International Advance Journal of Engineering, Science and Management (IAJESM)
ISSN -2393-8048, Juanuary-June 2015, Submitted in February 2015, iajesm2014@gmail.com

Basic Concept of OSGI Bundles and Its Life Cycle

Mahender Kumar Research Scholar:- Bundelkhand University, Jhansi
Research Guide:- Dr. Rishpal Bangarh Bundelkhand University, Jhansi

Abstract

OSGI bundle is a *Run Time Environment* for Applications which provide the security and Module isolation. The Framework define is group of modularization is called a bundle. The JAR file format and manifest file provide a useful mechanism for assembling collections of java resources along with related meta-information .There are different types of Bundle such as Bundle Manifest, System Bundle, Fragment Bundle , Extension Bundle.

Keywords OSGI ,JAR ,JRE ,Bundle, Eclipse, META-INF

Introduction

OSGI stand for *Open Source Gateway Initiative* .OSGI is a Technology to creating and developing the dynamic software in java. It supports the bundles to creating the server based applications. There are many Technologies related to OSGI. Multiple applications can exist within same container in OSGI.

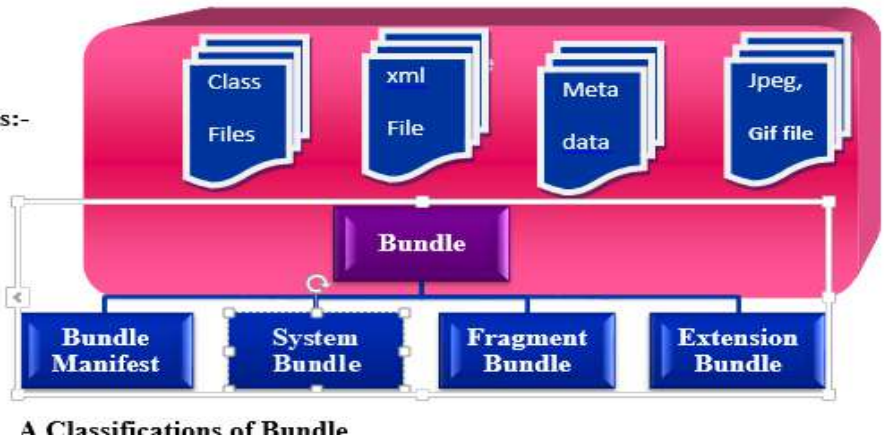
OSGI bundle is a *Run Time Environment* for Applications which provide the security and Module isolation. Eclipse IDE is built on top of OSGI.The Framework define is group of modularization is called a bundle. A bundle can share java package among importer and exporter bundles. A bundle has independent life cycle. It means A software architecture can consist of bundle which can change their implementation (such as start, stop, remove etc)without affecting the other bundle. The OSGI support this type of Architecture. OSGI is a Layer based Architecture.

Bundles

Bundle is a Unit of Modulations and development. In a bundle have java classes and other resources that can be shared between the other bundles .A bundle is deployed java Archive File(JAR).All bundles shares the extension ".jar".

- A jar file contain the resources such as class file of java, html,icons etc to provide functionalities.
- It can be a multi-release jar
- It can also contain optional documentations in OSGI-OPT directory of jar file that is used for to store the source code of a bundle.

There are different bundles:-
(1) Bundle Manifest
(2) System Bundle
(3) Fragment Bundle
(4) Extension Bundle



A Classifications of Bundle

Bundle Manifest:-It consist Manifest file META-INF/MANIFEST.MF. It is a XML documents which contain a list of application package and resources package. All JAR file manifest are required to contain a manifest-version. The JAR file format and manifest file provide a useful mechanism for assembling collections of java resources along with related meta-information.JAR file can be directly loaded and run inside a JRE .There are two attributes that provide information useful for running and linking with other libraries.

(a) Main-class:- it is a main application class which is initialize, load class and run by JRE at run time calling the **"main".** method passing any arguments.

(b) Class-i:-this Attribute is provide the relative URLs to other JAR File for loading the classes

The manifest bundle consist some headers

Manifest-version	**:** 1.0
Bundle-Name	**:** welcome plug-in
Bundle-Manifest Version	**:**2
Bundle Symbolic-Name	**:**p1.welcome
Bundle-version	**:**1.0.0
Bundle-Activator	**:**p1.welcome.welcome
Bundle-Vendor	**:**sgng
Bundle-Localization	**:**plug-in
Import-Package	**:**org.osgi.framework;version="1.3.0"

Explanation of Headers

(a) **Bundle-Name:** It is Name of Bundle"***welcome***" .

(b) Bundle-ManifestVersion:2:- This header tells the OSGI Container that It follow the rules of OSGI Specification.

© **Bundle-SymbolicName:-**it is specifies a unique name for the bundle.

(d) Bundle-Version:- This specifies the version of bundle for example Bundle-version is 1..0.0

(e) **Bundle_activator:-**This bundle specifies to start and stop events in Listener class.It change the lifecycle.

(f) Bundle-Vendor:-It contain the readable description of vendors.

(g) **Bundle-Localization:-** It contain location of file that where the file stored in the bundle. for example welcomes bundle does not contain any locale file but this header is generated by Eclipse IDE .

