

hihsoft 手拉手工具篇

个性化解决方案的核心

-----业务基础平台

文件状态： 【】 草稿 【】 正式发布 【√】 正在修改中	文件标识	hihsoft-flat-000009		
	文档名称	业务基础平台开发环境总体介绍		
	版本	日期	作者	备注说明
	1.0	2013-8-6	hihsoft	
	密级	公开		
	应用范围	喜欢hihsoft业务基础平台的拉丝们		
	版权所有	JavaHih为广大技术人员提供软件组件化架构和设计的交流平台，集思广益，打造良好的组件化产品，并有专业人士对文档进行校审，文档以LGPL等开放协议发布，允许并鼓励在网上转载及发布，并寻求与更多技术社区和网站的合作推广，该文档的所有权属于JavaHih手拉手研发团队。		

目 录

1 业务基础平台开发环境概述.....	5
1.1 编写目的.....	5
1.2 参考资料.....	5
1.3 环境概述.....	5
1.4 环境搭建清单.....	6
1.5 环境快速搭建.....	6
1.5.1 下载及安装 JDK.....	6
1.5.2 M2E 插件安装.....	9
1.6 项目结构介绍.....	9
1.7 项目编译运行.....	10
2 ECLIPSE 安装和配置篇.....	12
2.1 总体介绍.....	12
2.2 ECLIPSE 下载及安装.....	13
2.3 ECLIPSE 配置.....	14
3 ECLIPSE 插件篇.....	22
3.1 常用插件安装.....	22
3.1.1 属性编辑器.....	22
3.1.2 tomcat 插件.....	22
3.1.3 编辑器的列表.....	22

3.1.4	支持 freemarker 编程的插件	22
3.1.5	eclipse 样式(皮肤)切换的插件 : eclipseskins	23
3.1.6	反编译插件 : jadclipse	23
3.1.7	支持编辑器全屏插件 :	23
3.1.8	Eclipse 加速插件 KeepResident	23
3.1.9	JBoss-IDE 有许多开发的插件 , 可以自己按需选择	23
4	ECLIPSE 快捷键篇	23
4.1	常用快捷键	23
4.1.1	Ctrl+Space	23
4.1.2	Ctrl+Shift+Space	24
4.1.3	Ctrl+/.	24
4.1.4	Ctrl+Shift+/.	24
4.1.5	Ctrl+Shift+ 	24
4.1.6	Ctrl+Shift+F	24
4.1.7	Ctrl+1	24
4.1.8	Ctrl+F6	24
4.1.9	Ctrl+Shift+M	25
4.1.10	Ctrl+Shift+O	25
4.1.11	Ctrl+Alt+S	25
4.1.12	Ctrl+H	25
4.1.13	Ctrl+D	25

4.1.14	Ctrl+Shift+G	25
5	ECLIPSE 调试篇.....	26
5.1	ECLIPSE 调试快捷键	26
5.2	调试方法	26
5.2.1	条件断点	26
5.2.2	变量断点	27
5.2.3	方法断点	28
5.2.4	重新调试	29
5.2.5	远程调试	30
5.2.6	异常断点	32
6	ECLIPSE 编译打包篇.....	33
6.1	ECLIPSE 编译	33
6.2	ECLIPSE 打包	33
7	交流、反馈、参与贡献.....	36

1 业务基础平台开发环境概述

1.1 编写目的

本文档主要对 hihsoft 项目的开发环境搭建做相关说明,是体验 HihSoft 业务基础平台的第一步,正所谓“磨刀不误砍柴工”,搭建好环境成为必备本领。

本文将从 JDK 的安装、MAVEN 安装及 Eclipse 工具介绍及使用技巧、优化等角色做详细介绍,使你真正驾驭 J2EE 开发工具。

1.2 参考资料

《计算机软件产品开发文件编制指南 (GB8567-88) 》

《JHIH 手拉手项目文档规范》

1.3 环境概述

hihsoft 项目的开发环境用 MAVEN 来构建,便于 jar 版本管理以及强大的插件功能。因为 MAVEN 是一个异常强大的构建工具,能够帮我们自动化构建过程,从清理、编译、测试到生成报告,再到打包和部署。我们不需要也不应该一遍又一遍地输入命令,一次又一次地点击鼠标,我们要做的是使用 Maven 配置好项目,然后输入简单的命令(如 mvn clean install),Maven 会帮我们处理那些烦琐的任务。

Maven 是跨平台的,无论是在 Windows 上,还是在 Linux 或者 Mac 上,都可以使用同样的命令。

Java 不仅是一门编程语言，还是一个平台，通过 JRuby 和 Jython，我们可以在 Java 平台上编写和运行 Ruby 和 Python 程序。所以我们也应该认识到，Maven 不仅是构建工具，还是一个依赖管理工具和项目信息管理工具。

基于上面的这几点我们选择 MAVEN 作为开发环境构建工具。

1.4 环境搭建清单

SN	环境名称	备注说明
1	JDK1.6+	大家都懂的
2	ECLIPSE3.6	必须的!版本最好是 3.0 以上
3	M2E	Eclipse 对于 Maven 的支持插件 http://www.eclipse.org/m2e
4	MAVEN3.0.4	下载地址: http://maven.apache.org/
5	TOMCAT	这个大家都知道(在这里可用可不用)，要用的话 6.0 版本以上
6	推荐工具	使用 spring tool suit 都集成

