

# OSGi-Harmony Runtime

## Introduction To OSGi-ed JRE

### The main purpose

OSGi has become one of the best ways to govern the jar files in a number of applications , such as the power it exhibits in eclipse .And as we all know,the Harmony runtime is a modularized one and each module , with its manifest file , is just like one bundle in the OSGi framework .So we are thinking of governing the modules in Harmony runtime in a OSGi-ed way.

### Ways to meet the main purpose

Naturally, there are two ways to meet this purpose:

- Implementing the OSGi framework in VM in C/C++ to govern all the java modules
- Using one of the OSGi-ed framework implemented in java to implemented a half OSGi-ed JRE.

Finally I picked the plan B with the well known OSGi framework :Felix .And the reasons are as follows :

- Implementing a OSGi-ed framework from scratch in c/c++ in the vm is difficult and time-consuming .
- Plan A suffers from the facts that implementations differs huge from VM to VM .Plan B won't have this kind of problem as long as the modules comply with the OSGi specifications and we have a adapted java.exe , the rest is nearly the same.
- Besides the framework ,each OSGi implementation has a lot of add-ons.This will facilitate future development of new features and utilities .For example : the could use the bundle fileinstall to install all the jar files lies in a folder ,very helpful in certain circumstances.This could be done rapidly if we choose plan B .

### Problems and fixes

The difficulties ,or the main tasks ,of this project are as follows

- A\* Find the minimum working environment of OSGi-framework and modify the rest modules in Harmony ,mainly the manifest file ,to resolve the coupling
- B\* Modify the classloading mechanism and booting of the jre to be able run normal java applications

And the fixes are

#### For A:

Harmony is a modularized runtime .So determine the minimum working environment is no more than choose the right module to stay with the VM . However ,although the coupling is little , it does exist . So a modification of the module is necessary ,mainly the manifest file .And as I want to remain the original structure of Harmony as possible as I can , more contents (classes ) will be included in the minimum environment other than splitting the modules apart .The final jre include the following component :

- luni.jar
- annotation.jar
- security.jar
- math.jar
- text.jar
- regex.jar
- noi\_char.jar
- nio.jar
- logging.jar
- concurrent.jar
- archive.jar
- the ICU4J folder.

The remaining jars are governed by the OSGi framework to handle. One more thing to mention is the two classes,"java.beans.PropertyChangeListener" , "java.beans.PropertyChangeEvent" has to be moved out of beans.jar .This is because of the coupling of beans.jar with other modules in the minimum runtime .Well the good thing is these two classes havn't imports any other classes in the modules governed by the OSGi framework .I think we could include them in any modules lie in the minimum runtime as long as the VM could find them .

#### For B:

I added one extra sphere between the VM and the main class to run :org.apache.osgi.FelixStarter . , the following is done in this class

1. Loading the configurations and Start Felix ,the OSGi framework we use
2. Initializing the class org.apache.osgi.OsgiClassLoader ,this class will load the main class and subsequently all the classes used in the main class
3. Transfer the arguments to the "real " main class ,invoke the main method .

The sequence of class loading is as follows :

1. Check the class name to determine whether it should be loaded by the parent class loader .This work is done by specifying the "org.osgi.framework.bootdelegation" property to include all the package names in the modules stays in the VM .
2. load the classes in the bundle governed by the framework .I will examine each bundle's export and import property to search for the class name ;
3. the classes in the application's class path .
4. we could not find the class and a [ClassNotFoundException](#) will be thrown .

Both the OSGi framework and the Harmony do not allow the define , imports or exports of java.\* classes and packages ,so a lot of examines has been removed to facilitate our purpose .