(h) **Import-Package:-** we can import the package for bundle such as org.osgi.framework

(1) **System Bundle:-**This is OSGI framework itself. The bundle which contain a framework implementation is called *System Bundle.* It is not loaded by framework but it export the java package such as javax.* This bundle is already installed for use. It is a unique bundle which are directly enables for all classes. The system bundle is not permitted to use *import-package* header.

(2) **Fragment Bundle:-**This bundle attached with host bundle. It has no own life cycle but it can provides classes and package to export the other package. This bundle cannot has own activator and class loader. It resolved the Fragment dependencies . if it is not possible then it will not attach with host bundle. The Fragment bundle cannot override the information which is present in the host bundle. This bundle provide to permitted the export and import package and also contain Fragment-Host header. It provide the Bundle-Symbolic Name of the bundle to which this fragment should be attached.

(3) **Extension Bundle:-** This bundles are delivered as fragment bundles to the System Bundle. It can also Export-Package header but it not contain any Require-Bundle header or import-package. It has similar visibility as the system bundle. The Extension bundle are two types:-

(a) *Boot class path :*-This extension bundle gives the permission to deliver the resources that are used in Java Run Time environment boot class loader. This is bundle is optional bundle.

(4) *Framework Extension Bundle:-* It has facility to the capability augment the underlying framework implementation. this bundle is used to provide classes that alter aspects of

how the framework performs class and resource loading. this bundle is used to provide classes that alter aspects of how the framework performs class and resource loading.

Life cycle of Bundle:-

During the new installing bundle, update the bundle or and uninstall the bundle then reinstall the bundle it must be create a unique Identity which should not changed this unique Identity. OSGI Bundle is a Dynamic platform. It allowing to deployment the software components. It has six state to manage the life cycle.

(a) Installed (b) Resolved (c) Starting
(d) Active (e) Stopping (f) Uninstalled

(a) Installed State:-In this stage Bundle is installed successfully. but it is not yet resolved . This stage are performed only required steps such as defining bundle attributes analyzing its manifest file. we can also installed the bundle or reinstalled bundle.

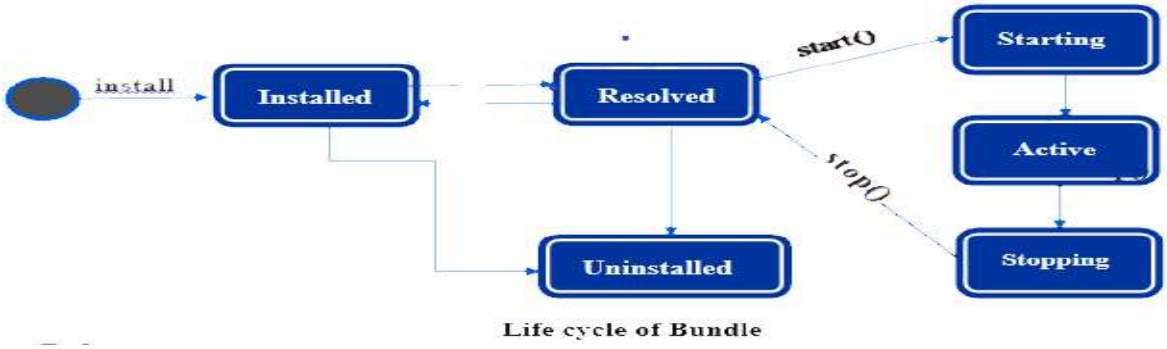
(b) Resolved:-All classes are available in this bundle which are required. This state ready to enter start state.

(c) Starting state:-The BundleActivator.start() methods is called the bundle is started. According to Activation policy means when bundle has lazy policy then bundle will remain in the starting state until bundle is activated.

(d) Active state :- After Starting state it enter into Activate state. The bundle is successfully starting and running, its BundleActivator.start() has been called and returned.

(e) Stopping state:- when BundleActivator.stop() method is called then bundle is stopped. but this method has yet not returned.

(f) Uninstalled state:-This is last state. The bundle is uninstalled from OSGI Container and the bundle can't move into another state.



Life cycle of Bundle

Reference

- OSGI Alliance, OSGI Service Platform Service Compendium Release 4, Version 4.1 http:/www.osgi.org2007.
- OSGI Alliance,"About the OSGI Service Platform, Technical Whitpaper."http://www.osgi.org/documents/colacteral/OSGiTechnical Whitepaper.pdf 2007
- J.McAffer,S.Kaegi,"Eclipse, Equinox and OSGi "JBoss."JBoss OSGi integration project"http://jirra.jboss.com/jira/browse/JBOSGi, 2007
- Building Modular Cloud Apps with OSGi, O'REILLY.
- OSGi in Depth:-By Alex Alves, Published by Manning, Distributed by Simon & Schuster,ebook.
- ebooks. https://en.wikipedia.org/wiki/OSGi
- Knopflerfish OSGi Framework, Knopflerfish.
- O.Gruber,B.J.Hargrave,J.McAffer,P.Rapicault,T.Watson."The Eclipse 3.0 plateform: Adopting OSGi Technology".
- Sun Microsystems, Java Naming and Directory Interface(JNDI).http://java.sun.com/products/jndi/2003.
- Sun Microsystems java 2 standard Edition Extension Mechanism Architecture 1999.