

Eclipse 开发工具使用指南

2005-06-17 09:32 作者：倪大鹏出处：IBM 责任编辑：方舟

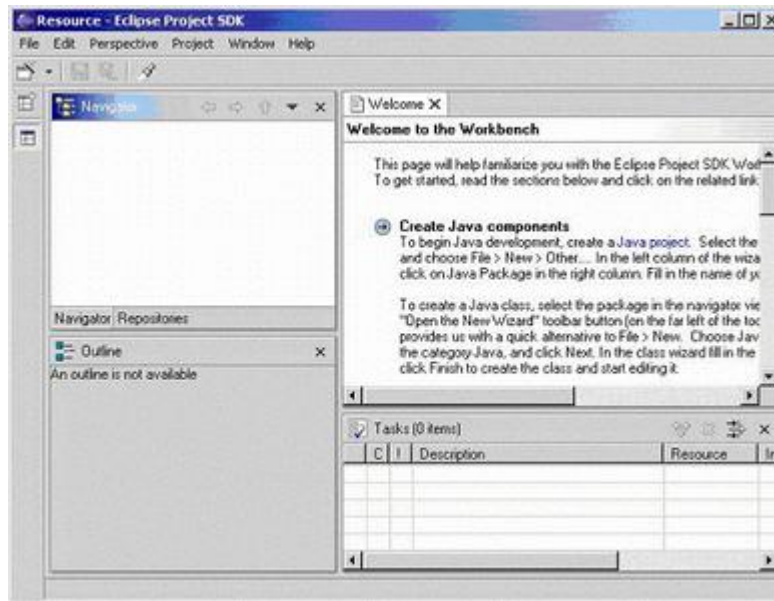
2001 年 11 月 IBM 宣布捐出了价值 4 千万美金的开发软件给开放源码的 Eclipse 项目。如此受青睐的 Eclipse 是什么样子呢，如何使用呢？本文的第一部分会使你对 Eclipse 有一个初步的认识。并且，Eclipse 这个号称未来能进行任何语言开发的 IDE 集大成者，又将如何实现它的这一宏伟目标？第二部分介绍的 Eclipse 的插件开发将是其中的关键所在。

Eclipse 是替代 IBM Visual Age for Java (以下简称 IVJ) 的下一代 IDE 开发环境，但它未来的目标不仅仅是成为专门开发 Java 程序的 IDE 环境，根据 Eclipse 的体系结构，通过开发插件，它能扩展到任何语言的开发，甚至能成为图片绘制的工具。目前，Eclipse 已经开始提供 C 语言开发的功能插件。更难能可贵的是，Eclipse 是一个开放源代码的项目，任何人都可以下载 Eclipse 的源代码，并且在此基础上开发自己的功能插件。也就是说未来只要有人需要，就会有建立在 Eclipse 之上的 COBOL, Perl, Python 等语言的开发插件出现。同时可以通过开发新的插件扩展现有插件的功能，比如在现有的 Java 开发环境中加入 Tomcat 服务器插件。可以无限扩展，而且有着统一的外观，操作和系统资源管理，这也正是 Eclipse 的潜力所在。

虽然目前 Eclipse 项目还没有最后完成，但从已有的版本中已经能领略到 Eclipse 设计主导思想和主要功能特点。现在就了解 Eclipse 不但能使广大程序员对这款业界期望很高的 IDE 能一睹为快，更为重要的是如果能参加到 Eclipse 项目的开发中或是阅读它的开放源代码，这对广大程序员来说无疑是一个千载难逢的提高编程水平的好机会。Eclipse 计划提供多个平台的版本，象 Windows, Linux, Solaris, HP-UX 和 AIX，以下只介绍 Windows 版本。本文第一部分先介绍 Eclipse 的基本使用方法。第二部分介绍如何进行 Eclipse 的插件开发。

