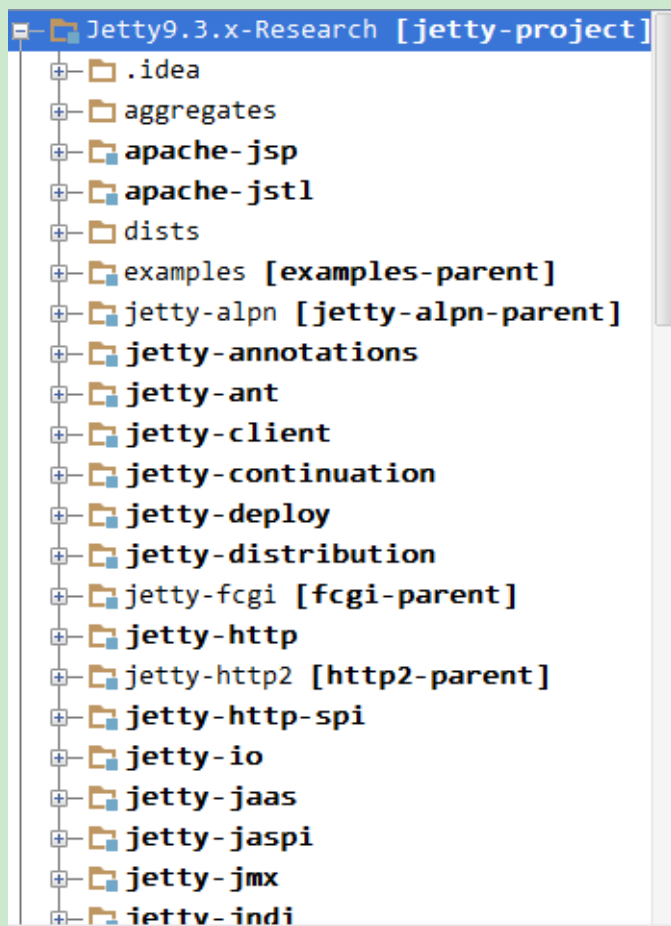


Jetty9.3.x源码下载及构建

源码下载很简单,可直接从github clone出来.

然后进入主目录,看下README.TXT文件,说的很明白,要构建项目的话执行 `mvn clean install` .或者如果你想偷懒的话直接用 IDE 直接导入也行.这里使用的是 IDEA.

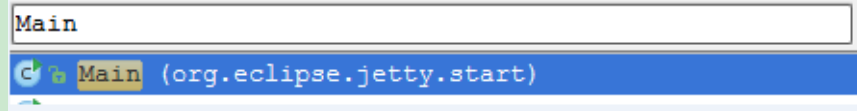
最后的整个项目的结构如下:



接下里所说源码配置是以能够在 IDE 中正常启动 jetty 为前提的.一开始也遇到了一些问题,这里也一并做了记录.写的可能稍微有些啰嗦,(ノωノ).

源码配置过程及遇到的问题

其它的暂时先不管,先将项目运行起来,首先找到 `Main` 类(`start.jar` 中的 Main-Class).如下:



首先我们应该至少要添加两个运行参数: `jetty.home` 和 `jetty.base` .
编译运行.然后发现出现了错误:

```
E:\Java\soft\jdk\jdk1.7.0_45\bin\java ...  
  
WARNING: Nothing to start, exiting ...  
Usage: java -jar start.jar [options] [properties] [configs]  
       java -jar start.jar --help # for more information  
  
Process finished with exit code -2
```

有点蛋疼，此时添加运行参数 `--debug`，再次运行错误就比较明显了：

```
WARNING: Nothing to start, exiting ...
java.lang.ClassNotFoundException: org.eclipse.jetty.xml.XmlConfiguration
    at java.net.URLClassLoader$1.run(URLClassLoader.java:366)
    at java.net.URLClassLoader$1.run(URLClassLoader.java:355) <1 internal calls>
    at java.net.URLClassLoader.findClass(URLClassLoader.java:354)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:425)
    at java.lang.ClassLoader.loadClass(ClassLoader.java:358)
    at org.eclipse.jetty.start.Main.invokeMain(Main.java:224)
    at org.eclipse.jetty.start.Main.start(Main.java:661)
    at org.eclipse.jetty.start.Main.main(Main.java:87) <5 internal calls>

Usage: java -jar start.jar [options] [properties] [configs]
       java -jar start.jar --help # for more information
```

看下 `Main` 类的配置：

Name: ☐ Share ☐ Single instance

Configuration | Code Coverage | Logs

Main class:

VM options:

Program arguments:

Working directory:

Environment variables:

Use classpath of module:

☐ Use alternative JRE:

`jetty-start` 模块不依赖于其它的模块,很显然,我们肯定是无法加载 `XmlConfiguration` 类的,此类在 `jetty.xml` 模块下。

接下来就说明如何解决这些问题,以及源码配置的几种方式。

源码配置的几种方式

修改 `Classpath` 类的源代码

要注意到上面的方式本不是我们最初的想法,我们当然还是希望启动 `org.eclipse.jetty.start.Main`。安装 `jetty` 后我们也是以 `start.jar` 运行的,其实就是运行 `org.eclipse.jetty.start.Main` 类,如何解决该问题?

