Apache Karaf

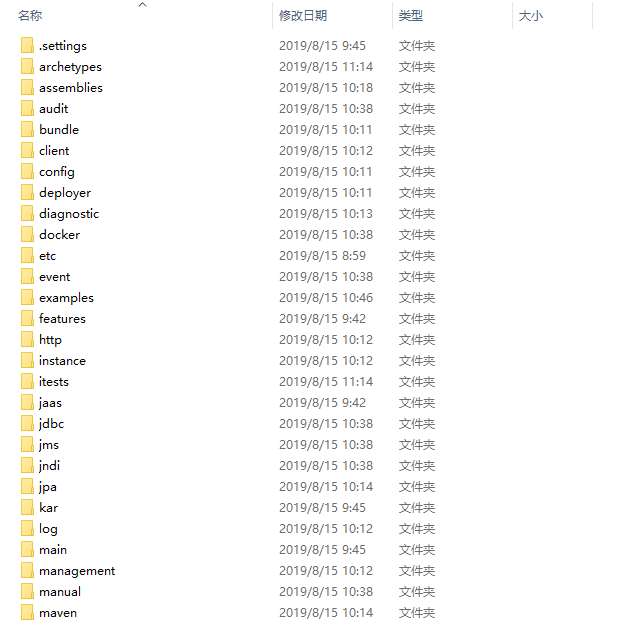
Apache Karaf基于Felix或者Equinox ，其中这两种框架又是在OSGI基础上发展而来（目前使用的大多数版本是6.0.0），在Karaf中基本单元是bundle，许多bundle构成feature，其中feature可以包含其他feature，The feature packaging verifies a features.xml descriptor using the `karaf:verify` goal.最终构成了整个karaf应用。

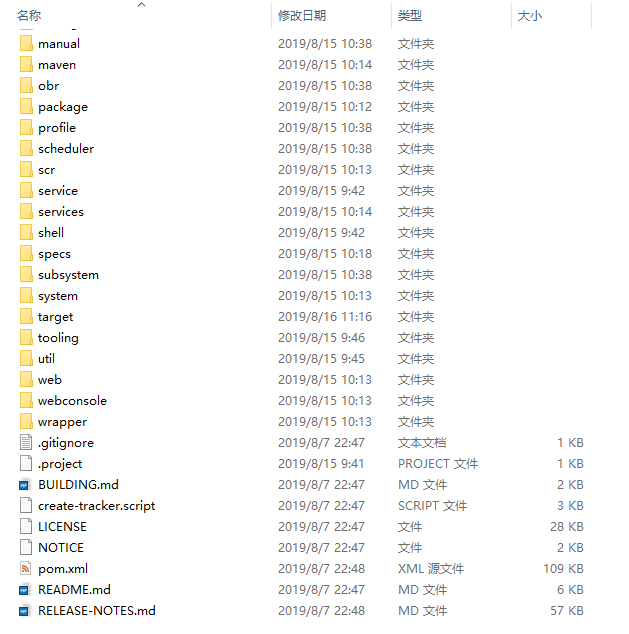
Overview

**编译后二进制目录结构：**



**编译后源代码目录结构（由于目录较多，拆分成两张）：**





Apache Karaf is a modern and polymorphic container.

Karaf can be used standalone as a container, supporting a wide range of applications and technologies.

It also supports the "run anywhere" (on any machine with Java, cloud, docker images, ...) using the embedded mode.

It's a lightweight, powerful, and enterprise ready platform.

With this flexibility, Karaf is the perfect solution for microservices, systems integration, big data, and much more.

Apache Karaf is powered by OSGi (but you don't need to know what OSGi is to use Karaf).其中Apache Karaf由OSGI技术强力支撑，但是你不必知道什么样的OSGI用于其中

Apache Karaf uses either the **Apache Felix or Eclipse Equinox OSGi frameworks**, providing additional features on top of the framework.

Apache Karaf can be scaled from a very lightweight container to a fully featured enterprise service: it's a very flexible and extensible container, covering all the major needs.

Here is a short list of provided features:

**下面列出一些提供地简要特性：**

**热部署：**

**\* \*Hot deployment\*:** simply drop a file in the **`deploy`** directory, Apache Karaf will detect the type of the file and try to deploy it.

也就是说将你自己开发的OSGI应用放入deploy目录,Apache Karaf将会自动检测文件类型并尝试加载它们。

**完整终端：**

**\* \*Complete Console\*:** Apache Karaf provides a complete Unix-like console where you can completely manage the container.

**动态配置：**

**\* \*Dynamic Configuration\*:** Apache Karaf provides a set of commands focused on managing its own configuration.

All configuration files are centralized in the **`etc`** folder. Any change in a configuration file is noticed and reloaded.

Apache Karaf提供了一系列命令来专注管理它自己的配置，所有的配置文件被击中在/etc目录，  
注意并重新加载配置文件中的任何更改。

**高级日志系统：**

**\* \*Advanced Logging System\*:** Apache Karaf supports all the popular logging frameworks (slf4j, log4j, etc). Whichever logging framework you use, Apache Karaf centralizes the configuration in one file.

**服务开通（供应）：**

**\* \*Provisioning\*:** Apache Karaf supports a large set of URLs where you can install your applications (**Maven repository, HTTP,file, etc**). It also provides the concept of "Karaf Features" which is a way to describe your application.