一 . Eclipse 简介

Eclipse 是开放源代码的项目，你可以到 www.eclipse.org 去免费下载 Eclipse 的最新版本，一般 Eclipse 提供几个下载版本：Release, Stable Build, Integration Build 和 Nightly Build，建议下载 Release 或 Stable 版本，笔者用的是 Build20020125 (Stable 版本)。Eclipse 本身是用 Java 语言编写，但下载的压缩包中并不包含 Java 运行环境，需要用户自己另行安装 JRE，并且要在操作系统的环境变量中指明 JRE 中 bin 的路径。安装 Eclipse 的步骤非常简单：只需将下载的压缩包按原路径直接解压既可。需注意如果有了更新的版本，要先删除老的版本重新安装，不能直接解压到原来的路径覆盖老版本。在解压缩之后可以到相应的安装路径去找 Eclipse.exe 运行。如果下载的是 Release 或 Stable 版本，并且 JRE 环境安装正确无误，一般来说不会有什么问题，在闪现一个很酷的月蚀图片后，Eclipse 会显示它的缺省界面：



图一

乍一看起来，Eclipse 的界面有点象 JBuilder，但实际操作进程中会发现它更象 IVJ，毕竟开发 Eclipse 的主导用量是开发 IVJ 的原班人马（可参考 www.oti.com）。另外还值得一提的是 Eclipse 项目的参预者除了 IBM 以外，还有象 Borland，Rational Software，RedHat，Merant 等一大批业界的佼佼者，这也为 Eclipse 的未来奠定了良好的基础。

下面将分别对 Eclipse 的各种特性作简单介绍，包括：文件存放，开发环境，编译与运行，版本管理，使用插件。

1. 文件存放

大多 IVJ 的初学者都对找不到 Java 源代码感到非常不适应，因为 IVJ 把所有的源代码都存储到一个 repository 库文件中，想要得到文本格式的源代码必须用 Export 功能从 repository 中导出源代码。使用了 repository 的 IVJ 对源代码的管理功能几乎达到极致，正是这一点使得许多程序员对 IVJ 钟爱有加。而 Eclipse 将源代码以文本方式保存，却实现了 IVJ 对源代码管理的几乎全部功能，并且还增加了一些新的功能，能达到这一点，不能不惊叹于 Eclipse 开发者的高超技巧。

安装 Eclipse 之后，在安装路径的下一层路径中会有一个 workspace 文件夹。每当在 Eclipse 中新生成一个项目，缺省情况下都会在 workspace 中产生和项目同名的文件夹以存放该项目所用到的全部文件。你可以用 Windows 资源管理器直接访问或维护这些文件。

将已有的文件加入到一个项目中目前有三种方式：第一种是象在 IVJ 中的一样，用 IDE 的 "File" 菜单中的 "Import" 功能将文件导入到项目中。这也是推荐的方式。第二种是从 Windows 的资源管理器中直接拖动文件到项目中。第三种就是直接将文件拷贝到项目文件夹中，然后在 Eclipse 的资源浏览窗口中选择项目或文件夹并执行从本地刷新功能（Refresh from locate）。需要说明的一点是，项目文件夹可以放在计算机的任何位置，并且可以在 Eclipse 中用新建项目的方法将项目路径指定到已经存在的项目文件夹，然后在 Eclipse 中刷新即可。但要注意的是，任何项目文件夹建立或指定，目前都只能在 Eclipse 中用新建项目

的方法来实现，即使是在缺省存储项目文件夹的 workspace 路径下新建一个文件夹，在 Eclipse 环境中也是无法将它变成一个项目，也就是说，这个文件夹对 Eclipse 是不可视的。

2. Eclipse 开发环境

和 IVJ 一样，Eclipse 开发环境被称为 Workbench，它主要由三个部分组成：视图（Perspective），编辑窗口（Editor）和观察窗口（View）。在下面的介绍中，希望读者能知道 Java 视图，Java 包浏览窗口，资源视图，资源浏览窗口等区别（其实最主要的也就是要区别视图和窗口），以免在进一步的阅读中产生混淆。图二是它们之间的关系结构略图：



图二

在图二中，可以看出 Workbench 包含多个视图，而每个视图又包含不同的窗口。由于每个编辑窗口有很大的共性，而且缺省情况它们都在同一区域中显示，因此我们只在每个视图中标出一个编辑窗口，以绿色表示。观察窗口则各不相同，这里以红色表示。