这里提供一种解决方案: 将其它模块的 `classpath` 加进来(`jetty-start` 中的 `Main` 类加载 `org.eclipse.jetty.xml.XmlConfiguration` 是通过另外的类加载器进行加载的)。

在 `Main` 的 `start()` 方法中。

```
1. // Set current context class loader to what is selected.
2. ClassLoader cl = classpath.getClassLoader();
3. Thread.currentThread().setContextClassLoader(cl);
```

此时我们需要做的就是修改下 `org.eclipse.jetty.start.Classpath#getClassLoader()` 方法。

```
1. // XXX 加上其他项目的classpath -> 在 IDE 中进行 debug
2. URL[] new_urls = addOtherModuleClassPath(urls);
3. return new Loader(new_urls, parent);
```

```

4.
5.    //return new Loader(urls, parent);

```

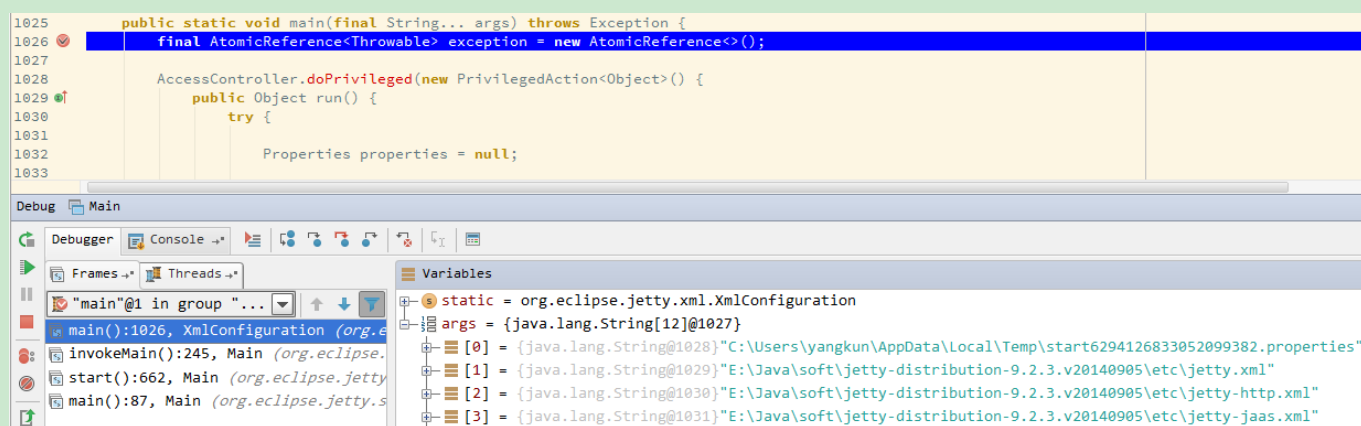
addOtherModuleClassPath() 方法也比较简单(当然这里不必加载全部的,你也可以省略一部分):

```

1.    private URL[] addOtherModuleClassPath(URL[] urls) {
2.        // user.dir 为 "jetty.project"
3.        String[] otherModules = {
4.            "jetty-ant",
5.            "jetty-util",
6.            "jetty-jmx",
7.            "jetty-io",
8.            "jetty-http",
9.            ...
10.           "jetty-monitor",
11.           "jetty-http-spi"
12.        };
13.        String classes = "\\target\\classes\\";
14.
15.        URL[] new_urls = new URL[urls.length + otherModules.length];
16.        try {
17.            for (int i = 0; i < otherModules.length; i++) {
18.                String path = otherModules[i] + classes;
19.                File file = new File(path);
20.                URL url = file.toURI().toURL();
21.                new_urls[i] = url;
22.            }
23.
24.            System.arraycopy(urls, 0, new_urls, otherModules.length, urls.length);
25.        } catch (Exception e) {
26.            e.printStackTrace();
27.        }
28.
29.        return new_urls;
30.    }

```

运行时先将其它的模块编译好.此时尝试下打断点,现在就可以从 **jetty.start** 模块 debug 跳转到 **jetty.xml** 模块了.如下图:



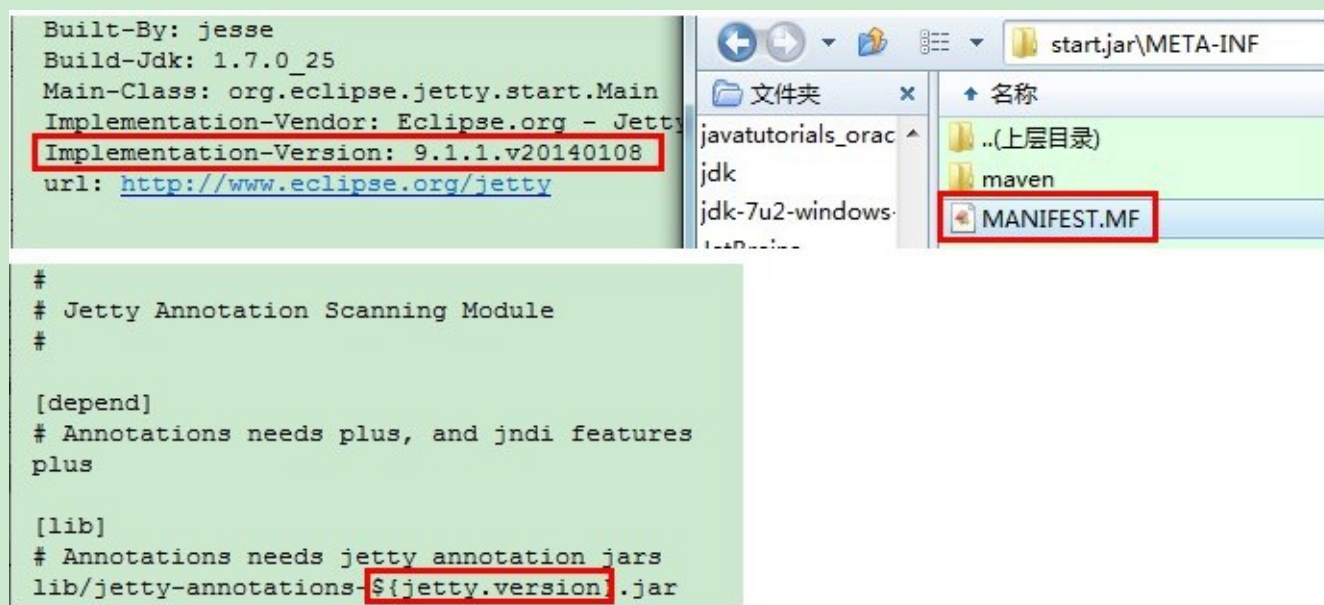
到这里我们已经完全可以调试和运行jetty了,不过你可能会有一个疑问:我们在命令行下使用 **java -jar start.jar** 是完全可以加载其他模块的啊(这里都是加载 **{jetty.home}/lib** 目录下的一个个jar文件,比如: **jetty-xml-9.1.1.v20140108.jar**)!为什么这里就不行了.就算不能直接加载其其它模块的 **classes** 文件(这里指的是加载 **...\jetty.project\jetty-xml\target\classes** 目录下的 **class** 文件),但是至少应该可以加载 **{jetty.home}/lib** 目录下的其它模块的jar文件才对.对于这个问题确实要先去读源代码.

我们看下两种方式的启动(这个图中的版本比当前所调试的低一点,这不是重点,(* ^_^)):

```
E:\Java\soft\jetty-distribution-9.1.1.v20140108>java -jar start.jar
2014-02-23 21:18:52.086:INFO:oejs.Server:main: jetty-9.1.1.v20140108
2014-02-23 21:18:52.241:INFO:oejdp.ScanningAppProvider:main: Deployment monitor [file:/E:/Java/soft/jetty-distrib
0108/webapps/] at interval 1
2014-02-23 21:18:52.533:INFO:oejs.ServerConnector:main: Started ServerConnector@5d84d309{HTTP/1.1}{0.0.0.0:8080}
2014-02-23 21:19:11.858:INFO:oejs.ServerConnector:Thread-0: Stopped ServerConnector@5d84d309{HTTP/1.1}{0.0.0.0:8080}

2014-02-23 21:21:53.720:INFO:oejs.Server:main: jetty-TEST
2014-02-23 21:21:54.383:INFO:oejdp.ScanningAppProvider:main: Deployment monitor [file:/E:/Java/soft/jetty-distrib
2014-02-23 21:21:54.911:INFO:oejs.ServerConnector:main: Started ServerConnector@6a71e64d{HTTP/1.1}{0.0.0.0:8080}
```

