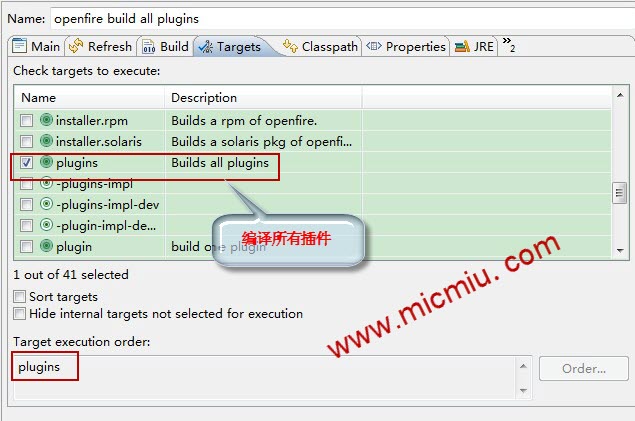
摘自网络博客，地址：

http://www.micmiu.com/opensource/openfire/openfire-plugins-build/

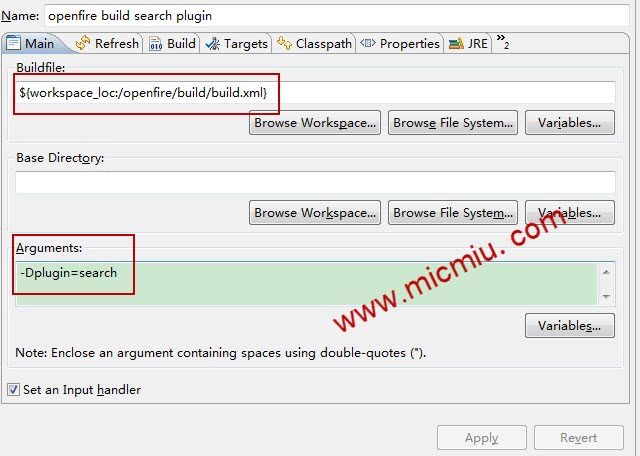
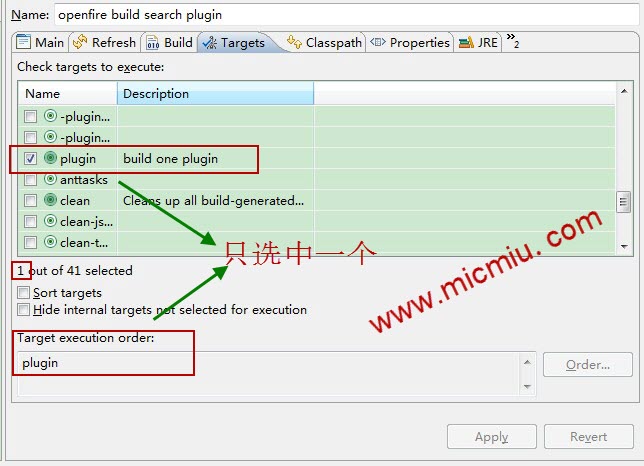
本文主要是介绍Openfire插件开发的环境配置，该指南是在你已经完成了Eclipse中Openfire源码开发环境配置（详见：<http://www.micmiu.com/opensource/openfire/openfire-src-config/>）的基础之上的。

* 配置编译所有插件
* 配置编译一个已存在的插件
* 配置编译自己的插件
* 如何运行和调试插件

[一]、配置编译所有插件

* 点击  Run –> External Tools –> Open External Tools Dialog….
* 选择 Ant Build. 右击 Ant Build 选择 New. 创建一个新的配置
* 把新配置的默认名称：New\_configuration 修改成：openfire build all plugins .
* 选择Main 标签，点击 Buildfile –> Browse Workspace…. 弹出 Choose Location 窗口
* Choose Location 窗口 左边 选择 openfire –> build ，右边窗口选择  build.xml ，点击OK 返回主配置界面
* 选择Targets 标签，选中名称为：plugin**s** 的target，其他全部取消选中。[](http://www.micmiu.com/wp-content/uploads/2013/03/openfire-plugins-build-01.jpg)
* 选择Common 标签，在 Display in favorites menu 下选中 the External Tools，便于以后快捷编译插件
* 点击主配置界面中的 Apply 和 Close 按钮
* 你只要点击：Run –> External Tools –> openfire build all plugins （和之前配置的名称一致）即可执行编译所有的插件

