

Oracle WebLogic Server

The #1 Application Server for
Enterprise Java and SOA Developers



ORACLE®

WebLogic Server 11gR1 PS3 (10.3.4) DEMO *Filtering ClassLoader & Classloader Analysis Tool*

Jeffrey West
Application Grid Product Management

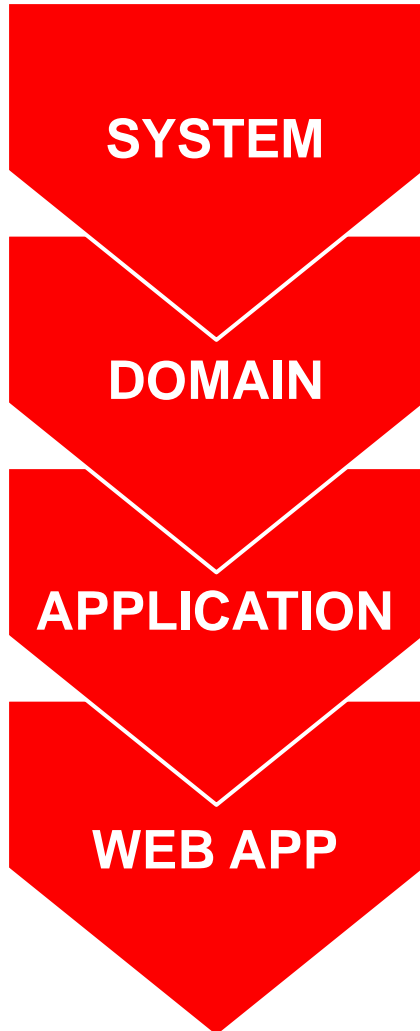


Agenda

- • Overview of WebLogic's ClassLoaders
- Overview of WebLogic's ClassLoader Analysis Tool
- Demo of WebLogic's ClassLoader Analysis Tool
- Conclusion & WebLogic Resources

WebLogic ClassLoader Hierarchy

Top-Down Class Loading (Default)



SYSTEM ClassLoader loads:

- WebLogic Implementation classes
- Classes on the System Classpath
- PRE_CLASSPATH and EXT_PRE_CLASSPATH

DOMAIN ClassLoader loads:

- Classes from <domain_dir>/lib

APPLICATION ClassLoader loads:

- Java EE Shared Libraries referenced in weblogic-application.xml
- Any Modules defined in the application
- Libraries from Java EE 5 <library-directory> directive or /lib if none configured
- Libraries from <EAR>/APP-INF/lib

WEB APP ClassLoader loads:

- Java EE Shared Libraries referenced in weblogic.xml
- WEB-INF/classes
- WEB-INF/lib
- Enables Servlets and JSPs to see EJB classes
- Enables redeployment of Web Apps w/o redeploying EJBs

WebLogic Filtering ClassLoader

Force classes to be loaded from the APPLICATION



- The FilteringClassLoader mechanism allows you to specify classes that should always be loaded from the application (not the SYSTEM ClassLoader)
 - This allows you to use alternate versions of applications, such as Xerces and Ant, than those that are packed with WebLogic
- Enables resources from the child of the FilteringClassLoader (an Application classloader) down to the calling classloader are returned before the ones from the system classloader
- The FilteringClassLoader is configured with a list of packages specified in weblogic-application.xml or weblogic.xml (introduced in 10.3.3) files.

Filtering ClassLoader Configuration

Specify Packages to load from APP-INF/lib & WEB-INF/lib

- Specify the packages that should be loaded by the Application including:
 - Java EE Shared Libraries
 - EAR: /lib (preferred)
 - EAR: APP-INF/lib
 - WAR: WEB-INF/lib
- Overrides the classes that are loaded with WebLogic allowing you to use libraries that may conflict with those included in WebLogic

```
<prefer-application-packages>  
  <package-name>org.apache.log4j.*</package-name>  
  <package-name>antlr.*</package-name>  
</prefer-application-packages>
```

