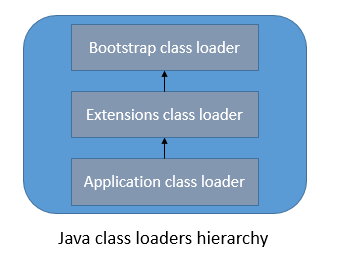
# CHAPTER 11: CLASS LOADERS

## Theory

With the help of the class loaders, WebSphere Application Server enables applications to access available class and resources. Having known the structure and the hierarchy of class loaders will help you to understand and troubleshoot advances environments against errors such as “ClassNotFoundException”.

Each Java class must be loaded by a class loader that locates and loads the required classes. During the startup of a JVM, following class loaders are used:

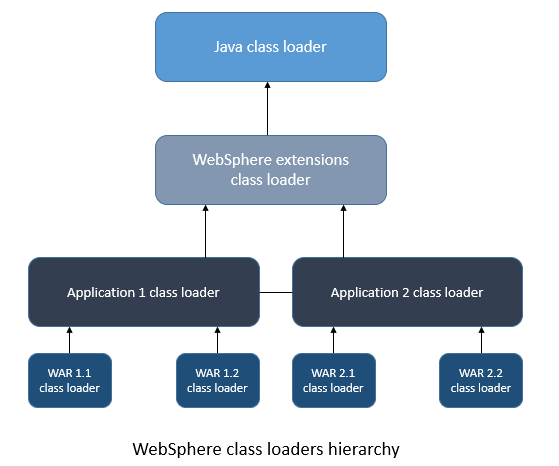
* The *bootstrap class loader*, loads the core Java libraries that are located under “<JAVA\_HOME>/jre/lib”. This class loader is written in native code.
* The *extensions class loader*, loads the code in the extensions directories such as “<JAVA\_HOME>/jre/lib/ext” or any other directory that is specified by the “java.ext.dirs” system property. This class loader is implemented by the “sun.misc.Launcher$ExtClassLoader” class.
* The *application class loader*, loads the code defined in the “java.class.path” that maps to the “CLASSPATH” environment variable. This is implemented by the “sun.misc.Launcher$AppClassLoader” class.



The *WebSphere extensions class loader* is where WebSphere Application Server itself is loaded. The class path that is used by the extensions class loader is retrieved from the “ws.ext.dirs” system property, which is initially derived from the “WAS\_EXT\_DIRS” environment variable set in the “setupCmdLine” script file. Each directory listed in the ws.ext.dirs environment variable is added to the WebSphere extensions class loaders class path, and every .jar file and .zip file in the directory is added to the class path.

Application or web module class loaders load elements of enterprise applications running on the server. These application elements can be:

* Web modules
* EJB modules
* Application client modules
* Resource adapter archives
* Utility JAR files



Each class loader is a child of the previous class loader. WebSphere Application Server provides the ability to change the class loading behavior.

* **Application Server class loaders:** You can configure 2 types of class loader policies,
  + Single: In this case, applications are not isolated and uses a single application loader to load all of the modules, libraries and JAR files that are required by all the applications installed into the JVM.
  + Multiple: All applications are isolated and each of them has its own class loader to load EJB modules, libraries and etc.

You can also configure the class loading mode as ***parent first*** that sets the loading of the classes to its parent class loader before attempting to load classes from its local path, or as ***parent last*** that starts loading classes from its local class path before asking its parent.

* **Application class loaders:** You can set the class loader order and WAR class loader policy within WebSphere Application Server.  
  For class loader order you have 2 options:
  + *Classes loaded with parent class loader first*, sets the loading of classes to its parent class loader before attempting to load the class from its local class path.
  + *Classes loaded with application class loader first,* tells the class loader to start with loading classes from its local class path before asking its parent.

For WAR class order policy, you have the following options:

* *Class loader for each WAR file in application* that creates a separate class loader is assigned to each WAR file.
* *Single class loader for application* single class loader is assigned to all WAR files.
* **Web module class loaders:** You can set the class loader order that specifies the search order of the class whether to look first in the parent class loader or in the application class loader. You can set the order as following:
  + *Classes loaded with parent class loader first*, the class loader searches first the application class loader
  + *Classes loaded with application class loader first*, the class loader searches first in the WAR class loaders to load a class.