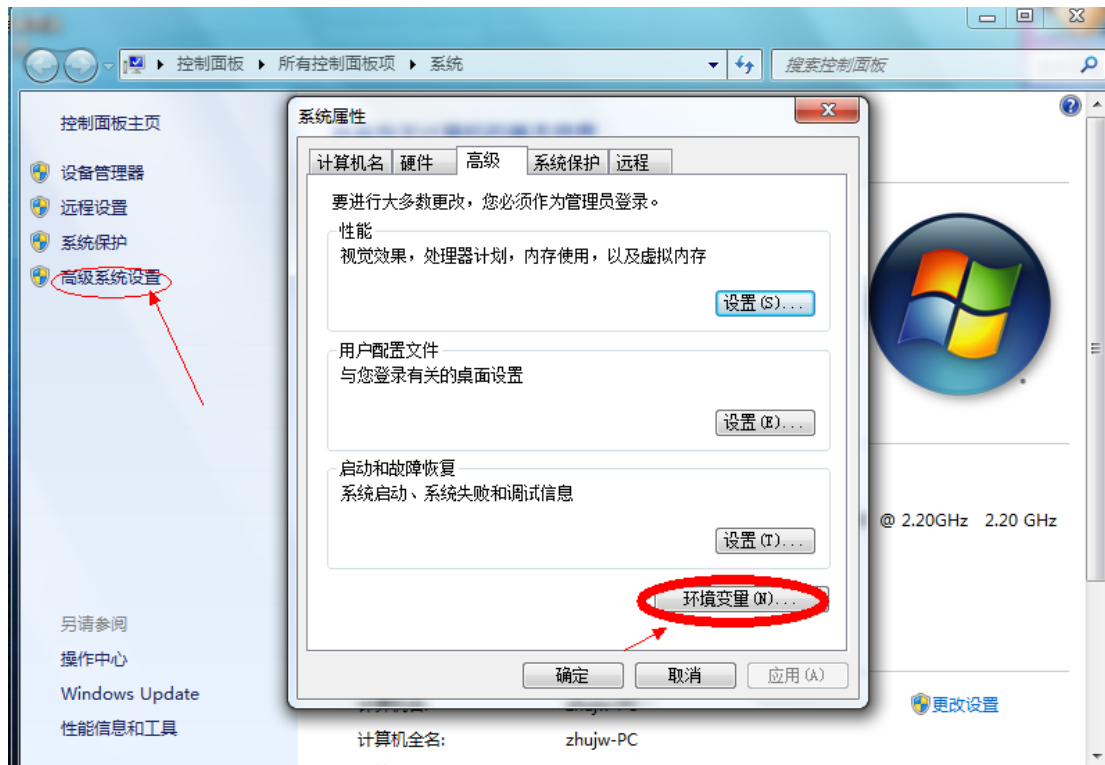
1.5 环境快速搭建

1.5.1 下载及安装 JDK

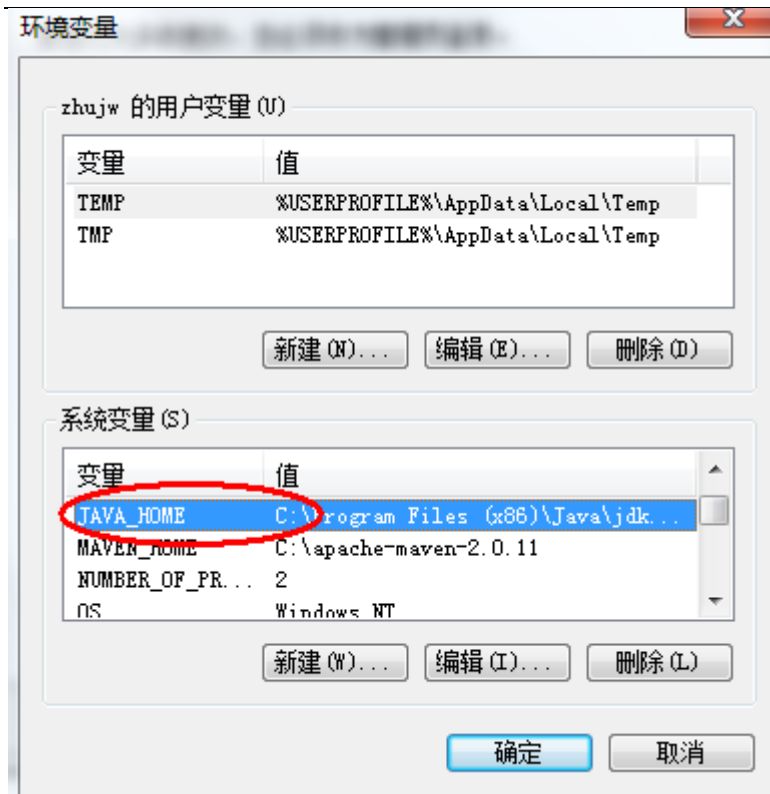
下载合适版本的 JDK，推荐选择 1.6+

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

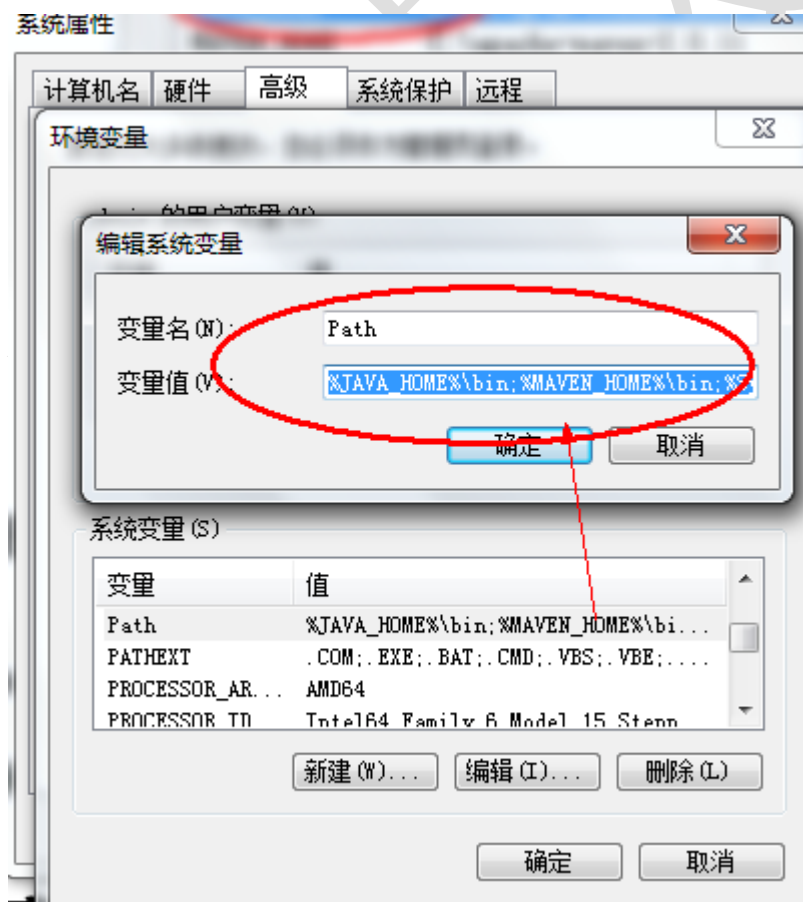
安装相对简单，下一步，步步完成即可。JDK 安装完成后，设置环境变量



新建环境变量 JAVA_HOME= C:\Program Files (x86)\Java\jdk1.6.0_23(调整为自己安装的 JDK 目录即可)

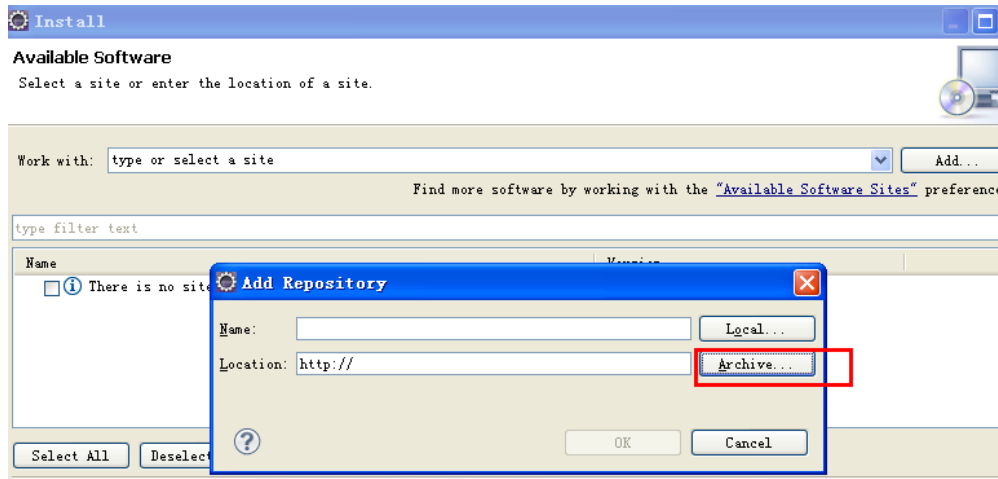


修改 path 值在最前面加入%JAVA_HOME%\bin;

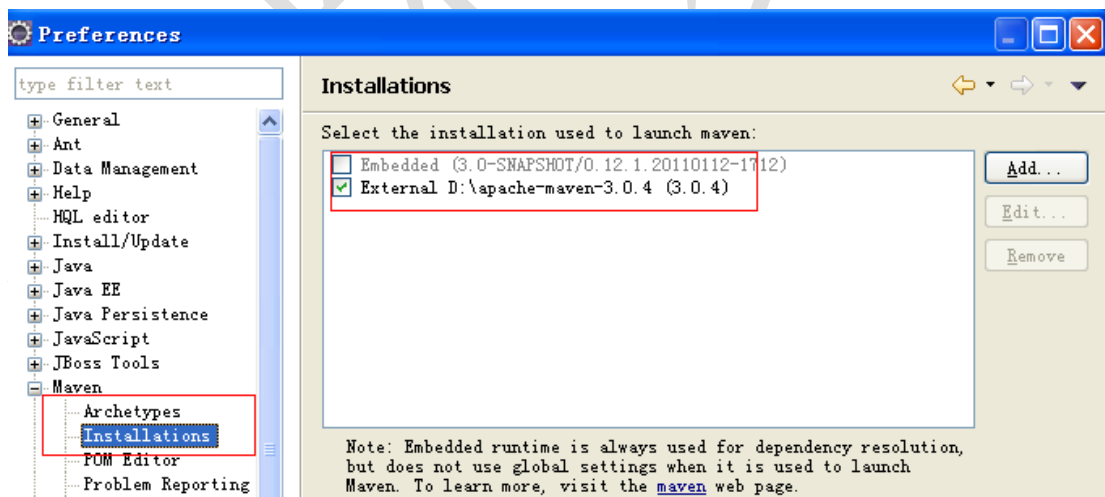


1.5.2 M2E 插件安装

打开 Eclipse , help->Install New Software..->

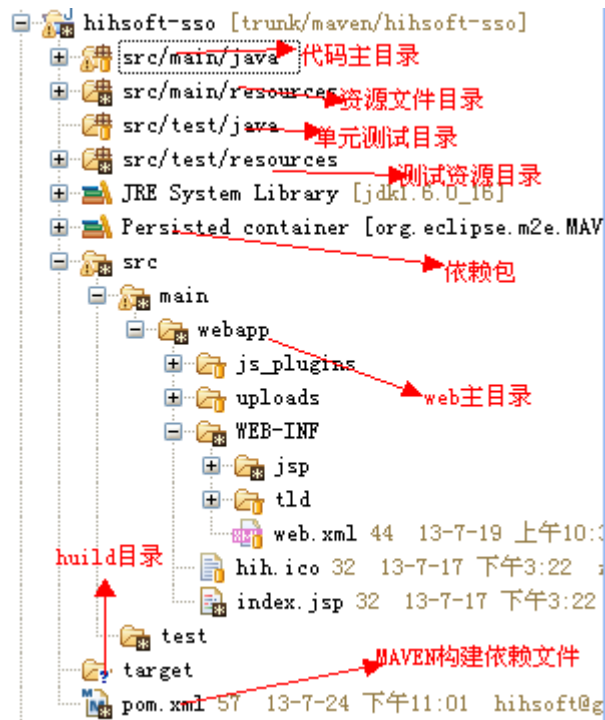


选择你刚才下载的插件，然后一直下一步就可以了。安装好插件后需配置下 MAVEN 路径，类似于配置 TOMCAT。



1.6 项目结构介绍

可以从我们指定的地址下载到源码，可以用 eclipse 导入为 maven 项目。由于之前项目不是 MAVEN 的且代码的基类都是久经测试过的，所以这里的 test 目录下并没有给出相关的单元测试。



1.7 项目编译运行

在这里我们只介绍一些基本的使用到的构建命令，如果有拉丝想知道更详细的 MAVEN 构建方面的知识可到 MAVEN 官网学习之。

项目导入 eclipse 后，目录结构应该和我上面截图的一样，这里需要注意的是你的工作目录页就是 workspace 的字符集应该是 UTF-8 的，不然会出现乱码错误，编译 JDK 版本需要 JDK1.6+，这 2 个都是可以 eclipse 里面设置的。

打开 jdbc.properties 文件

```
jdbc.url=jdbc:mysql://localhost:3306/hihsoft-sso?useUnicode=true&characterEncoding=UTF-8
```

```
jdbc.username=root
```

```
jdbc.password=root
```

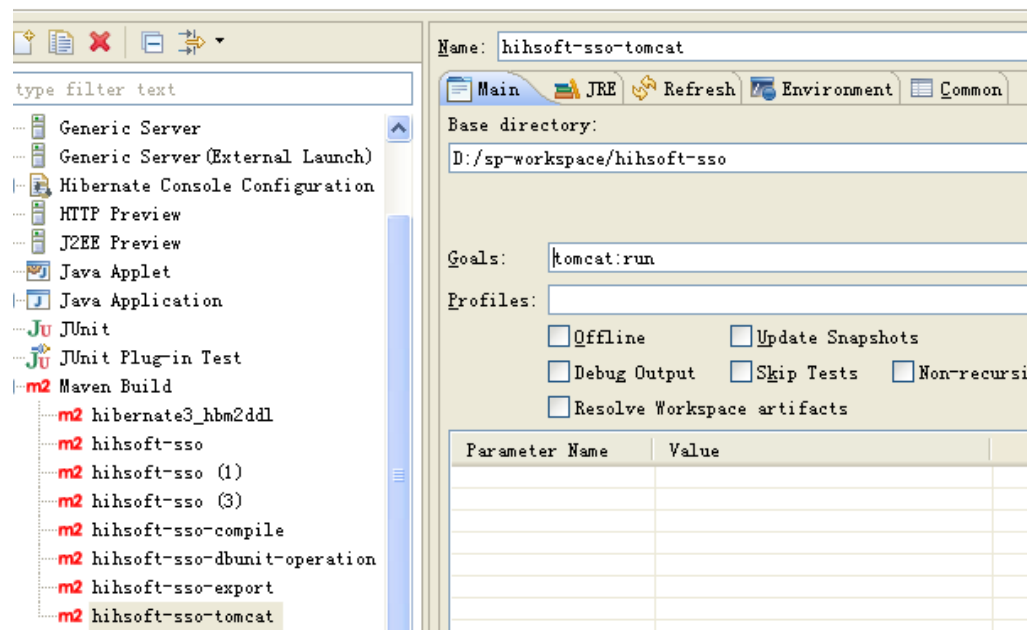
修改此处的内容改为你的数据库配置。

本地创建好数据库后，找到 hihsoft-initdb-base.xml。保证 isLoad = "true"。

这个文件是用来初始化数据库数据的。

以上都完成后，可以用 Run As -> Maven clean 来清理生成目录。

再用 maven 内置 TOMCAT 插件来运行这个项目。



Tomcat 的端口号在 POM 文件里面的

```
<plugin>
    <groupId>org.codehaus.mojo</groupId>
    <artifactId>tomcat-maven-plugin</artifactId>
    <configuration>
        <uriEncoding>UTF-8</uriEncoding>
        <path>/sso</path>
        <!--在这里定义端口号 -->
```

```
<port>9081</port>
```

```
</configuration>
```

```
</plugin>
```