下面首先介绍编辑窗口。所有文件的显示和编辑都包含在编辑窗口里。缺省情况下打开的多个文件是以标签（TagTable）方式在同一个窗口中排列，可以用拖动方式将这些文件排列成各种布局。方法是拖动某一个文件的标签（tag）到编辑窗口的边框，当光标有相应的变化时再释放。

当文件被加入到项目中后，在资源浏览或 Java 包浏览窗口双击文件，Eclipse 会试图打开这个文件：其中 Eclipse 内嵌的编辑器能缺省打开一些文件，如 *.java，*.txt，*.class 等等。如果是其它类型的文件，Eclipse 会调用操作系统相应的缺省编辑器打开，如 word 文档，PDF 文件等。同时 Eclipse 也可以象 IVJ 一样用指定的编辑器打开相应的文件。例如在 Eclipse 项目中双击 HTML 文件时，可能希望是用 Notepad 打开，而不是用系统缺省的 IE 浏览器打开。实现的方法是打开菜单栏中的 WorkBenchàPreferences 对话框，之后在对话框中选择 WorkBenchàFile Editors，然后添加文件类型，如 *.html，再为其指定编辑器即可。

在编辑窗口，还值得注意的是，习惯了 IVJ 的程序员在编辑 Java 程序的时候，更愿意以方法为独立的

编辑单位（即在编辑窗口中只显示单个的方法，而不是程序全部的源代码），这种方式也的确是非常合理的开发方式，不仅代码的显示更加简捷，还能辅助程序员编出封装性更好的类。在 Eclipse 在工具栏上提供了一个切换按钮，可以在"显示全部代码"和"只显示所选单元"（这里的单元指的是单个方法、变量、导入的包等）之间切换（可参考下面的图三）。建议没

有使用过 IVJ 的程序员也尝试一下在"只显示所选单元"状态下进行代码开发。

其次要介绍的是观察窗口，它配合编辑窗口并提供了多种的相关信息和浏览方式。常用的观察窗口有资源浏览窗口（Navigator），Java 包浏览窗口(Packages)，控制台（Console），任务栏（Task）等等。

浏览窗口和 Java 浏览窗口是观察窗口核心部分。前者和 Windows 的浏览器差不多，能浏览项目文件夹中的所有文件，后者用来浏览项目中的 Java 包，包中的类，类中的变量和方法等信息。在 Java 浏览窗口中可以通过用鼠标右键的菜单中的 Open Type Hierarchy 打开层次浏览窗口（Hierarchy），这个窗口非常实用，它能非常清晰的查看类的层次结构。类中的编译出错信息可以在任务窗口中查到，同时它也可以成为名符其实的任务窗口：向其中添加新的任务描述信息，来跟踪项目的进度。控制台则主要用来显示程序的输出信息。在调试程序的时候，会有更丰富的观察窗口来帮助程序员进行调试，如变量值察看窗口，断点窗口等等。

观察窗口是任何 IDE 开发环境的核心，用好观察窗口是也就是用好 IDE 开发环境。Eclipse 提供了丰富的观察窗口，能真正用好这些窗口恐怕要得经过一段时间的磨练。

最后介绍视图。一个视图包括一个或多个编辑窗口和观察窗口。在开发环境的最左侧的快捷栏中的上部分显示的就是当前所打开的视图图标。视图是 Eclipse 的最灵活的部分，可以自定义每个视图中包含的观察窗口种类，也可以自定义一个新视图。这些功能都被包括在"Perspective" 菜单中。在 Eclipse 的 Java 开发环境中提供了几种缺省视图，如资源视图（Resource Perspective，它也是第一次启动 Eclipse 时的缺省视图），Java 视图（Java Perspective），调试视图（Debug Perspective），团队视图(Team Perspective) 等等。每一种视图都对应不同种类的观察窗口。可以从菜单栏中的 PerspectiveàShow View 看到该视图对应的观察窗口。当然，每个视图的观察窗口都是可配置的，可以在菜单栏中的 PerspectiveàCustomize 进行配置。多样化的视图不但可以帮助程序员以不同角度观察代码，也可以满足不同的编程习惯。