## AIM

The aim of the lab exercise is to get familiar with class loader settings in different levels. In order to achieve this goal, you need to perform following tasks:

Task 1: Configure class loaders on server level

Task 2: Configure class loaders on application level

Task 3: Configure web module class loader

# Lab Exercise 11: CLASS LOADERS

|  |
| --- |
| Server Level  Appl. Level  Web Module |

## Configure class loaders on server level

## Configure class loaders on application level

## Configure web module class loader

Server Level

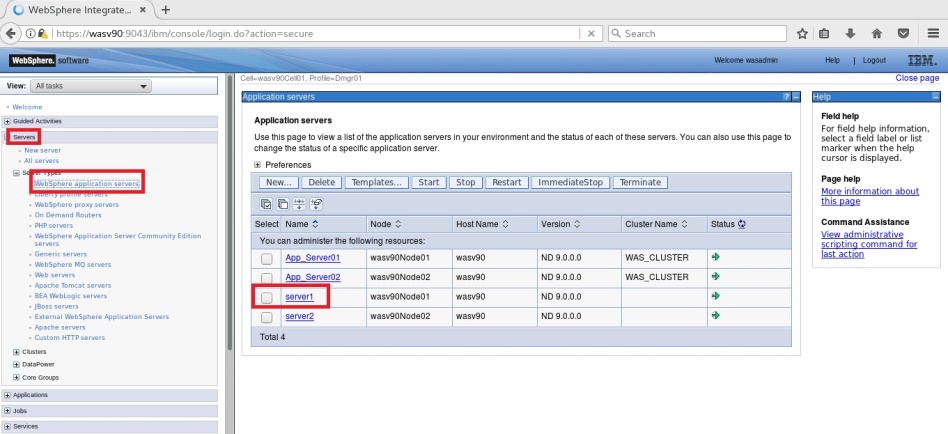
Appl. Level

Web Module



**Task 1: Configure class loaders on server level**

**Step 1:** Navigate to “Servers>Server Types>WebSphere application servers” and then click on the server name, “server1”.

****

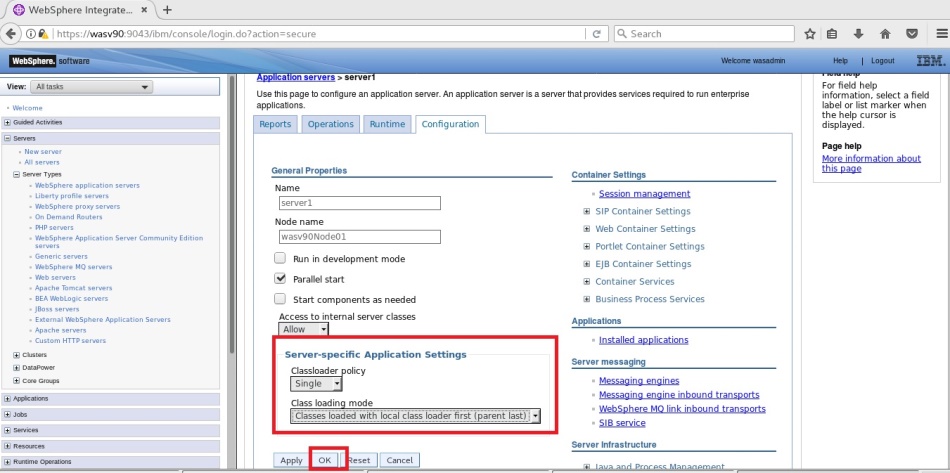
Server Level

Appl. Level

Web Module



**Step 2:** Under the “Server-specific Application Settings”, select “Single” as “Classloader policy” and “Classes loaded with local class loader first (parent last)” as “Class loading policy” and then click “OK”.