这样你就可以访问了 <http://localhost:9081/sso/>。这里不端口和 path 不建议修改,修改后会导致页面转向 404。等部署后可以进入修改模块地址再改端口和 path。

注:maven 需要从 maven 的中央仓库 <http://repo1.maven.org/>下载各种依赖包,插件包,所以先请保证你的网速是可以的,如果网速原因下载不是很顺利可能会导致构建报错提示 jar 无法找到,则可以在 C:\Documents and Settings\Administrator\.m2\repository 目录下删除该 jar 的文件夹让 maven 重新下载即可。

2 Eclipse 安装和配置篇

2.1 总体介绍

Eclipse 是一种可扩展的开放源代码 IDE。2001 年 11 月,IBM 公司捐出价值 4,000 万美元的源代码组建了 Eclipse 联盟,并由该联盟负责这种工具的后续开发。集成开发环境(IDE)经常将其应用范围限定在“开发、构建和调试”的周期之中。为了帮助集成开发环境(IDE)克服目前的局限性,业界厂商合作创建了 Eclipse 平台。Eclipse 允许在同一 IDE 中集成来自不同供应商的工具,并实现了工具之间的互操作性,从而显著改变了项目工作流程,使开发者可以专注在实际的嵌入式目标上。Eclipse 框架的这种灵活性来源于其扩展点。它们是在 XML

中定义的已知接口,并充当插件的耦合点。扩展点的范围包括从用在常规表述过滤器中的简单字符串,到一个 Java 类的描述。任何 Eclipse 插件定义的扩展点都能够被其它插件使用,反之,任何 Eclipse 插件也可以遵从其它插件定义的扩展点。除了解由扩展点定义的接口外,插件不知道它们通过扩展点提供的服务将如何被使用。利用 Eclipse,我们可以将高级设计(也许是采用 UML)与低级开发工具(如应用调试器等)结合在一起。如果这些互相补充的独立工具采用 Eclipse 扩展点彼此连接,那么当我们用调试器逐一检查应用时,UML 对话框可以突出显示我们正在关注的器件。事实上,由于 Eclipse 并不了解开发语言,所以无论 Java 语言调试器、C/C++ 调试器还是汇编调试器都是有效的,并可以在相同的框架内同时瞄准不同的进程或节点。Eclipse 的最大特点是它能接受由 Java 开发者自己编写的开放源代码插件,这类似于微软公司的 Visual Studio 和 Sun 微系统公司的 NetBeans 平台。Eclipse 为工具开发商提供了更好的灵活性,使他们能更好地控制自己的软件技术。eclipse 是一款非常受欢迎的 java 开发工具,使国内的用户越来越多,实际上使用它的 java 开发人员是最多的。缺点就是较复杂,对初学者来说,理解起来比较困难。

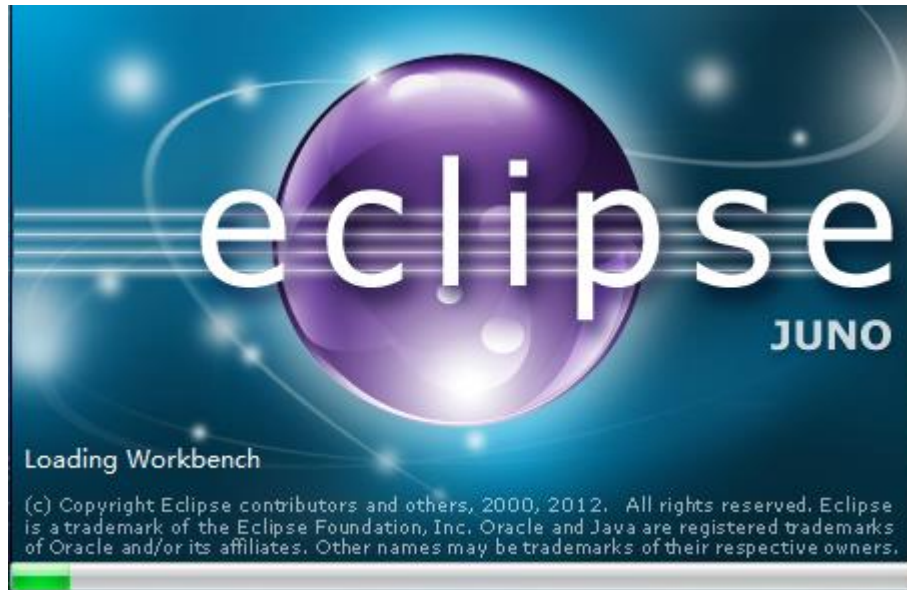
针对 eclipse 的插件目前已有上千种,eclipse 的核心就是插件扩展机制,所谓一切皆为插件。我们也可以用 eclipse 开发出自己的 eclipse 插件。