3. 编译与运行

在 IVJ 中调试功能非常强大，多种跟踪方式，断点设置，变量值察看窗口等等。这些在 Eclipse 中都有提供。在本文下面介绍插件的时候，会结合例子更加详细的介绍如何使用配置项目环境，如何运行和调试程序。

在 Java 视图中，工具栏中有两个按钮，分别用来进行调试和运行。并且可能由于安装的插件不同 Eclipse 会存在多种运行/调试程序的方式，为了确定当前项目用那一种方式运行，需要在项目的属性选项中的设置 LauncheràRun/Debug 选项。通常我们需要用的是"Java Application"方式。在这种方式下，如果当前位置是包含 main()方法的 Java 程序，点击调试/运行按钮就会立即开始执行调试/运行功能。如果当前位置是在包或项目上，Eclipse 会搜索出当前位置所包含的所有可执行程序，然后由程序员自己选择运行那一个。

在目前的 Eclipse 的 Release 和 Stable 版本中缺省安装了插件开发环境（Plug-in Development

Environment，即 PDE，它本身也是一个插件），此时系统除了"Java Application" 运行方式，可能还有另外两种方式："Run-time WorkBench"和"Run-time WorkBench with Tracing"，当用 PDE 开发插件的时候会用到这两种运行方式，在下面我们也有提到。

4. 版本管理

可以把 Eclipse 的版本管理分为个人（或称为本地）和团队两种。

Eclipse 提供了强大的个人版本管理机制，每一次被保存的更改都可以得到恢复。而且可以精确到每一个方法的版本恢复。操作也十分方便，在任何一个能看到所要操作文件的观察窗口中，例如资源浏览窗口，选中该文件，点击鼠标右键，选择 Compare with 或 Replace with，如果是恢复已经被删除的方法则可以选择 Add from local history，之后相应的本地历史记录就会显示出来，按照你的需求找到相应的版本就可以了。强大的个人版本管理功能为程序员提供了更多的信心：只管编下去，任何不小心的错误都可以恢复，在 Eclipse 下开发，是有"后悔药"的！

Eclipse 缺省为版本管理工具 CVS 提供了接口，可以非常方便的连接到 CVS 服务器上。通过 CVS 版本管理，Eclipse 为团队开发提供良好的环境。要连接 CVS 服务器需要先打开团队视图（Team Perspective），然后在 Repositories 观察窗口中点击鼠标右键并选择新建（New），在打开的对话框中可以填入要连接的 CVS 库所需要的信息，如 CVS 服务器类型，目前 Eclipse 支持三种方式 pserver、extssh 和 ext，还要填入用户名，主机名，密码，repository 地址等信息。

在 Eclipse 中使用 CVS 需要注意的是一些术语和功能的变化，CVS 中的 Branch 这里被称为 Stream，取消了 CVS 中 check out、import 和 commit 等功能，统统用鼠标右键菜单中的 Team->Synchronized with Stream 来替代。这些功能都通过图形界面完成，在每次操作中都会有当前文件和以前各个版本的比较窗口，操作非常直观，易于掌握，因此这里也就不再做进一步介绍了。

5. 使用插件

使用插件可以丰富 Eclipse 的功能。下面将介绍如何应用插件来嵌入 Tomcat 服务器。这个插件并不是 Eclipse 项目组开发的，而是一家叫 sysdeo 的公司开发，非常小巧，只有 27.8K。你可以到 <http://www.sysdeo.com/eclipse/tomcatPlugin.html> 去免费下载。另外，这个插件只支持 Tomcat 4.0 以上的版本，可以在 www.apache.org 得到 Tomcat 的最新版本。