****

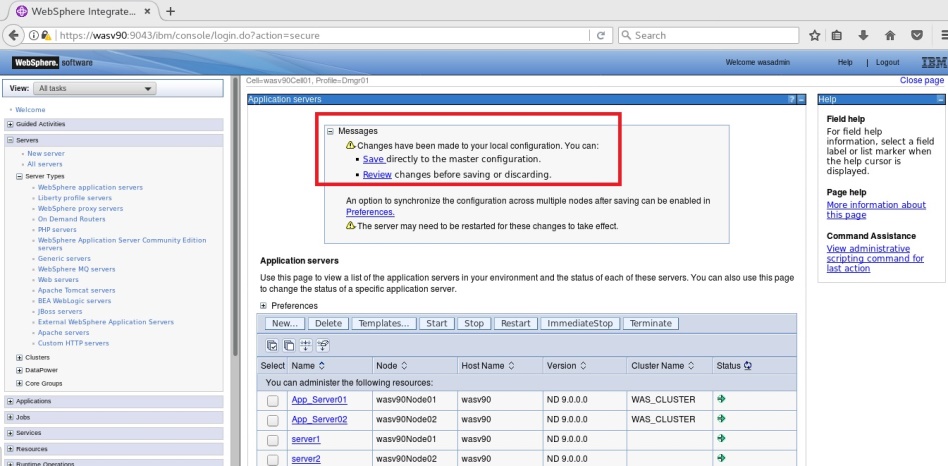
Server Level

Appl. Level

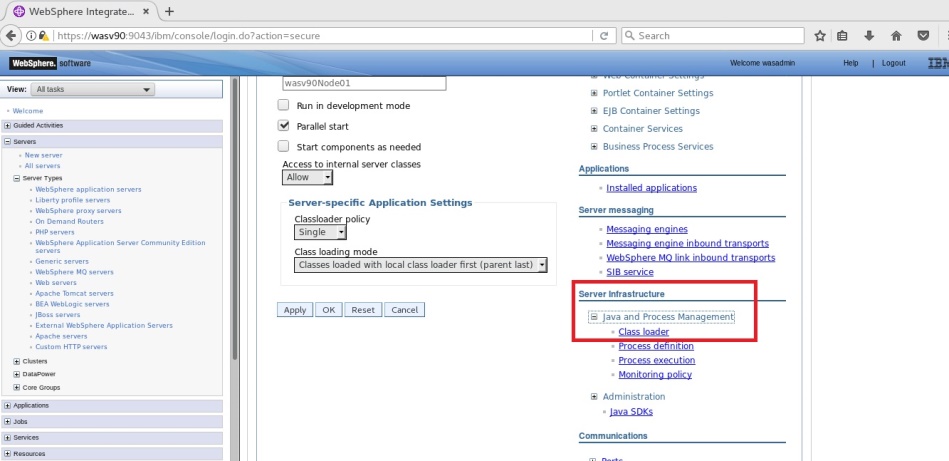
Web Module



**Step 3:** Click “Save” to write changes directly.

****

**Step 4:** Navigate to “Server Infrastructure>Java and Process Management>Class loader”.

****

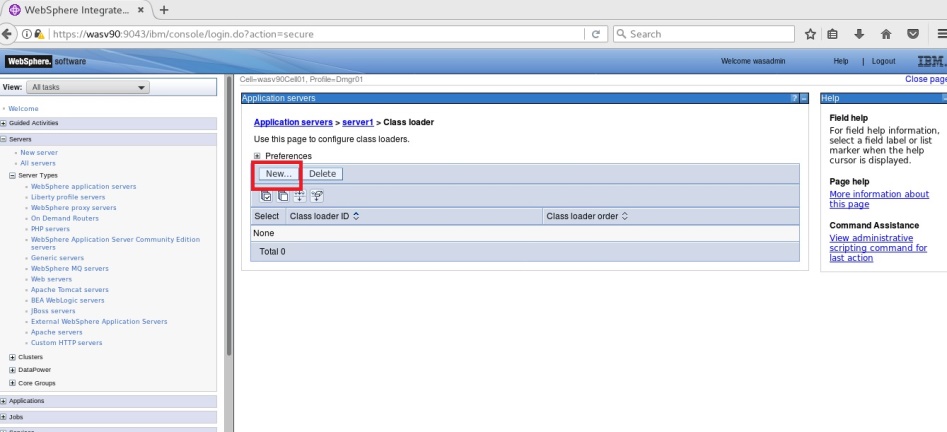
Server Level

Appl. Level

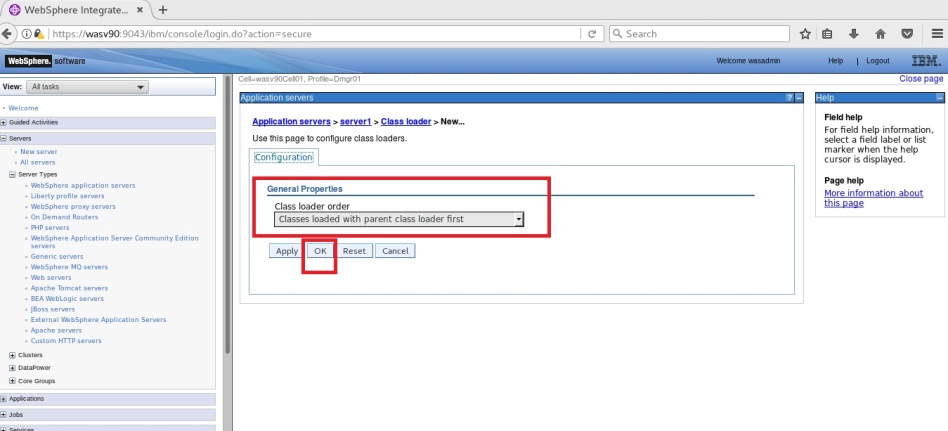
Web Module



**Step 5:** Click “New” to configure new class loader.

****

**Step 6:** Select “Classes loaded with parent class loader first” then click “OK”.

****

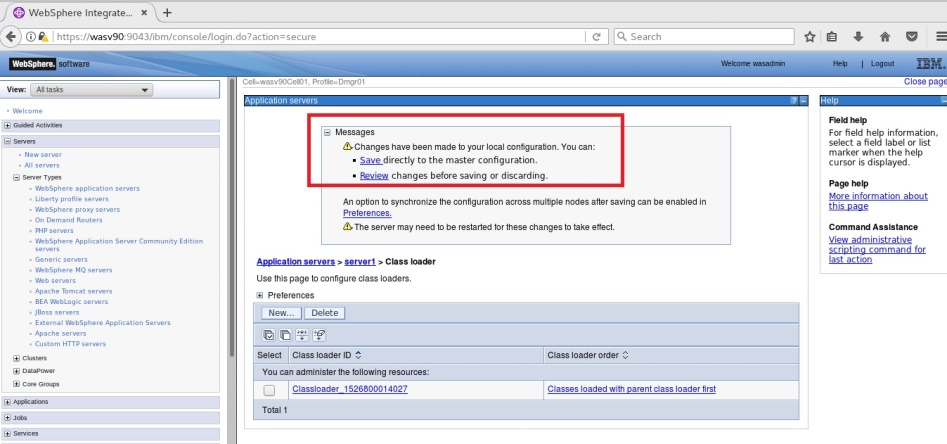