2.2 Eclipse 下载及安装

Eclipse 下载地址：<http://www.eclipse.org/downloads/>

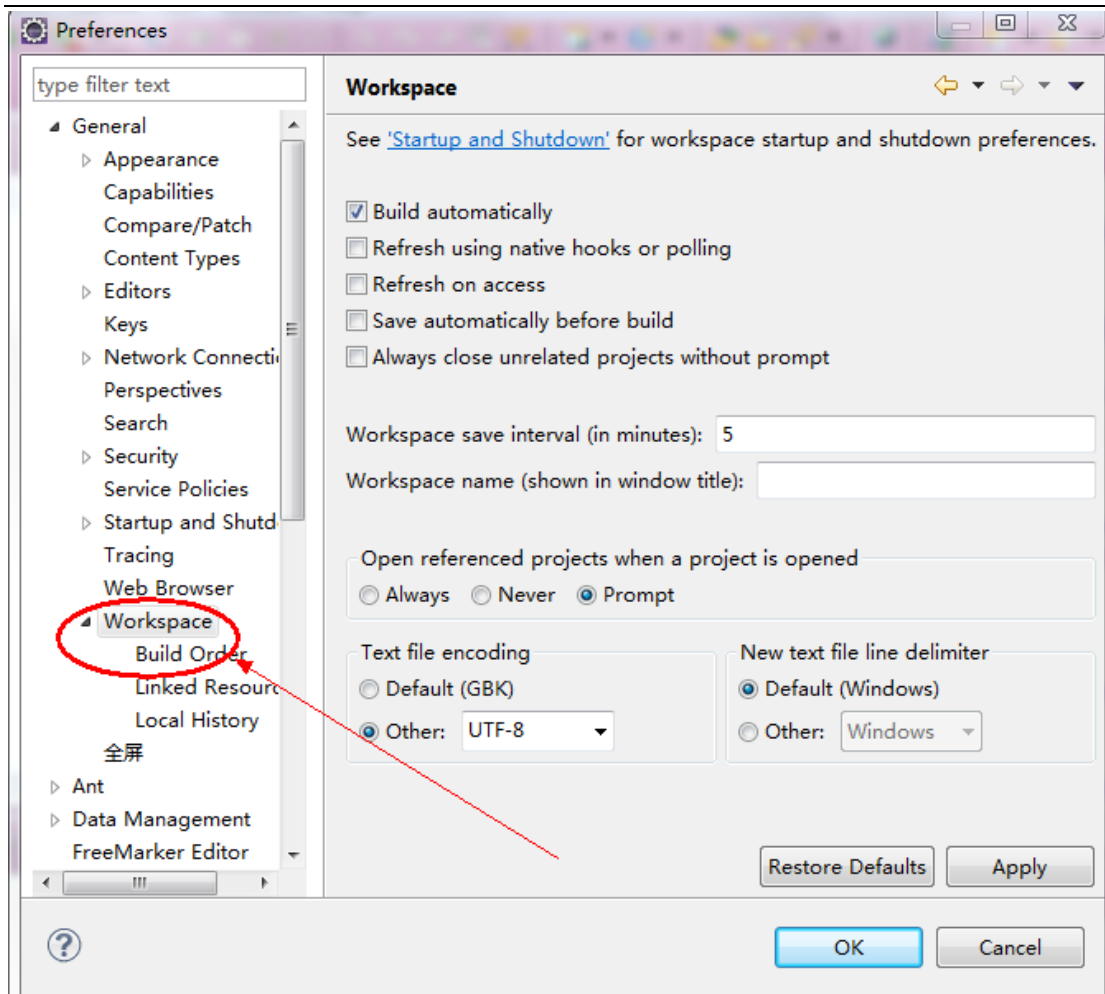
安装解压到目录下：例如 javaide

启动 eclipse , 如下图

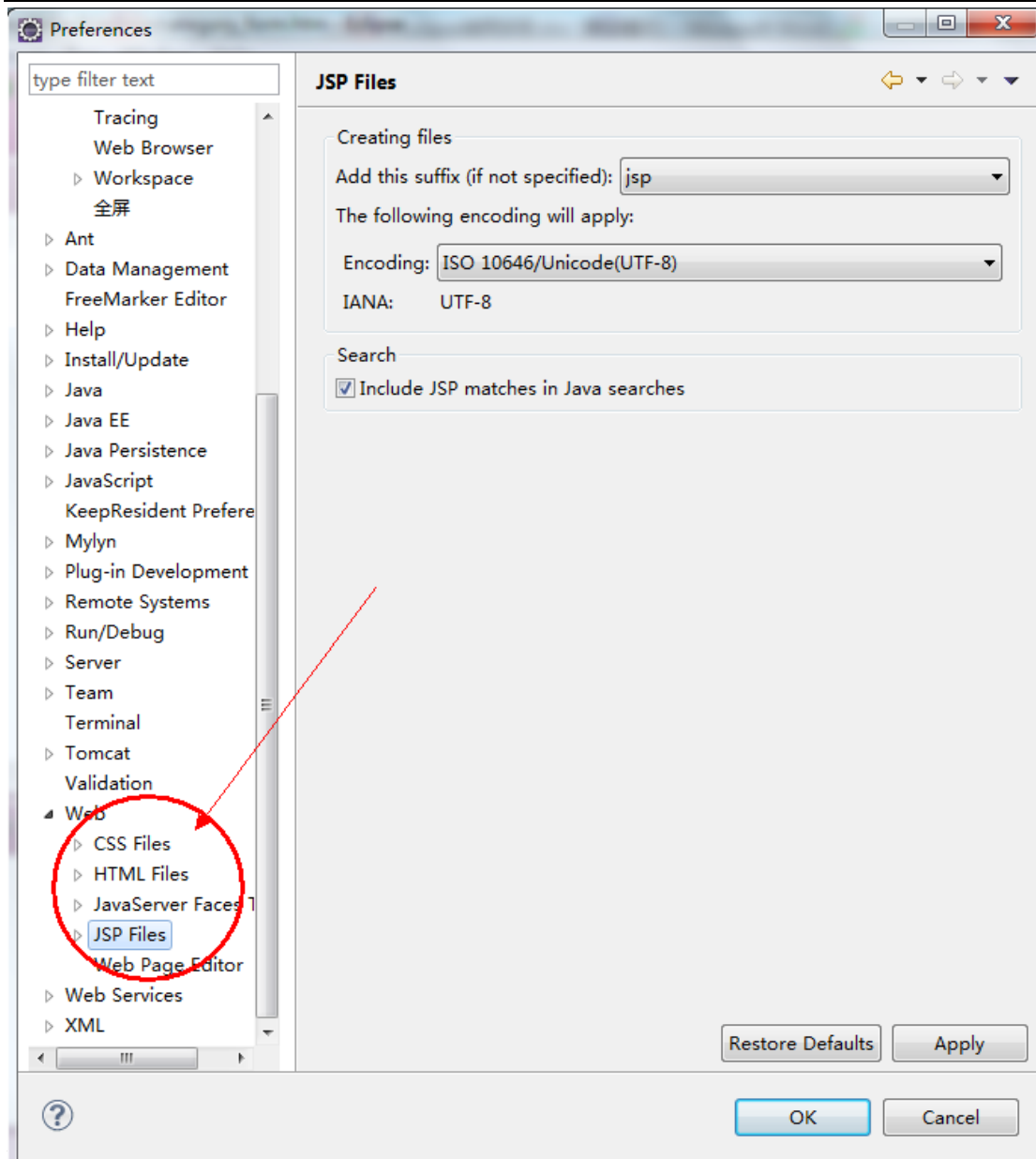


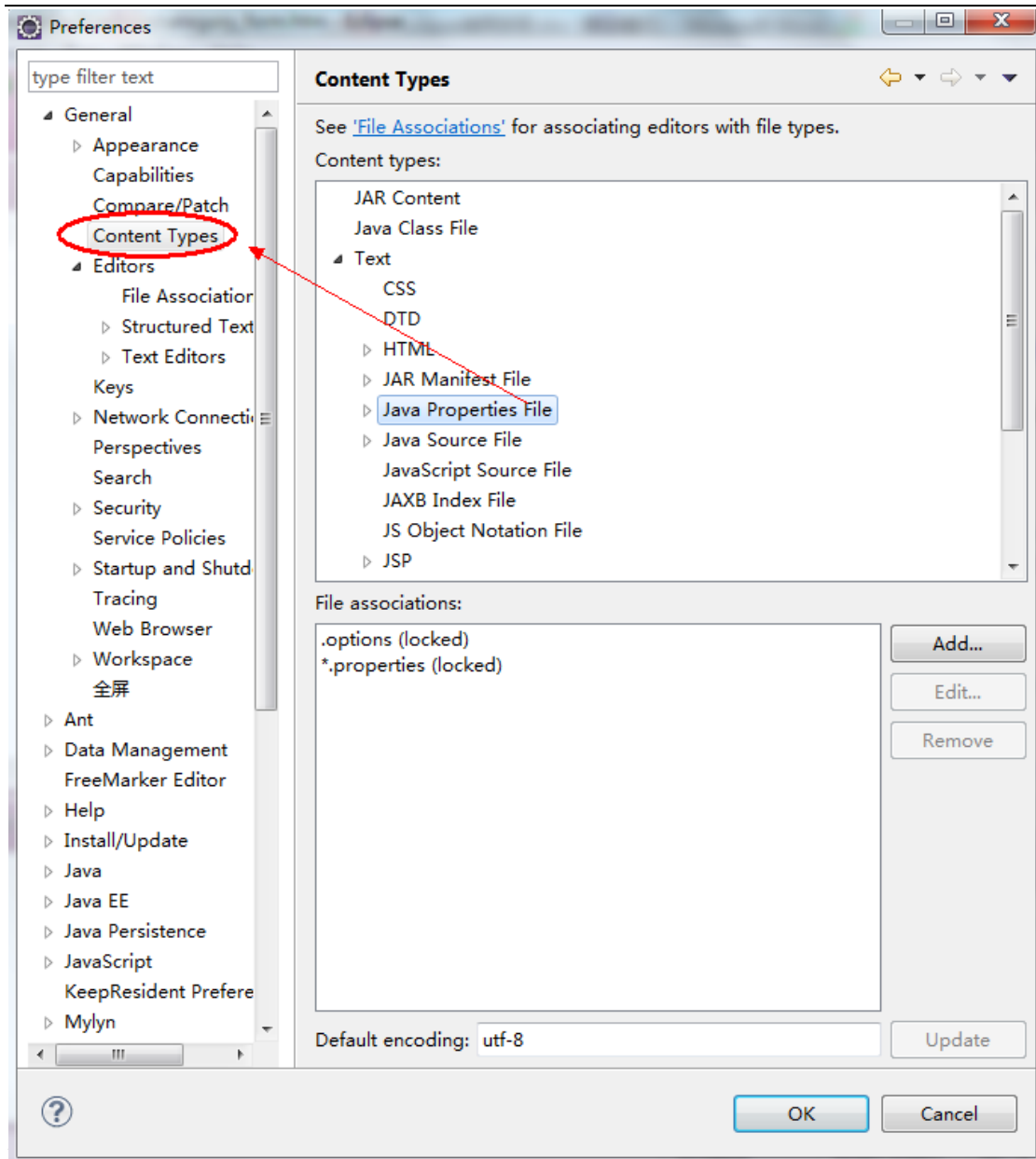
2.3 Eclipse 配置

设置工作空间的编码, 在项目中根据多年实践, 为避免中文乱码或系统迁移到其他操作系统上出现编码的情况, 推荐统一使用 UTF-8