[二]、配置编译一个已存在的插件（以 search 插件为例）

* 点击  Run –> External Tools –> Open External Tools Dialog….
* 选择 Ant Build. 右击 Ant Build 选择 New. 创建一个新的配置
* 把新配置的默认名称：New\_configuration 修改成：openfire build search plugins .
* 选择Main 标签，点击 Buildfile –> Browse Workspace…. 弹出 Choose Location 窗口
* Choose Location 窗口 左边 选择 openfire –> build ，右边窗口选择  build.xml ，点击OK 返回主配置界面
* 在Main 标签中的 Arguments 设置中添加参数配置：-Dplugin=search 。 [](http://www.micmiu.com/wp-content/uploads/2013/03/openfire-plugins-build-02.jpg)
* 选择Targets 标签，选中名称为：plugin （区别编译由于插件的target：plugin**s**）的，其他全部取消选中。[](http://www.micmiu.com/wp-content/uploads/2013/03/openfire-plugins-build-03.jpg)
* 选择Common 标签，在 Display in favorites menu 下选中 the External Tools，便于以后快捷编译插件
* 点击主配置界面中的 Apply 和 Close 按钮
* 之后只要点击：Run –> External Tools –> openfire build search plugins 即可执行编译一个指定的插件
* 编译好后，刷新项目会在 openfire\_src\target\openfire\plugins 看到已经打包好的插件
* 编译时一定要注意编码！

[三]、配置编译自己的插件

自定义插件，最基本要满足如下两个条件：

* 根目录下要包含：plugin.xml
* 根目录下包含一个java源码文件夹，同时要有一个实现了org.jivesoftware.openfire.container.Plugin 接口的java类。

插件开发的详细参见：<http://www.igniterealtime.org/builds/openfire/docs/latest/documentation/plugin-dev-guide.html>

本文自定义一个 mmlog 的插件，结构如下：

..\openfire\_src\src\plugins\mmlog

│ plugin.xml

│

└─src

└─java

└─com

└─micmiu

└─openfire

└─plugins

└─mmlog

MMLogPlugin.java

plugin.xml :

|  |  |  |
| --- | --- | --- |
| 1 | <?xml version="1.0" encoding="UTF-8"?> | |
| 2 |  |

|  |  |
| --- | --- |
| 3 | <plugin> |
| 4 | <class>com.micmiu.openfire.plugins.mmlog.MMPlugin</class> | |

|  |  |
| --- | --- |
| 5 | <name>mmlog</name> |
| 6 | <description>This is an example plugin.</description> | |

|  |  |  |
| --- | --- | --- |
| 7 | <author>Michael</author> | |
| 8 | <version>1.0</version> |

|  |  |  |
| --- | --- | --- |
| 9 | <date>3/19/2013</date> | |
| 10 | | <minServerVersion>3.0.0</minServerVersion> | |

|  |  |
| --- | --- |
| 11 |  |
| 12 | <adminconsole> | |

|  |  |  |
| --- | --- | --- |
| 13 | </adminconsole> | |
| 14 | </plugin> |

MMLogPlugin.java ：

|  |  |  |
| --- | --- | --- |
| 1 | package com.micmiu.openfire.plugins.mmlog; | |
| 2 |  |

|  |  |  |
| --- | --- | --- |
| 3 | import java.io.File; | |
| 4 |  |

|  |  |
| --- | --- |
| 5 | import org.jivesoftware.openfire.container.Plugin; |
| 6 | import org.jivesoftware.openfire.container.PluginManager; | |

|  |  |
| --- | --- |
| 7 |  |
| 8 | /\*\* | |

|  |  |  |
| --- | --- | --- |
| 9 | \* | |
| 10 | | \* @author <a href="[http://www.micmiu.com](http://www.micmiu.com/)">Michael Sun</a> | |

|  |  |
| --- | --- |
| 11 | \*/ |
| 12 | public class MMLogPlugin implements Plugin { | |

|  |  |
| --- | --- |
| 13 |  |
| 14 | @Override | |

|  |  |  |
| --- | --- | --- |
| 15 | public void initializePlugin(PluginManager manager, File pluginDirectory) { | |
| 16 | System.out.println("===> initializePlugin ..."); |

|  |  |
| --- | --- |
| 17 |  |
| 18 | } | |

|  |  |
| --- | --- |
| 19 |  |
| 20 | @Override | |

|  |  |
| --- | --- |
| 21 | public void destroyPlugin() { |
| 22 | System.out.println("===> destroyPlugin ..."); | |

|  |  |
| --- | --- |
| 23 |  |
| 24 | } | |

|  |  |  |
| --- | --- | --- |
| 25 |  | |
| 26 | } |

其他的编译配置的步骤和上面  第[三] 部分介绍编译一个已存在的插件的过程一致。

编译好后，刷新项目会在 openfire\_src\target\openfire\plugins 看到已经打包好的插件：mmlog.jar

[](http://www.micmiu.com/wp-content/uploads/2013/03/openfire-plugins-build-04.jpg)

[四]、如何运行和调试插件

一个插件本身不会运行，它会随着Openfire 服务端程序的的启动而启动，不管是运行还是调试都需要依赖Openfire 服务端主程序。

[五]、参考文档

* <http://community.igniterealtime.org/docs/DOC-1200>
* <http://www.igniterealtime.org/builds/openfire/docs/latest/documentation/plugin-dev-guide.html>
* <http://www.micmiu.com/opensource/openfire/openfire-src-config/>