Server Level

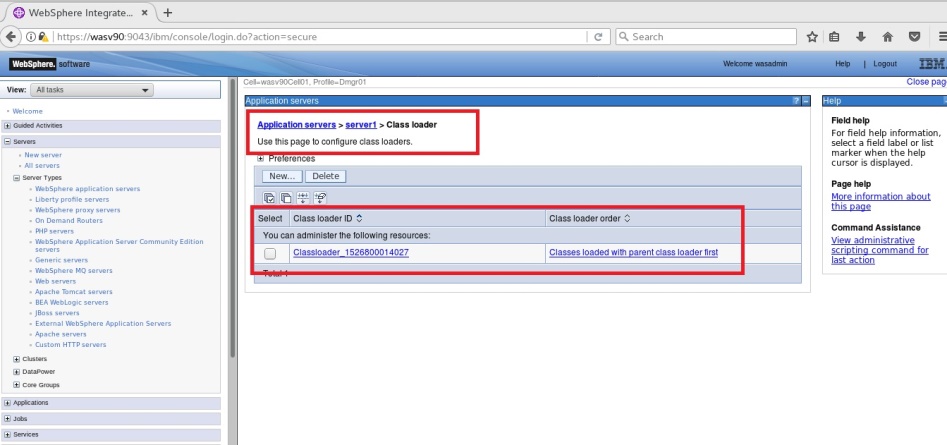
Appl. Level

Web Module



**Step 7:** Click “Save” to write changes to the master file.

****

****

**Task 1 is complete!**

Server Level

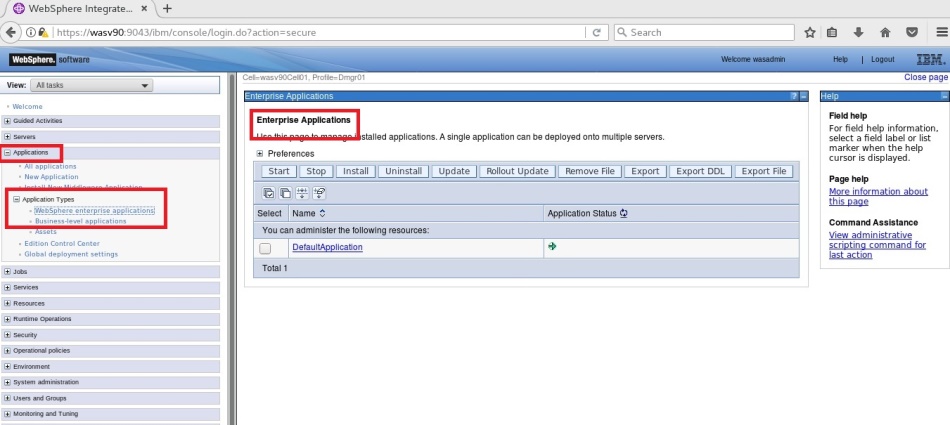
Appl. Level

Web Module



**Task 2: Configure class loaders on application level**

**Step 1:** Navigate to “Applications>Application Types>WebSphere enterprise applications” and then click on “DefaultAPplication”.