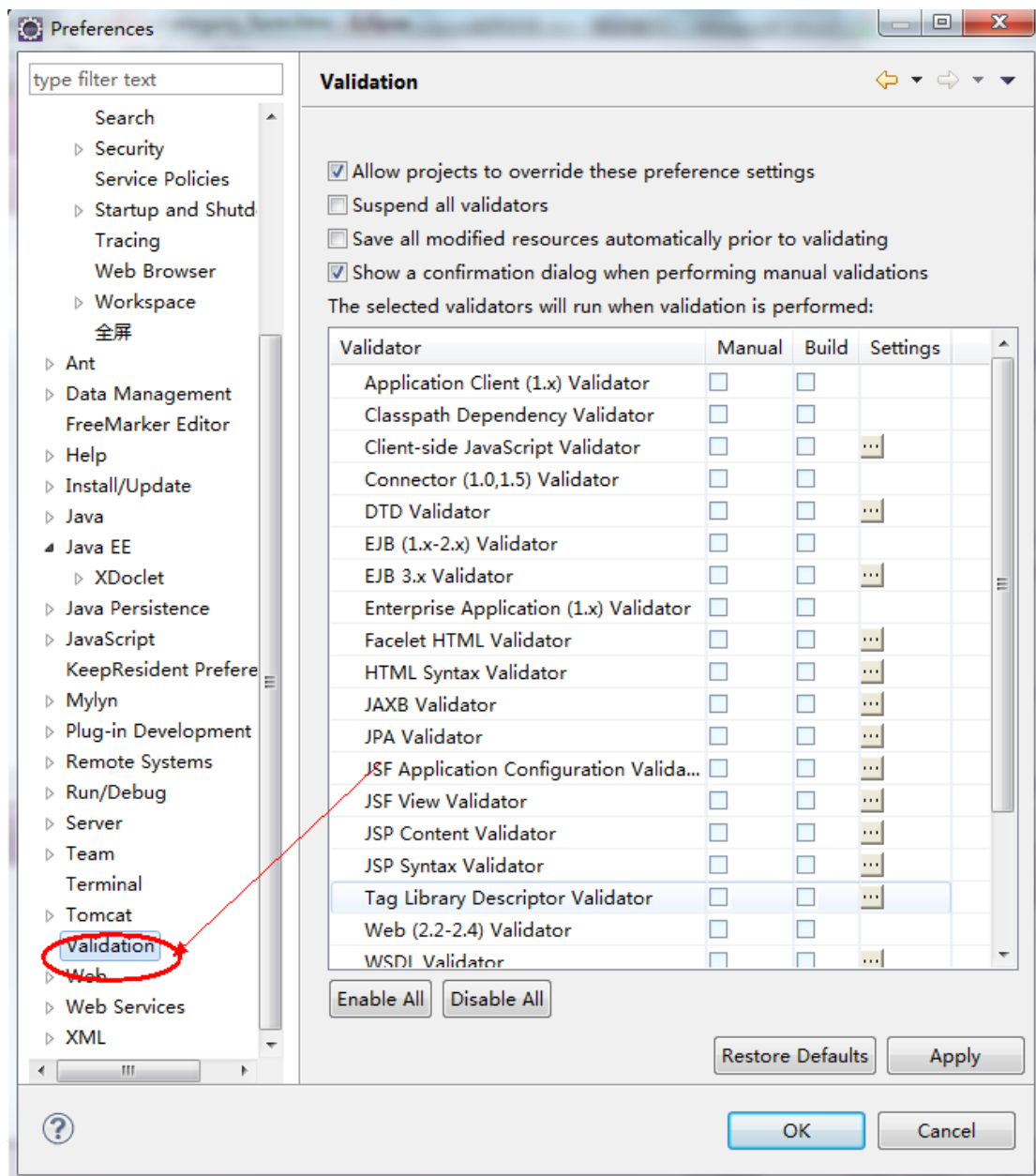


设置 JSP、javascript、CSS、属性文件 Properties、XML 等编码统一修改为 UTF-8

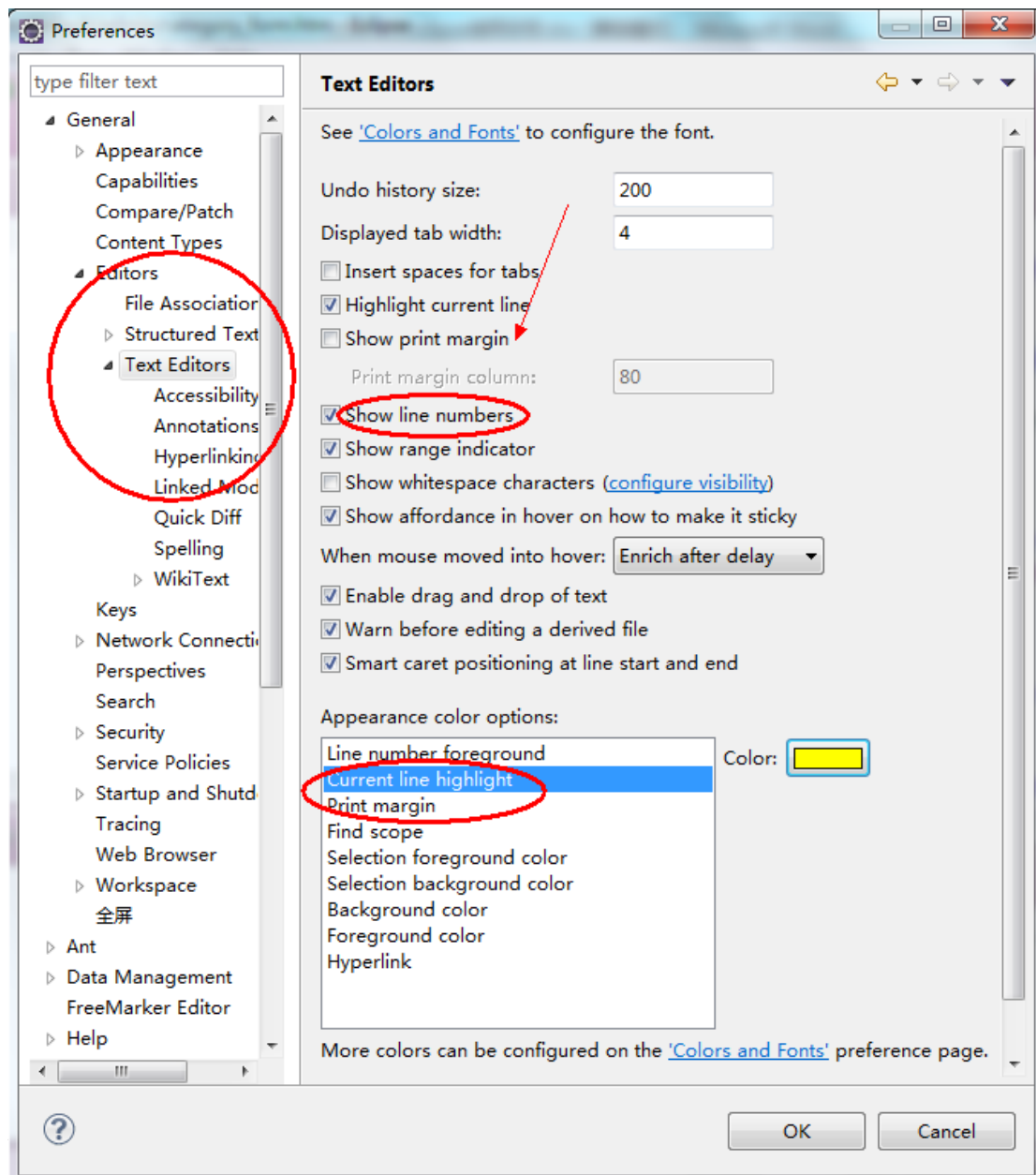




为了提高编译速度、可以去掉些文件的有效性验证



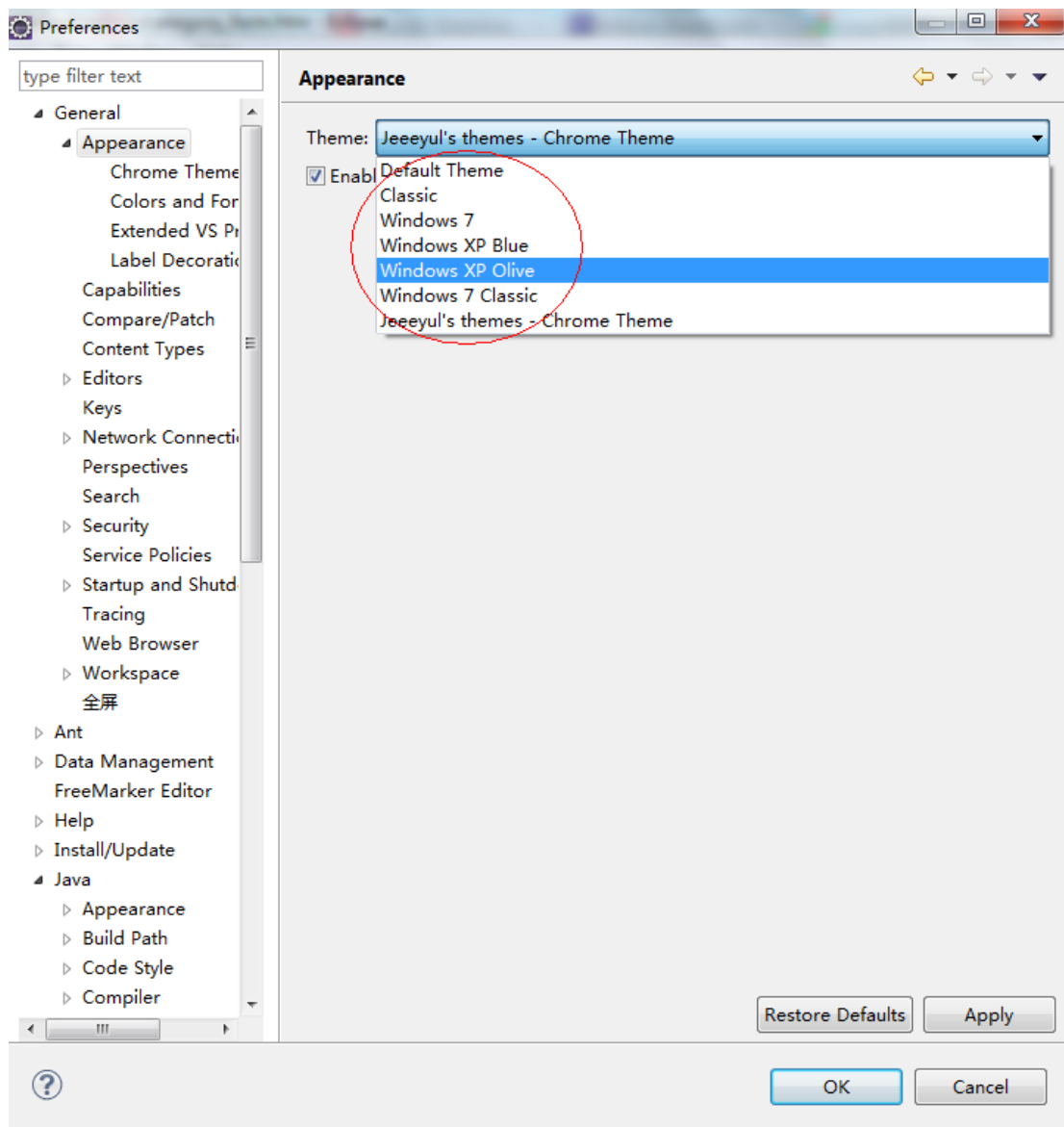
打开编辑器的行号及修改选定行的颜色

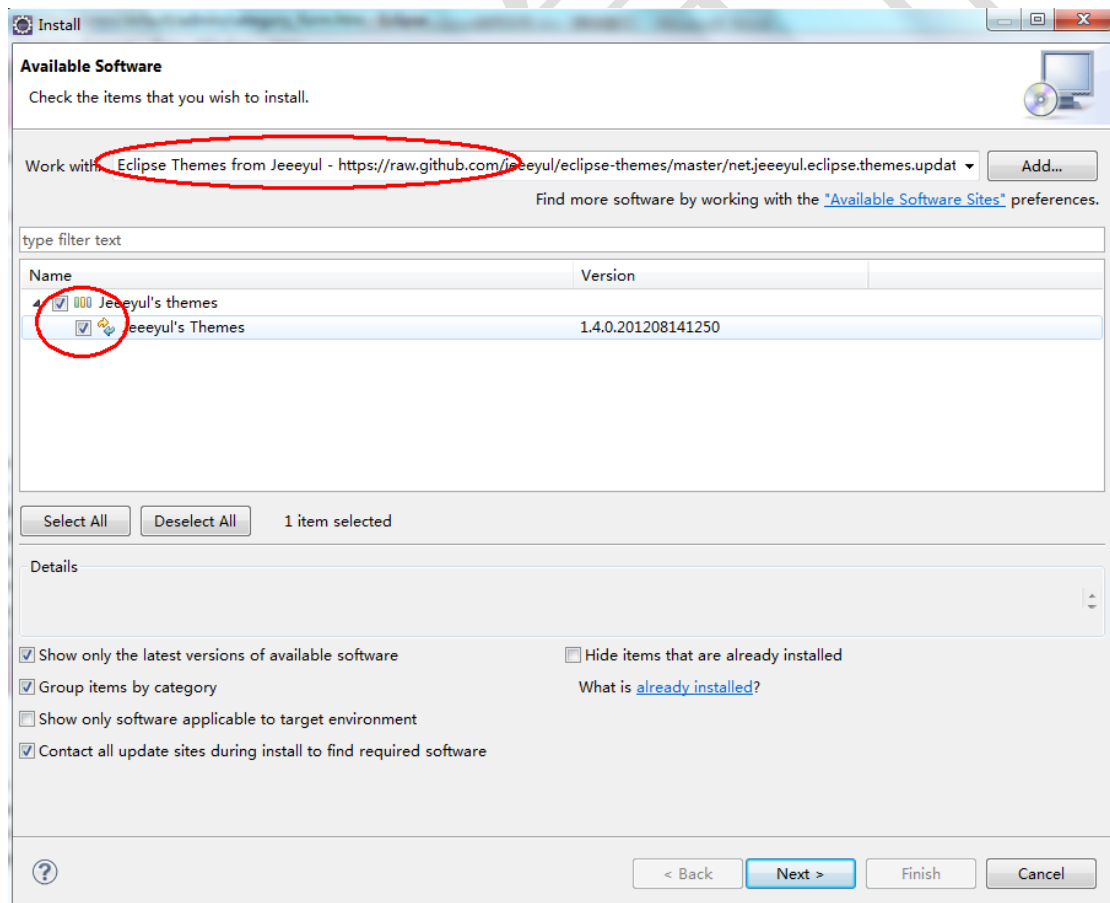
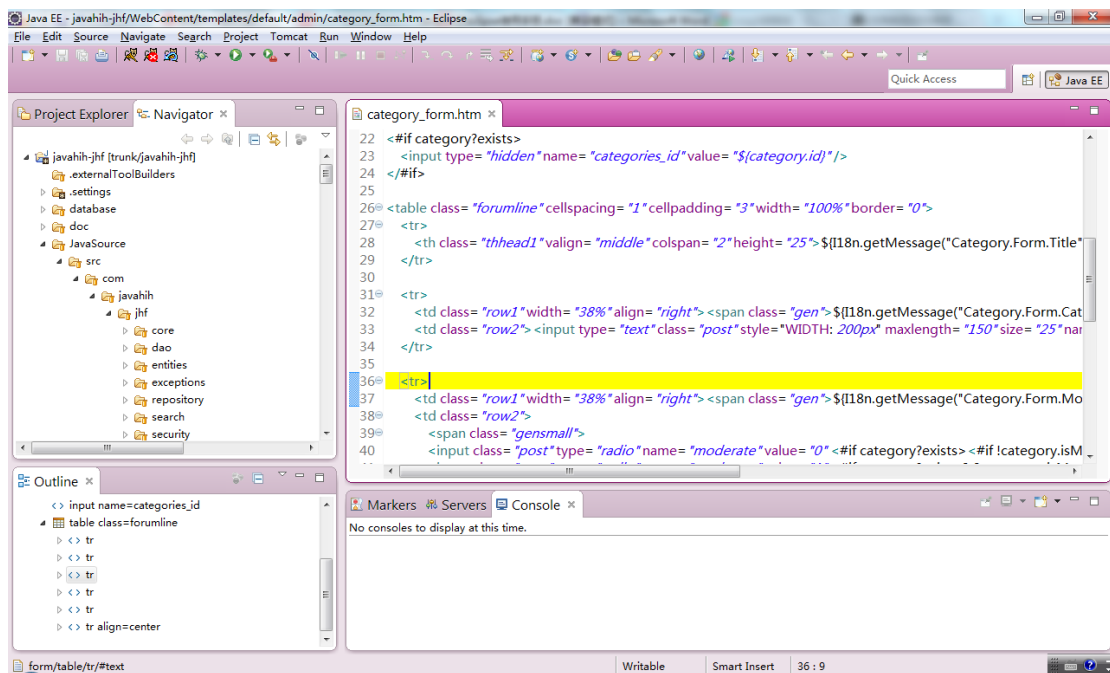