很明显你已经看出了,打印的版本是不一样的,我们在 idea 中配置运行后 jetty 的版本变成了 Test .什么原因导致了这样?查看 StartArgs 类中的静态代码块,再结合下图就可以理解了.具体的可以看下 StartArgs 类中的注释.



```
Built-By: jesse
Build-Jdk: 1.7.0_25
Main-Class: org.eclipse.jetty.start.Main
Implementation-Vendor: Eclipse.org - Jetty
Implementation-Version: 9.1.1.v20140108
url: http://www.eclipse.org/jetty

#
# Jetty Annotation Scanning Module
#

[depend]
# Annotations needs plus, and jndi features
plus

[lib]
# Annotations needs jetty annotation jars
lib/jetty-annotations-${jetty.version}.jar
```

嵌入式方式调试

当然你也不必像上面那么费劲,可以运行 examples\embedded 中的示例,这样也可以进行调试.如果你需要一个完整的过程,那么可以使用上面的方式.

关于 Jetty 的文档

不得不稍微吐槽下 jetty 的文档,虽然也有源码配置的章节,可说的实在太简单了点.还有如果你 clone 出 jetty 的文档,将其编译为 html 格式的,你会发现也存在问题.首先当然是漫长的 jar 包下载过程,结束后编译,就会出现如下的问题:

```
Not Cached:
http://git.eclipse.org/c/jetty/org.eclipse.jetty.project.git/plain/examples/embedded/src/main/java/
org/eclipse/jetty/embedded/ExampleServer.java
Error at xsl:variable on line 329 of file:/E:/Java/git/github/jetty-documentation/src/docbkx-stylesheet/h
tml/docbook.xsl:
Exception in extension function java.io.IOException: 文件名、目录名或卷标语法不正确。
```

具体也就是在如下的地方发生了错误: jetty-documentation\src\docbkx-stylesheet\html\docbook.xsl 中发生了错误:

```

325     <xsl:when test="@language='rjava'">
326       <xsl:attribute name="class">brush: java<xsl:copy-of select="$brushstyle"/></xsl:attribute>
327       <xsl:variable name="filename" select="./d:filename"/>
328       <xsl:variable name="methodname" select="./d:methodname"/>
329       <xsl:variable name="newText" select="ifetch:fetch($filename,$methodname)"/>
330       &lt;![CDATA[<xsl:value-of select="$newText"/>]]&gt;
331     </xsl:when>

```

然后看看前面也有类似的写法,想不通!不过注意到了-一个地方, `<xsl:when test="@language='rjava'">` 中的 `@language='rjava'`,目前也不清楚是干嘛的.先不用想太多,将 `@language=` 不是 `###java###` 的全都注释掉.再运行 `mvn clean compile`.编译成功,html 文档都正常生成了.

```

[INFO] Building Jetty :: Documentation 9.2.3.v20140905
[INFO] -----
[INFO]
[INFO] --- maven-enforcer-plugin:1.1.1:enforce (enforce) @ jetty-documentation ---
[INFO]
[INFO] --- xml-maven-plugin:1.0:validate (default) @ jetty-documentation ---
[INFO]
[INFO] --- maven-resources-plugin:2.5:copy-resources (filter-docbook-sources) @ jetty-documentation ---
[debug] execute contextualize
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Copying 297 resources
[INFO]
[INFO] --- docbkx-maven-plugin:2.0.14:generate-html (html) @ jetty-documentation ---
[INFO] Executing tasks
[INFO] Executed tasks
[INFO] Processing input file: jetty.xml
[INFO] Dumping to E:\Java\git\github\jetty-documentation\target\docbkx\generated\gen\jetty.xml
[INFO] Applying customization parameters
[INFO] Chunking output.
[INFO] See E:\Java\git\github\jetty-documentation\target\docbkx\html\jetty for generated file(s)
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----

```

不过文档中有些地方是 `??`, 对照了下在线的文档,存在同样的问题.

最后

当我在另外一台机器上将 Jetty9.3.x-Research clone 出来,用 IDEA 配置的时候,还是出现了一些问题,最主要的问题就是当编译运行 `org.eclipse.jetty.start.Main` 的时候,仍然出现了 `org.eclipse.jetty.xml.XmlConfiguration` 找不到的问题,查看原因发现, `jetty-xml` 根本就没有被编译,此模块下的 `target\classes` 目录下也就不存在 class 文件.囧!最简单的办法就是 `Build -> Rebuild Project`.

最后你可能发现虽然整个项目运行起来了,可我们并没有在浏览器中运行示例程序,将参数 `jetty.base` zhixiang `jetty.base=E:\Java\soft\jetty-distribution-9.1.1.v20140108\demo-base`.此时打开浏览器,输入 `http://localhost:8080/`,一切都 OK 了.