Web Filtering ClassLoader Configuration

Load Classes from WEB-INF

- There are two options for configuring the classloader in at the Web Application level
 - <container-descriptor> / <prefer-application-packages>
 - <container-descriptor> / <prefer-web-inf-classes>
- Only one of these options can be used at a time
- Oracle recommends using <prefer-application-packages> to configure the FilteringClassLoader

<prefer-application-packages> (Recommended)

Allows a Web Application to use its own third-party libraries from WEB-INF/lib by specifying certain packages that should always be loaded from the Web Application

<prefer-web-inf-classes>

Allows a Web application to use its own version third-party classes from WEB-INF/classes, which might also be part of WebLogic Server

Customer ClassLoader Structure

Advanced Configuration for Reloading Classes

- Custom classloader hierarchies allow better control over class visibility and reload-ability
- The ability to create custom module ClassLoaders provides a mechanism to declare alternate classloader organizations that allow the following:
 - Reloading individual EJB modules independently
 - Reloading groups of modules to be reloaded together
 - Reversing the parent child relationship between specific Web modules and EJB modules
 - Namespace separation between EJB modules

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <classloader-structure>
3 <module-ref>
4 <module-uri>ejb1.jar</module-uri>
5 </module-ref>
6 <module-ref>
7 <module-uri>web3.war</module-uri>
8 </module-ref>
9 <classloader-structure>
10 <module-ref>
11 <module-uri>web1.war</module-uri>
12 </module-ref>
13 </classloader-structure>
14 <classloader-structure>
15 <module-ref>
16 <module-uri>ejb3.jar</module-uri>
17 </module-ref>
18 <module-ref>
19 <module-uri>web2.war</module-uri>
20 </module-ref>
21 <classloader-structure>
22 <module-ref>
23 <module-uri>web4.war</module-uri>
24 </module-ref>
25 </classloader-structure>
26 <classloader-structure>
27 <module-ref>
28 <module-uri>ejb2.jar</module-uri>
29 </module-ref>
30 </classloader-structure>
31 </classloader-structure>
32 </classloader-structure>
```

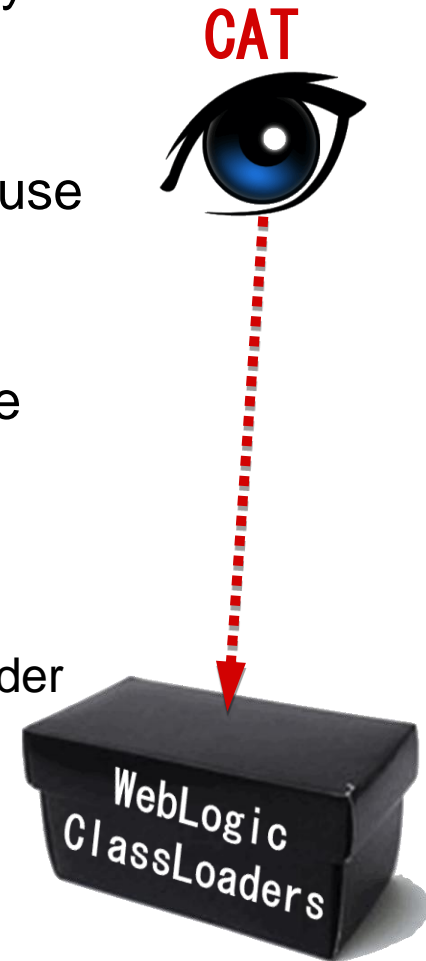
Agenda

- Overview of WebLogic's ClassLoaders
- • Overview of WebLogic's ClassLoader Analysis Tool
- Demo of WebLogic's ClassLoader Analysis Tool
- Conclusion & WebLogic Resources

WebLogic ClassLoader Analysis Tool

NEW in WebLogic 11gR1 (10.3.4)

- Application provided libraries can collide with 3rd party libraries used by WebLogic Server
 - Hard to diagnose class and library conflicts
- Filtering Classloader feature enables applications to use their own libraries
 - Correctly configuring it can be a challenge
- Classloader Analysis Tool Helps Identify and Resolve Conflicts Quickly
 - CAT cracks open the classloader black box
 - Displays classloaders' hierarchies and sources
 - Allows you to search for a class/resource on a classloader
 - Views class definitions, interfaces
 - Analyzes classpath conflicts, generates corresponding filtering classloader configuration



Analyze Classpath Conflicts

WebLogic 10.3.4 ClassLoader Analysis Tool

- The ClassLoader Analysis tool shows you where the class conflicts are
- This allows you to see the conflicting libraries, where they are located and allows you to decide how to resolve it



Conflicts

Resource: antlr.ANTLRException

Checksum: 36cf1a5650c0cfb5793e6ef3c77c93e5

Load Location:
zip:C:/Oracle/Middleware/user_projects/domains/ops_domain/servers/AdminServer/tmp/_WL_user/hudson-1/sjndxx/war/WEB-INF/lib/antlr-2.7.6.jar!/antlr/ANTLRException.class

Classloader Type: weblogic.util.classloaders.ChangeAwareClassLoader

Classloader Hash Code: 36105987

Classloader Search Order: 36105987 ->36105987

Alternative Locations:

- file:/C:/Oracle/Middleware/modules/com.bea.core.antlr.runtime_2.7.7.jar!/antlr/ANTLRException.class

Source:

```
Compiled from "ANTLRException.java"
public class antlr.ANTLRException extends java.lang.Exception{
    public antlr.ANTLRException();
    public antlr.ANTLRException(java.lang.String);
    public antlr.ANTLRException(java.lang.String, java.lang.Throwable);
    public antlr.ANTLRException(java.lang.Throwable);
}
```

ClassLoader Configuration Suggestion

WebLogic 10.3.4 ClassLoader Analysis Tool

- The ClassLoader Analysis tool provides the XML configuration necessary for configuring the Filtering ClassLoader
- This significantly eases the configuration required to take advantage of this advanced feature
- Only available in WebLogic 10.3.4 and later

Suggested Solution

```
<prefer-application-packages>
  <package-name>antlr.*</package-name>
  <package-name>antlr.collections.*</package-name>
  <package-name>antlr.collections.impl.*</package-name>
  <package-name>antlr.debug.misc.*</package-name>
  <package-name>com.ctc.wstx.*</package-name>
  <package-name>com.sun.activation.*</package-name>
  <package-name>com.sun.mail.*</package-name>
  <package-name>com.sun.xml.*</package-name>
  <package-name>javax.activation.*</package-name>
  <package-name>javax.mail.*</package-name>
  <package-name>javax.mail.event.*</package-name>
  <package-name>javax.mail.internet.*</package-name>
  <package-name>javax.mail.search.*</package-name>
  <package-name>javax.mail.util.*</package-name>
  <package-name>javax.xml.*</package-name>
  <package-name>javax.xml.namespace.*</package-name>
  <package-name>javax.xml.stream.*</package-name>
  <package-name>org.apache.commons.*</package-name>
  <package-name>org.apache.log.*</package-name>
  <package-name>org.apache.oro.*</package-name>
  <package-name>org.apache.tools.*</package-name>
  <package-name>org.codehaus.stax2.*</package-name>
  <package-name>org.w3c.dom.*</package-name>
</prefer-application-packages>
```

Agenda

- Overview of WebLogic's ClassLoaders
- Overview of WebLogic's ClassLoader Analysis Tool
- • Demo of WebLogic's ClassLoader Analysis Tool
- Conclusion & WebLogic Resources

Agenda

- Overview of WebLogic's ClassLoaders
- Overview of WebLogic's ClassLoader Analysis Tool
- Demo of WebLogic's ClassLoader Analysis Tool
- • Conclusion & WebLogic Resources

Find us Online!



www.oracle.com/technetwork/middleware/weblogic



www.YouTube.com/OracleWebLogic



Give us feedback! [@OracleWebLogic](https://twitter.com/OracleWebLogic)

www.twitter.com/OracleWebLogic



www.facebook.com/OracleWebLogic