切换 Eclipse 的皮肤 themes , 以 eclipse4.2 为例

通过菜单 Help-install software 地址 :

<https://raw.githubusercontent.com/jeeeyul/eclipse-themes/master/net.jeeeyul.eclipse.themes.update.site/>





3 Eclipse 插件篇

3.1 常用插件安装

针对 eclipse 的插件目前已有上千种，eclipse 的核心就是插件扩展机制，所谓一切皆为插件。在我们日常开发过程中，下载些常用插件，提高开发效率。

3.1.1 属性编辑器

Properties Editor，并可以自动存盘为 Unicode 格式

下载地址：http://propedit.sourceforge.jp/index_en.html

3.1.2 tomcat 插件

把 tomcat 集成在 eclipse 中可以启动、停止、调试应用

<http://www.eclipse-tomcat.com/tomcatPlugin.html>

3.1.3 编辑器的列表

Editorlist：<http://editorlist.sourceforge.net/>

3.1.4 支持 freemarker 编程的插件

<http://www.freemarker.org/eclipse/update>

3.1.5 eclipse 样式(皮肤)切换的插件：eclipseskins

<http://marketplace.eclipse.org/content/eclipse-4-chrome-theme>

3.1.6 反编译插件：jadclipse

<http://sourceforge.net/projects/jadclipse/files/latest/download>

3.1.7 支持编辑器全屏插件：

<http://code.google.com/p/eclipse-fullscreen/downloads/list>

3.1.8 Eclipse 加速插件 KeepResident

<http://suif.stanford.edu/pub/keepresident/>

3.1.9 JBoss-IDE 有许多开发的插件，可以自己按需选择

<http://jboss.sourceforge.net/jbosside/updates/>

4 Eclipse 快捷键篇

4.1 常用快捷键

4.1.1 Ctrl+Space

说明:内容助理。提供对方法,变量,参数,javadoc 等的提示, 应运在多种场合,总之需要提示的时候可先按此快捷键。注:避免输入法的切换设置与此设置冲突

4.1.2 Ctrl+Shift+Space

说明:变量提示

4.1.3 Ctrl+/

说明:添加/消除//注释,在 eclipse2.0 中,消除注释为 Ctrl+\

4.1.4 Ctrl+Shift+/

说明:添加/* *//注释

4.1.5 Ctrl+Shift+\

说明:消除/* *//注释

4.1.6 Ctrl+Shift+F

说明:自动格式化代码

4.1.7 Ctrl+1

说明:批量修改源代码中的变量名,此外还可用在 catch 块上.

4.1.8 Ctrl+F6

说明:界面切换

4.1.9 Ctrl+Shift+M

说明:查找所需要得包

4.1.10 Ctrl+Shift+O

说明:自动引入所需要得包

4.1.11 Ctrl+Alt+S

说明:源代码得快捷菜单。其中的 Generate getters and setters 和 Surround with try/catch block 比较常用.建议把它们添加为快捷键.快捷键设置在 windows->preferences->Workbench->Keys

4.1.12 Ctrl+H

说明：在接口类中查找接口的实现类

4.1.13 Ctrl+D

删除行

4.1.14 Ctrl+Shift+G

在 workspace 中搜索选中元素的引用

5 Eclipse 调试篇

5.1 Eclipse 调试快捷键

F5 : Step Into (debug)

F6 : Step over (debug)

F7 : Step return (debug)

F8 : Resume (debug)

F11 : debug 上一个应用 (debug)

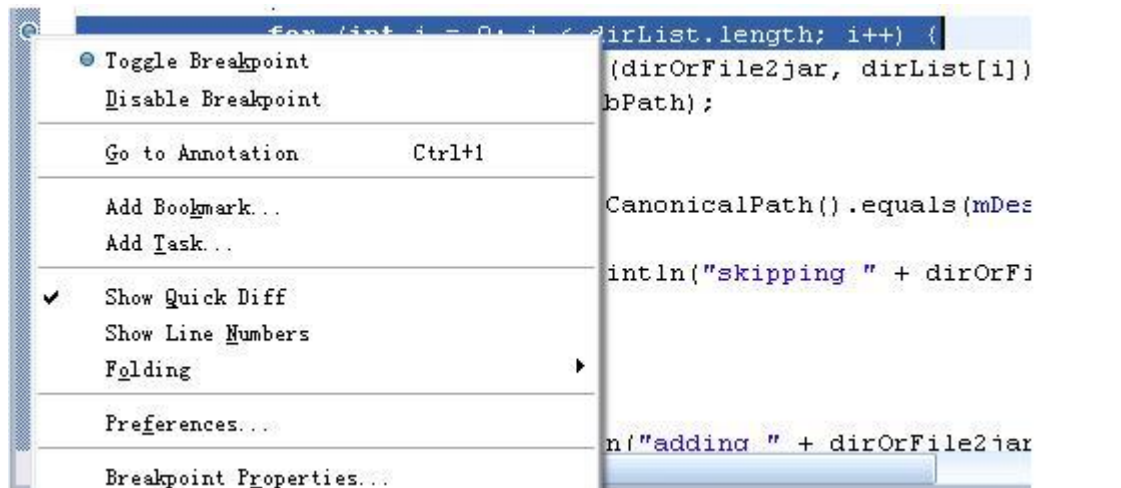
5.2 调试方法

5.2.1 条件断点

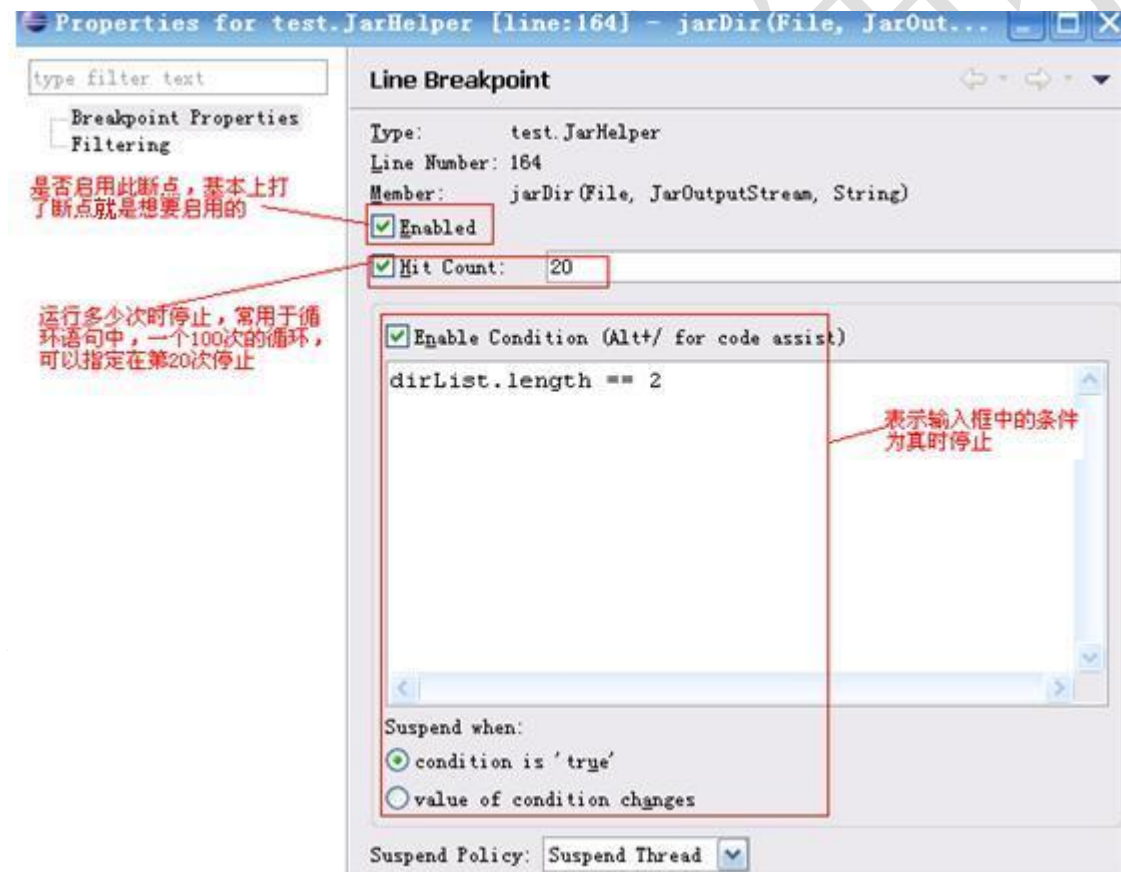
断点大家都比较熟悉，在 Eclipse Java 编辑区的行头双击就会得到一个断点，代码会运行到此处时停止。

条件断点，顾名思义就是一个有一定条件的断点，只有满足了用户设置的条件，代码才会在运行到断点处时停止。

在断点处点击鼠标右键，选择最后一个"Breakpoint Properties"



断点的属性界面及各个选项的意思如下图，



5.2.2 变量断点

断点不仅能打在语句上，变量也可以接受断点，

```
private static final int BUFFER_SIZE = 41  
  
private static String MAIN_CLASS;  
  
// =====
```

