setting
 - In that case, the application should be changed to support 'complete isolation' as soon as possible



Configuring the Class Loading Policy

- The policy is configured at two levels:
 - Application server's application class loading policy
 - MULTIPLE – Each enterprise application gets its own application class loader.
 - SINGLE – Only one Enterprise application class loader for all applications deployed in an application server.
 - Enterprise application's "WAR class loader" policy
 - MODULE – Every web module gets a separate WAR class loader.
 - APPLICATION – Web modules do not get a WAR class loader. The application class loader does the job.
- Each isolation option translates into different settings for these two configuration parameters

Configuring Application Server Policy

The screenshot shows the WebSphere Administration Console interface. On the left is a navigation tree with 'Application Servers' selected. The main area displays the configuration for 'server1'. A table titled 'General Properties' contains the following information:

General Properties		
Name	server1	The display name for the server.
Initial State	Started	The execution state requested when the server is first started.
Application classloader policy	Multiple	Specifies whether there is a single classloader for all applications ("Single") or a classloader per application ("Multiple").
Application class loading mode	Parent first	Specifies the class loading mode when the application classloader policy is "Single".

At the bottom of the configuration window are buttons for 'Apply', 'OK', 'Reset', and 'Cancel'. The copyright notice '© Web Age Solutions, Inc. 2003' is at the bottom left, and the page number '43' is at the bottom right.


To configure the application server's class loading policy, go to the list of application servers, then select the application server to configure. The application classloader policy is one of the General Properties of an application server.

Configuring Applications WAR Class Loader Policy

Property	Value	Description
Enable Distribution	<input checked="" type="checkbox"/>	binding, extensions, and deployment descriptors located with the application deployment document, the deployment.xml file (default), or those located in the application's ear file. <input type="checkbox"/> Specifies whether the application will be distributed automatically to other nodes on the cell. The default is for automatic distribution.
Classloader Mode	* PARENT_FIRST	<input type="checkbox"/> Specifies whether classes are loaded via the parent classloader before this one.
WAR Classloader Policy	* Module	<input type="checkbox"/> Defines whether there is a single classloader for all WARs in the application or a classloader per WAR in the application.
Reload Enabled	<input checked="" type="checkbox"/>	<input type="checkbox"/> Specifies if class reloading is enabled for application files when they get updated.
Reload Interval	<input type="text"/>	<input type="checkbox"/> The timeperiod (in seconds) in which the application's filesystem will be scanned for updated files.

© Web Age Solutions, Inc. 2003 44

To configure the application's WAR classloader policy, go to the list of enterprise applications, then select the application to configure. The WAR classloader policy is one of the General Properties of an application.



Configuring the Options

Isolation Option	Application Server Policy	WAR Module Policy
Complete	Multiple	Module
Medium	Multiple	Application
Minimal	Single	Application

© Web Age Solutions, Inc. 2003

45

The application server policy is either “Multiple” or “Single”. Single indicates a single class loader for all applications. Multiple indicates multiple class loaders: one for each application.

The WAR module policy is either “Module” or “Application”. Application indicates a single class loader for the entire application. Module indicates a class loader for each module.

Choosing “Single” for the application server policy and “Module” for the WAR module policy doesn’t make sense, since “Single” means one class loader for the entire application server. If you choose “Single”, then “Application” is implied.