****

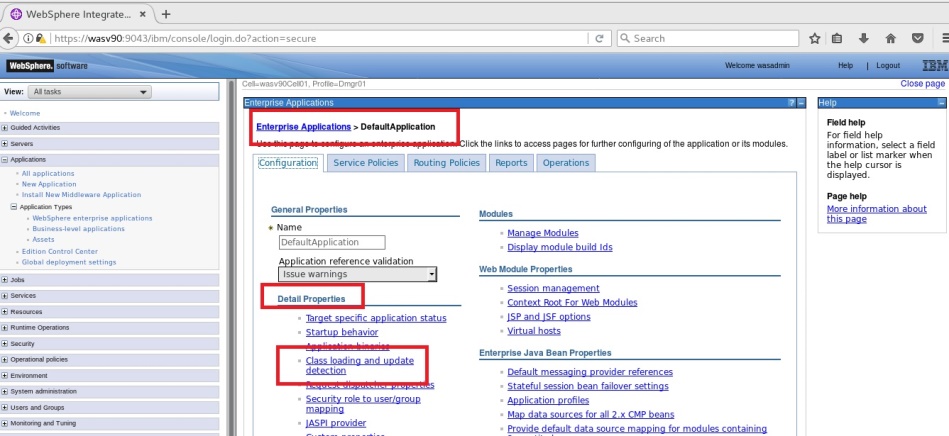
Server Level

Appl. Level

Web Module



**Step 2:** Under “Default Properties”, click on “Class loading and update detection”.

****

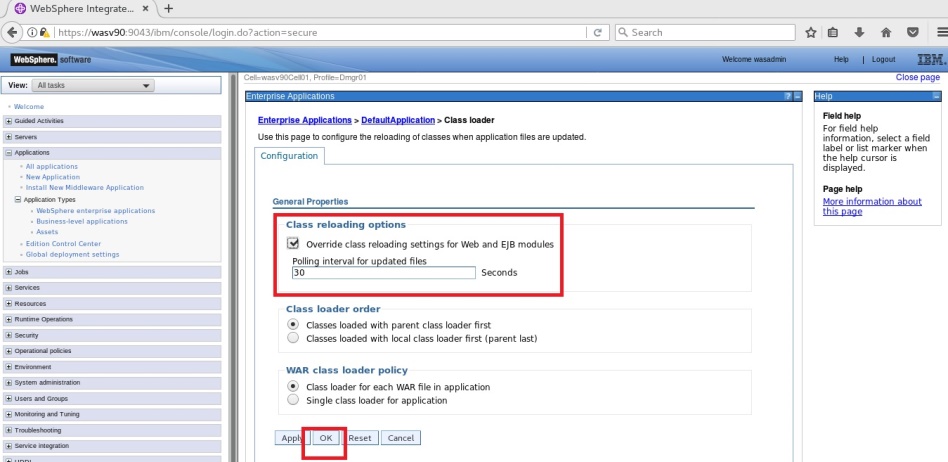
Server Level

Appl. Level

Web Module



**Step 3:** Select “Override class reloading settings for the Web and EJB module” and enter “30” for “Polling interval for updated files”, then click “OK”.