上图就是一个变量的打的断点,在变量的值初始化,或是变量值改变时可以停止,当然变量断点上也是可以加条件的,和上面的介绍的条件断点的设置是一样的。

5.2.3 方法断点

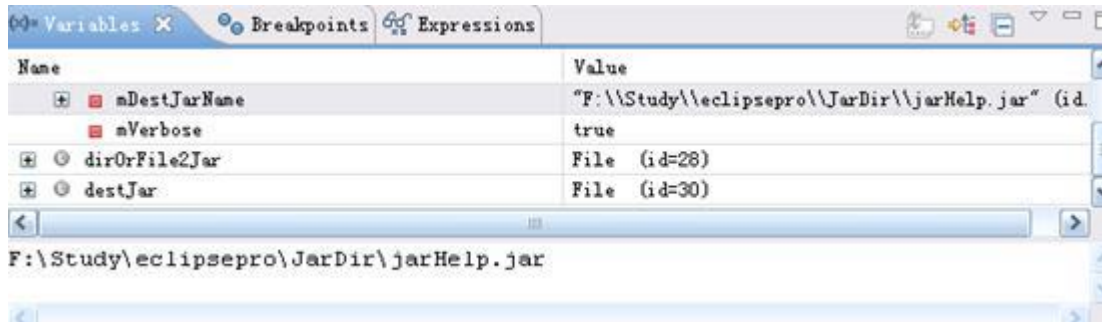
方法断点就是将断点打在方法的入口处,

```
*/  
public void jarDir(File dirOrFile2Jar, File destJar) throws IC  
  
if (dirOrFile2Jar == null || destJar == null)  
    throw new IllegalArgumentException();
```

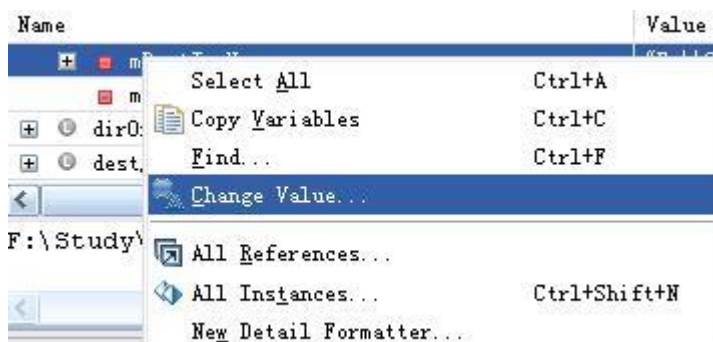
方法断点的特别之处在于它可以打在 JDK 的源码里,由于 JDK 在编译时去掉了调试信息,所以普通断点是不能打到里面的,但是方法断点却可以,可以通过这种方法查看方法的调用栈。

改变变量值

代码停在了断点处,但是传过来的值不正确,如何修改一下变量值保证代码继续走正确的流程,或是说有一个异常分支老是进不去,能不能调试时改一下条件,看一下异常分支代码是否正确?在 Debug 视图的 Variables 小窗口中,我们可以看到 mDestJarName 变量的值为 " F:\Study\eclipsepro\JarDir\jarHelp.jar "



我们可以在变量上右键，选择"Change Value..." 在弹出的对话框中修改变量的值，



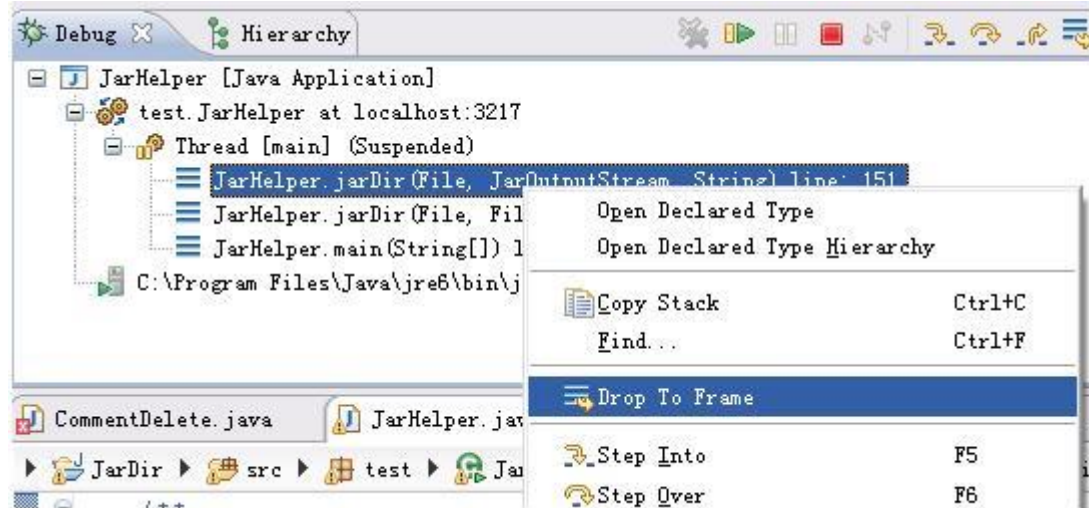
或是在下面的值查看窗口中修改，保用 Ctrl+S 保存后，变量值就会变成修改后的新值了。



5.2.4 重新调试

这种调试的回退不是万能的，只能在当前线程的栈帧中回退，也就是说最多只能退回到当前线程的调用的开始处。

回退时，请在需要回退的线程方法上点右键，选择 "Drop to Frame"



5.2.5 远程调试

用于调试不在本机上的程序，有两种方式，

本机作为客户端

本机作客户端比较常用，需要在远端的服务器上的 java 程序在启动时打开远程调试开关，

服务器端需要加上虚拟机参数 1.5 以前版本（1.5 以后也可用）：【-Xdebug -Xrunjdwp:transport=dt_socket,server=y,address=8000】

1.5 及以上版本：

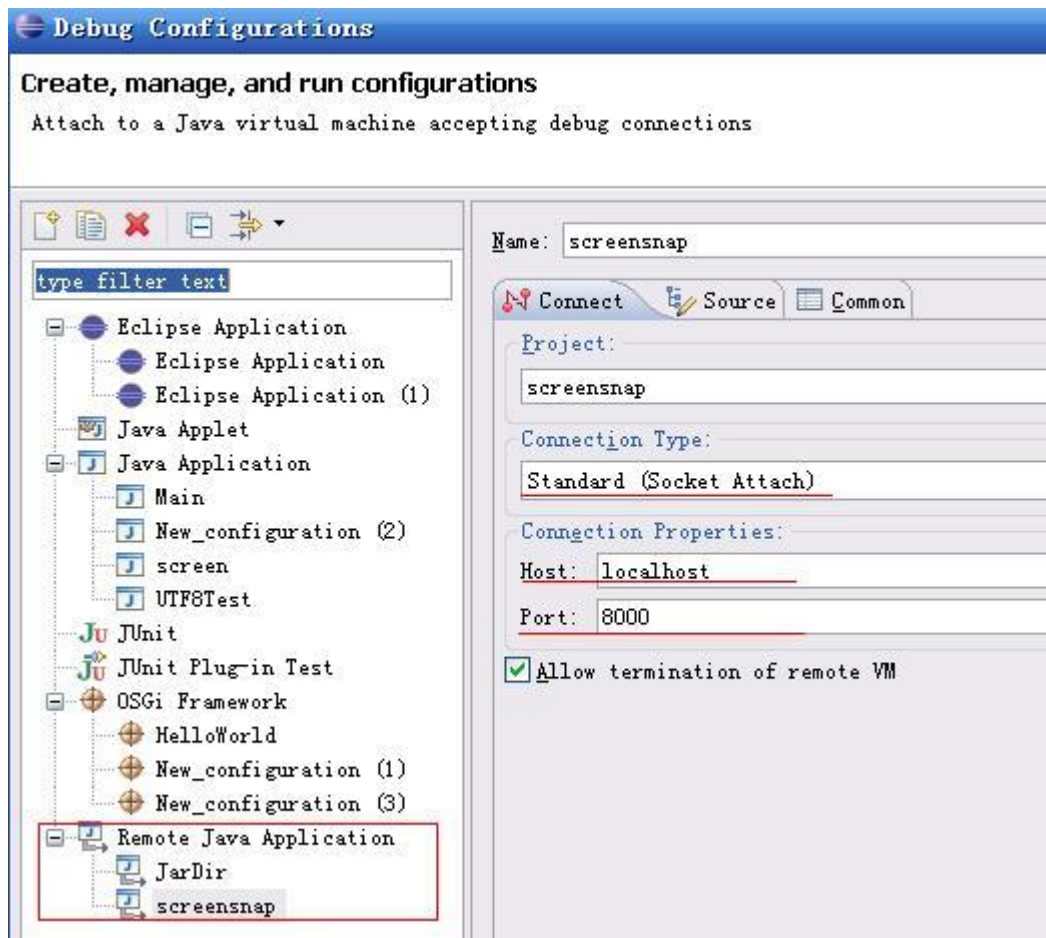
【-agentlib:jdwp=transport=dt_socket,server=y,address=8000】

F:\Study\eclipsepro\screensnap>java -Xdebug

-Xrunjdwp:transport=dt_socket,server=y,address=8000 -jar

screensnap3.jar

连接时远程服务器时，需要在 Eclipse 中新建一个远程调试程序

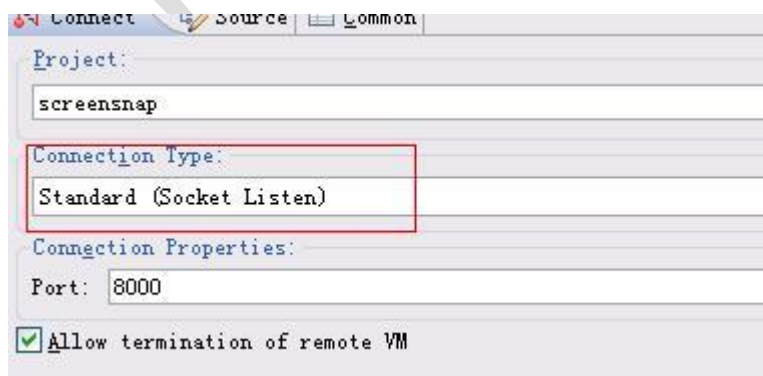


这里有一个小地方需注意，连接上的时候貌似不能自动切换到 Debug 视图，不要以为本机的调试程序没有连接到服务器端。

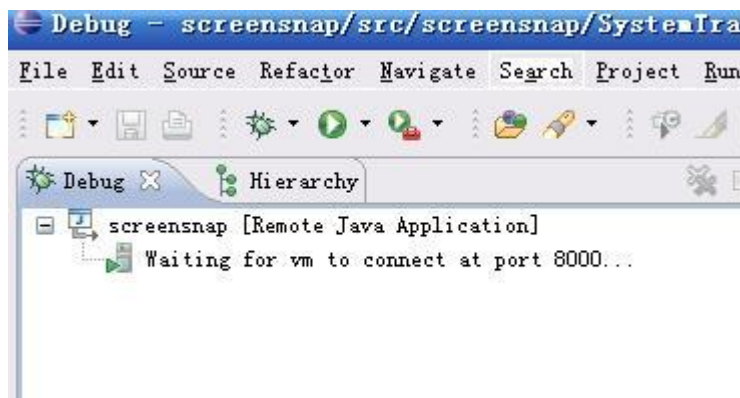
2、本机作为服务端

使用远程调试的前提是服务器端和客户端的代码是一致的。

同本机作为客户端相比，只需要修改一下“Connection Type”