Apache Karaf支持大量URL，您可以在其中安装应用程序（Maven存储库，HTTP，文件等）。它还提供了“Karaf Features”的概念，这是描述您的应用程序的一种方式。

**管理：**

**\* \*Management\*:** Apache Karaf is an enterprise-ready container, providing many management indicators(指标) and operations via JMX.

**远程：**

**\* \*Remote\*:** Apache Karaf embeds an SSHd server allowing you to use the console remotely. The management layer is also accessible remotely.

**安全：**

**\* \*Security\*:** Apache Karaf provides a complete security framework (based on JAAS), and provides a RBAC (Role-Based Access Control) mechanism for console and JMX access.

**多实例管理：**

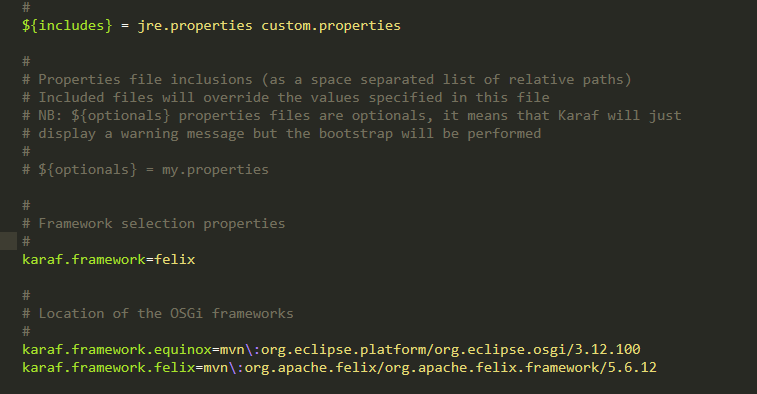
**\* \*Instances\*:** multiple instances of Apache Karaf can be managed directly from a main instance (root).

**OSGI框架选择：**

**\* \*OSGi frameworks\*:** Apache Karaf is not tightly coupled to one OSGi framework. By default, Apache Karaf runs with the Apache Felix

Framework, but you can easily switch to Equinox (just change one property in a configuration file).

默认情况下Apache Karaf运行于Felix框架，但是你可以轻松的切换到Equinox通过仅仅改变一下配置文件(**/etc/config.properties)**



**Quick Start**

These instructions should help you get Apache Karaf up and running in 5 to 15 minutes.

**对于Apache Karaf的运行文件的获取有两种方式：**

1. 通过官网下载二进制文件，这是一个编译好了运行文件
2. 通过官网下载源代码进行编译安装（依赖JAVA,MAVEN,具体版本可以通过README文件确定）

A.解压源代码文件后进入工作目录（如果是通过GIT方式或者的需要确定好版本，默认是master）

1. mvn clean install -DskipTests

C.编译成功后会在工程目录中生成assemblies目录，里面包含了apache-karaf与apache-karaf-minimal子工程目录。一个是标准版本的Karaf，另外一个是最小版本的Karaf。在这两个目录中的target目录下就会生成tar.gz与zip的打包文件，同时也有assembly目录下的未打包文件。

1. 无论是下载的二进制文件，还是自己手动编译后生成的二进制文件最后形同版本的文件是一致的，解压文件后在bin目录下会有针对与不同操作系统（主要是Linux与Windows）的启动文件（其实是不同平台下的shell脚本，用于设置一些环境变量及启动karaf容器）。找到此文件运行就会得到karaf的console控制台。

/ //\_/\_\_\_\_ \_\_\_\_\_\_\_\_\_\_ \_/ \_\_/

/ ,< / \_\_ `/ \_\_\_/ \_\_ `/ /\_

/ /| |/ /\_/ / / / /\_/ / \_\_/

/\_/ |\_|\\_\_,\_/\_/ \\_\_,\_/\_/

Apache Karaf (4.2.6)

Hit '<tab>' for a list of available commands

and '[cmd] --help' for help on a specific command.

Hit '<ctrl-d>' or type 'system:shutdown' or 'logout' to shutdown Karaf.

karaf@root()>

在控制台console中相关的命令可查询官方文档

**Deploy a sample application（部署一个简单的应用）**

对于**karaf应用的开发部署**有多种方式 ：1.通过OSGI插件编写好应用后打包，通过CONSOLE安装部署2.通过OSGI插件编写好应用后打包，放入deploy目录再修改配置文件3.在karaf源码中开发最后编译加载，可以参考karaf源码中的example文件4.修改karaf配置文件动态加载（可以通过URL加载）5.通过裁剪后的karaf容器（parent pom）开发，最后在标准版中安装6.其他方式

While you will learn in the Karaf user's guide how to fully use and leverage Apache Karaf, let's install a sample http://camel.apache.org[Apache Camel] application for now:

Copy and paste the following commands in the console:

----

feature:repo-add camel 2.20.0

feature:install deployer camel-blueprint aries-blueprint

cat > **deploy/example.xml** <<END

<blueprint xmlns="http://www.osgi.org/xmlns/blueprint/v1.0.0">

<camelContext xmlns="http://camel.apache.org/schema/blueprint">

<route>

<from uri="timer://test?fixedRate=true&amp;period=2000" />

<setBody>

<simple>Message at ${date:now:yyyy-MM-dd HH:mm:ss}</simple>

</setBody>

<to uri="log:test" />

</route>

</camelContext>

</blueprint>

The example installed is using Camel to start a timer every 2 seconds and output a message in the log.

The previous commands download the Camel features descriptor and install the example feature.