****

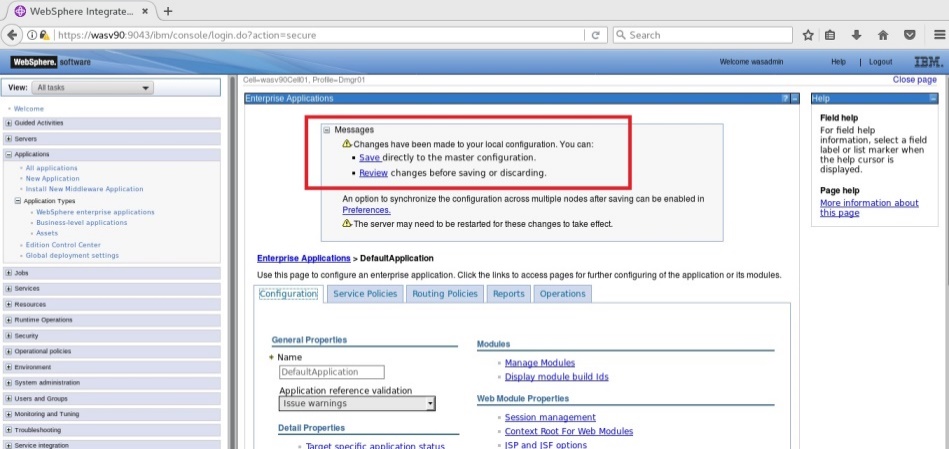
Server Level

Appl. Level

Web Module



**Step 4:** Click “Save” to write changes to master file.

****

**Task 2 is complete!**

Server Level

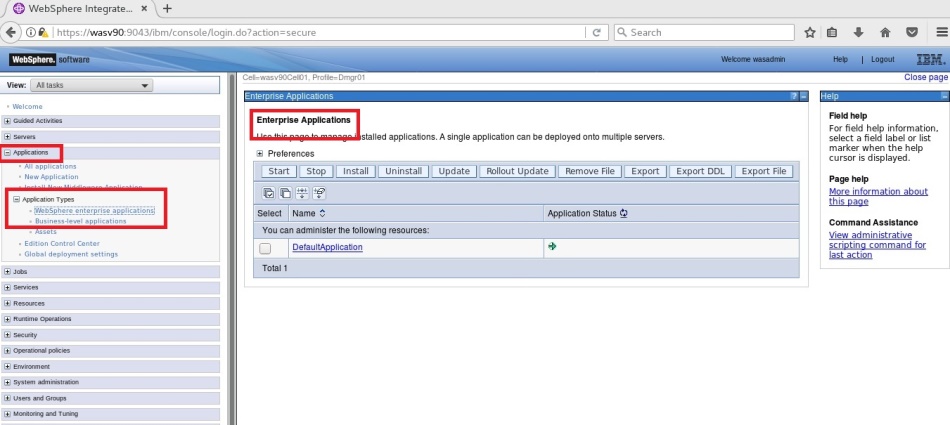
Appl. Level

Web Module



**Task 2: Configure web module class loader**

**Step 1:** Navigate to “Applications>Application Types>WebSphere enterprise applications” and click on “DefaultApplication”.