这时 Eclipse 会进入到等待连接的状态



连接程序使用如下参数即可连接本机服务器，IP 地址请用实现 IP 替换~~

【 -agentlib:jdwp=transport=dt_socket,suspend=y,address=127.0.0.1:8000】

F:\Study\eclipsepro\screensnap>java

-agentlib:jdwp=transport=dt_socket,suspend=y,address=127.0.0.1:8000

-jar screensnap3.jar

远程调试时本地的代码修改可同步到远程，但不会写到远程的文件里，也就是说本地修改会在下次启动远程程序时就没有了，不会影响到下次使用时的远程代码。

5.2.6 异常断点

经常遇见一些异常，然后程序就退出来了，要找到异常发生的地方就比较难了，还好可以打一个异常断点，



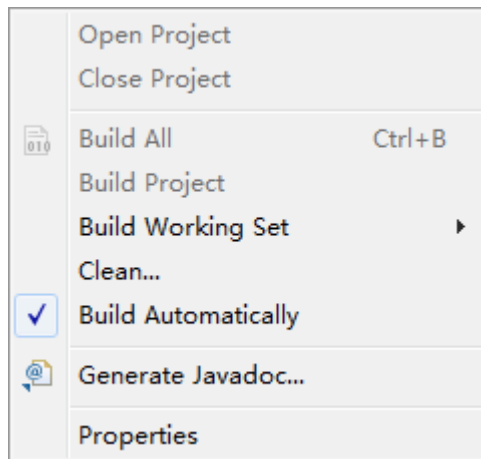
上图中我们增加了一个 NullPointerException 的异常断点，当异常发生时，代码

会停在异常发生处，定位问题时应该比较有帮助

6 Eclipse 编译打包篇

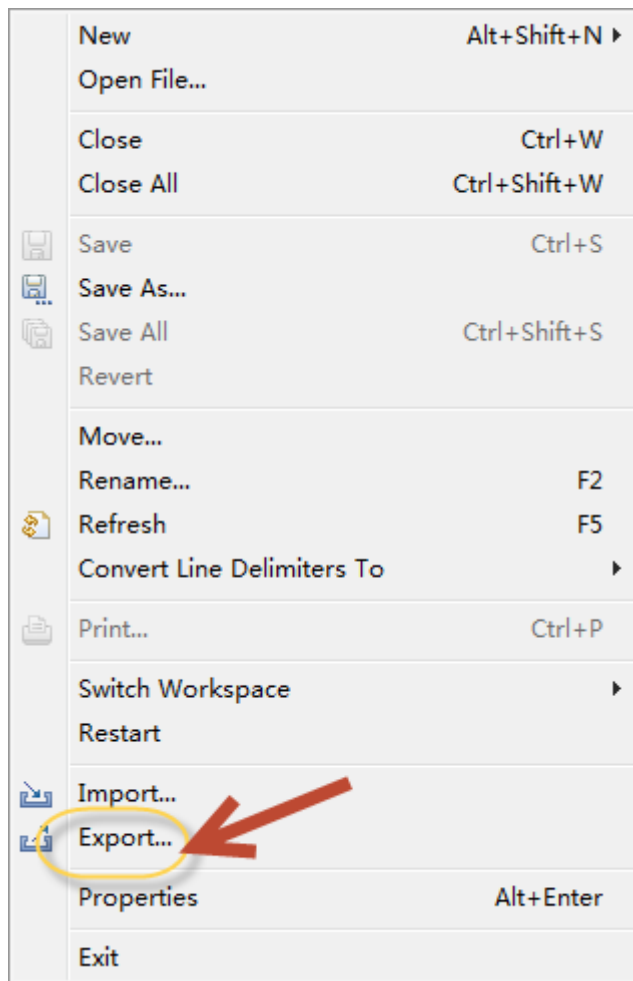
6.1 Eclipse 编译

默认 eclipse 在你执行 ctrl+shift+S 时会自动编译，不需要手工的 build，提高开发效率，当然也需要你的电脑配置好点才行。配置不高的，也可以去掉 Build Automatically，执行手工 Build Project

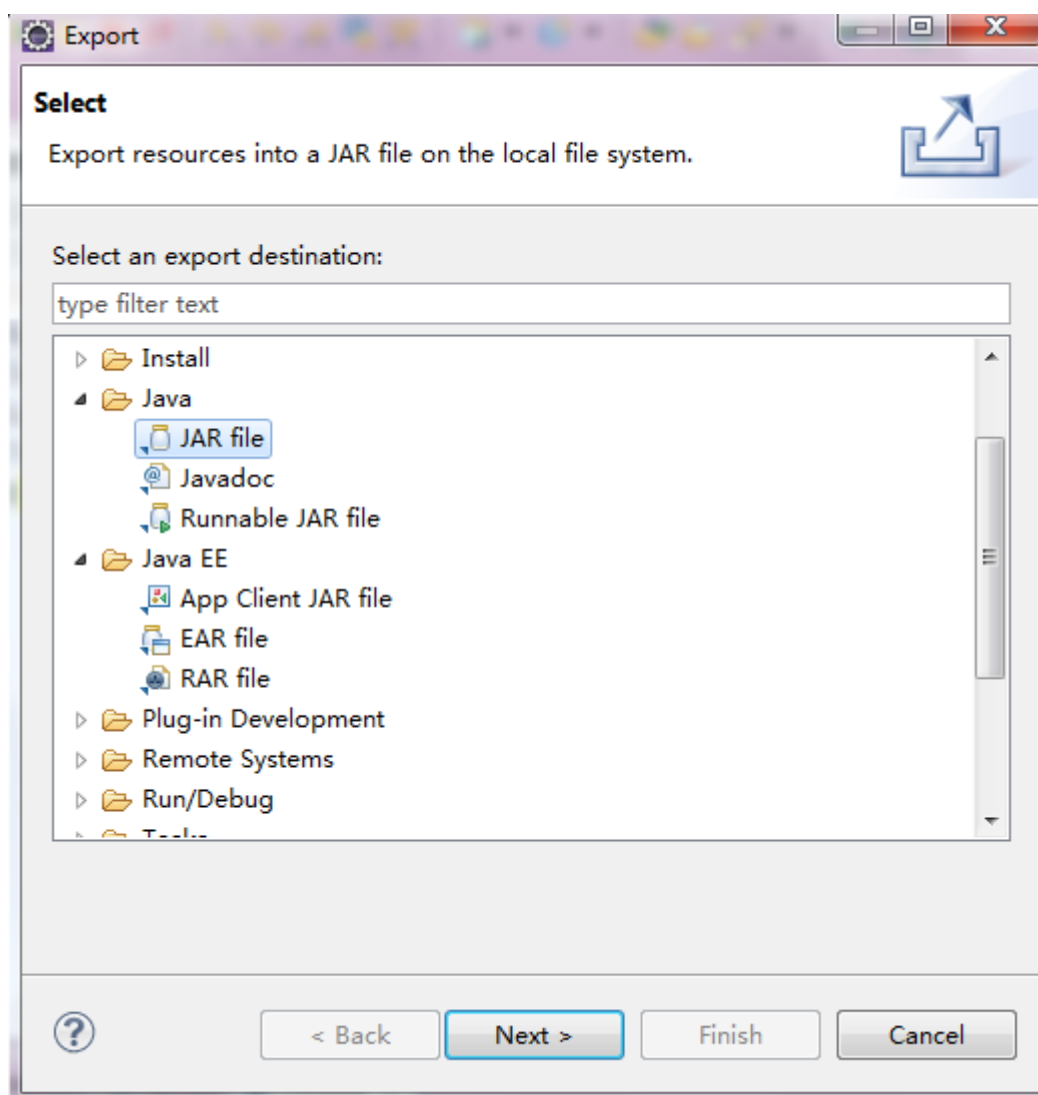


6.2 Eclipse 打包

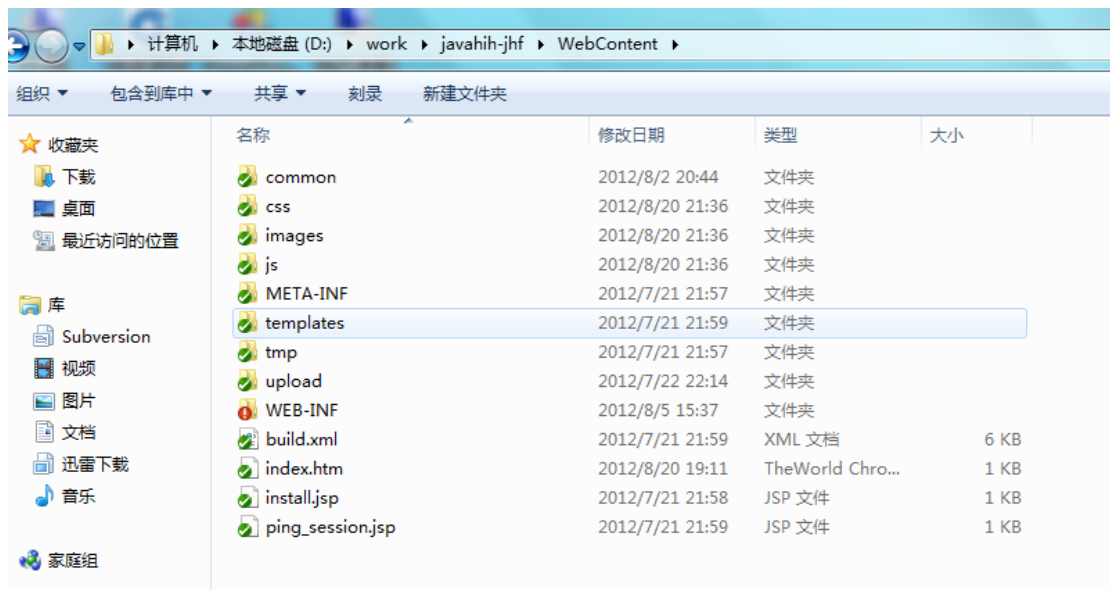
通过 Eclipse 的导出功能：Eclipse 主界面的 File 菜单下找到



根据自己的打包要求，生成相应的打包文件：例如 jar、war、ear 包文件



开始----运行输入 cmd 路径指定到 web 工程的目录下 , 执行命令:例如 : 工程目录为



```
jar -cvf javahih.war *
```

7 交流、反馈、参与贡献

QQ 群 : 80186309 , 以码会友

E-mail : hihsoft@gmail.com

Github : <https://github.com/hihsoft/sso>

googlecode : <https://code.google.com/p/hihsoft-sso/>

oschina : <https://git.oschina.net/hihsoft/sso>

如果你想参与进来共同完善它或有更好的建议 , 请联系我吧(^_^)。