****

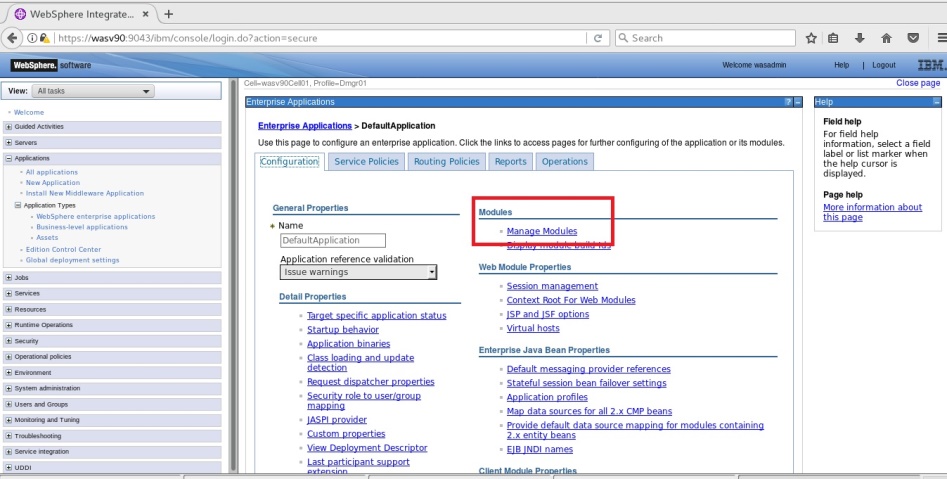
Server Level

Appl. Level

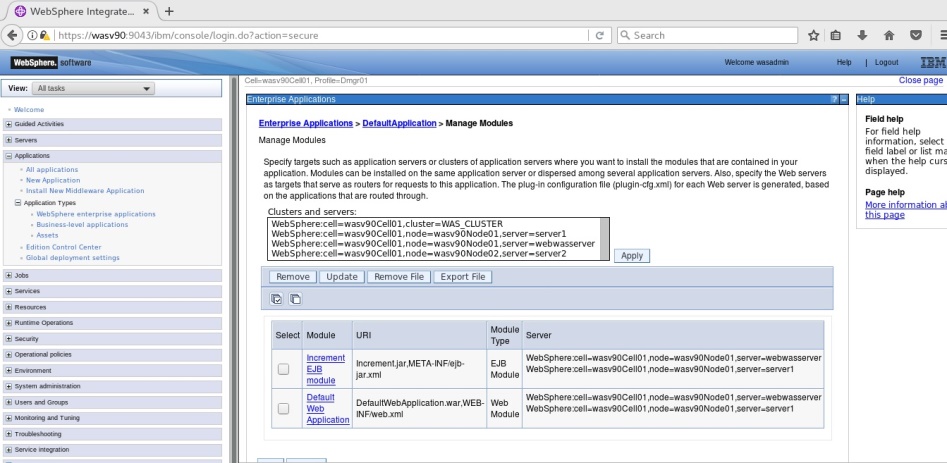
Web Module



**Step 2:** Click on “Manage Modules” under “Modules”.

****

**Step 3:** Click on the web module name, “Default Web Application”.

****

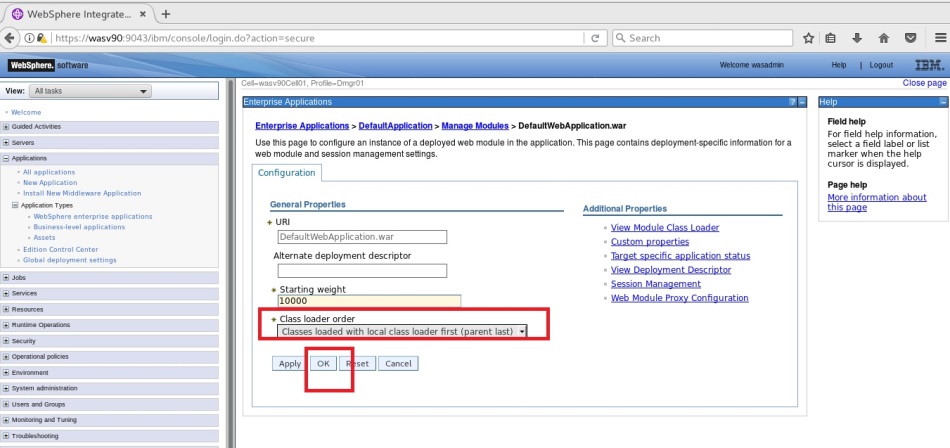
Server Level

Appl. Level

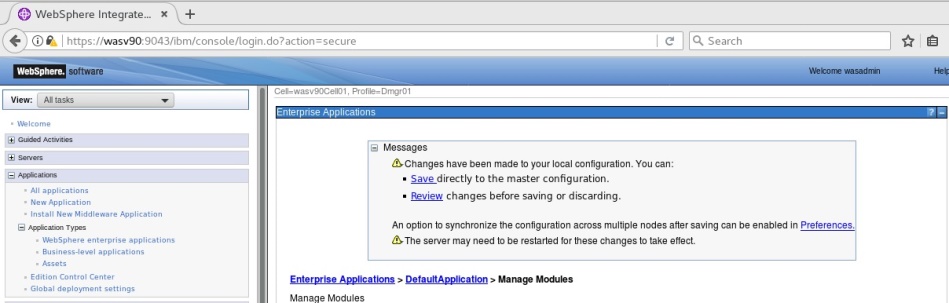
Web Module



**Step 4:** Change the class loader order, “Classes loaded with local class loader first (parent last)”.

****

**Step 5:** Click “Save” to write changes.

****

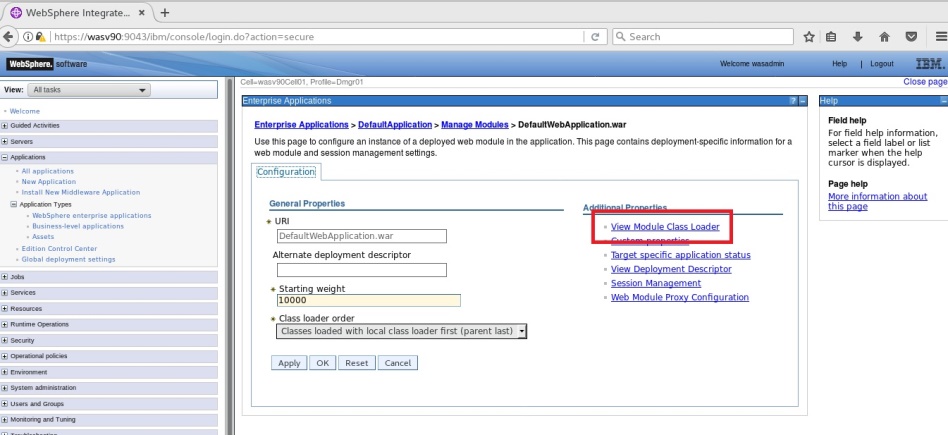
Server Level

Appl. Level

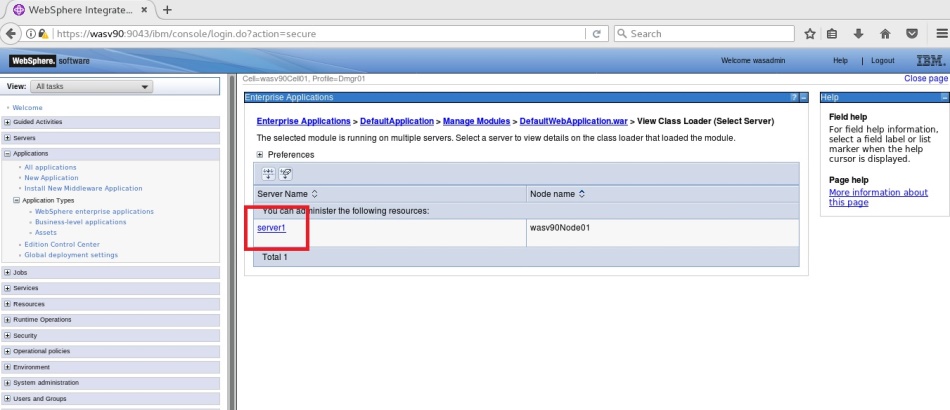
Web Module



**Step 6:** Click on “View Module Class Loader” under “Additional Properties”.

****

**Step 7:** Click on the server name, “server1”.

****

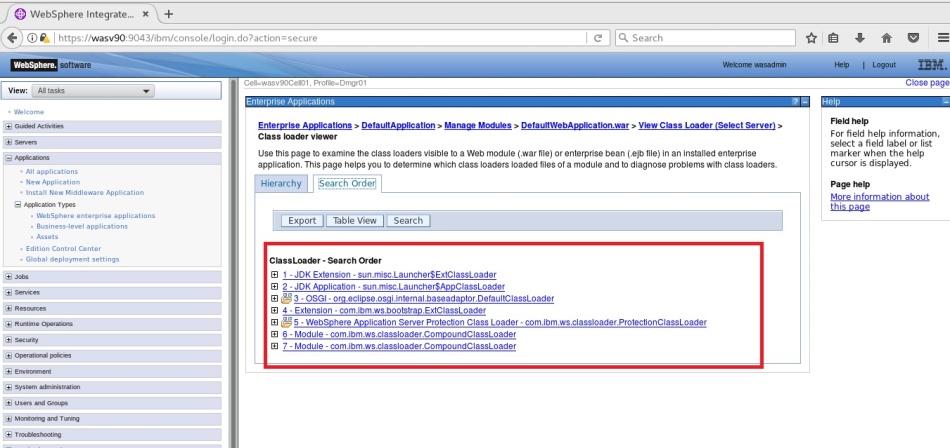
Server Level

Appl. Level

Web Module



**Step 8:** You should see the class loaders loaded files of a module to debug problems with class loaders.

****

**Task 3 is complete!**

# SUMMARY

WebSphere Application Server enables applications to access available class and resources using class loaders. Understand the class loaders will help you to identify and solve the issues in an advanced WebSphere environment. WebSphere Application Server enables you to configure class loaders on server level, application level and web module level.

# REFERENCES

* http://pic.dhe.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=%2Fcom.ibm.websphere.express.doc%2Finfo%2Fexp%2Fae%2Fcrun\_classload.html
* http://pic.dhe.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=%2Fcom.ibm.websphere.express.doc%2Finfo%2Fexp%2Fae%2Furun\_rclassloader\_inst.html
* https://publib.boulder.ibm.com/infocenter/iseries/v5r4/index.jsp?topic=%2Frzamy%2F50%2Fprogram%2Fclsadmcns.htm

# INDEX

application class loader 364

Application class loaders 366

Application Server class loaders 365

bootstrap class loader 364

class loader 364

extensions class loader 364

Web module class loaders 366