要安装插件只需将下载的 zip 文件按原路径解压到"你的 Eclipse 的安装路径\plugins"下面，然后重新启动 Eclipse。启动后在菜单栏上选择 PerspectiveàCustomize，在打开的对话框中选中 OtheràTomcat。之后马上会发现 Eclipse 有了两处变化：菜单栏中多了一个 Tomcat 选项，工具栏中多了两个按钮，上面是大家可能非常熟悉的 Tomcat 小猫，如下图三。除此之外，在菜单栏中选择：WorkbenchàPreferences，打开对话框后会发现这也多了一个 Tomcat 选项，在这里要求指定你的 Tomcat 安装根路径。之后还要检查一下在 Preferences 对话框中的 JavaàInstalled JRE 所指定的 JRE 和启动 Tomcat 的 JRE 是否为同一个 JRE，如果不是，可能会导致 Tomcat 不能正常启动。如果以上检查没有问题，就可以用工具栏上的"小猫"直接启动 Tomcat 了。要注意的是，启动过程很慢，要耐心等到以下信息出现：

```
Starting service Tomcat-Standalone
Apache Tomcat/4.0.1
```

Starting service Tomcat-Apache Apache Tomcat/4.0.1

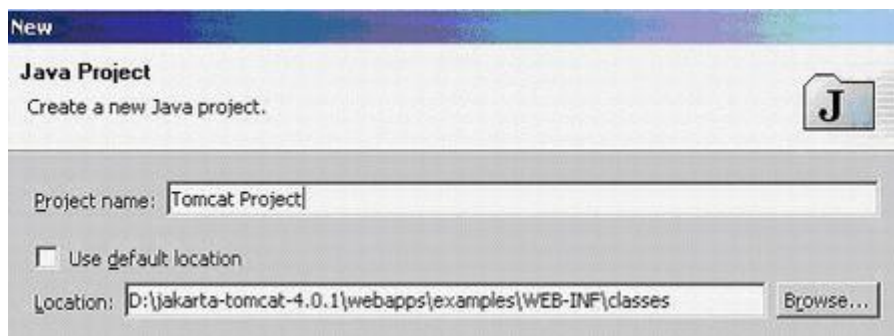
之后就可以在外部浏览器（如 IE）中输入 `http://localhost:8080` 来测试 Tomcat 是否正常。



图三

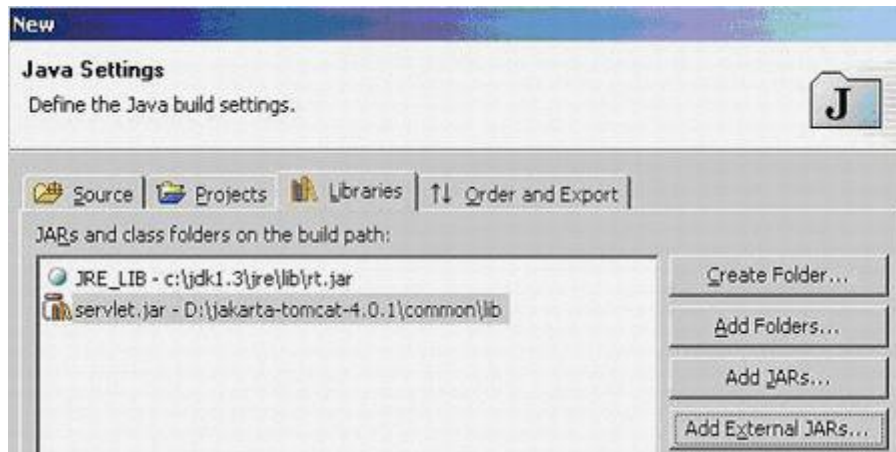
如果启动正常,可以进一步尝试在 Eclipse 中调试 Servlet 或 JSP 程序。下面我们将用 Tomcat 自带的 Servlet 例程 HelloWorldExample.java 来示范一下如何在 Eclipse 中调试 Sevlet 程序。

首先要在 Java 视图中新建一个 Java 项目 ,为了方便 ,可以直接将项目路径指定到 HelloWorldExmaple 程序所在路径 ,如图四 :



图四

之后按"Next",进入 Java Settings 对话框,选择 Libraries 标签,并用 Add External JARs 按钮来指定 Servlet.jar 包的位置。这里直接用了 Tomcat 中的 Servlet.jar 包。如图五 :



图五

最后，点击"Finish"完成项目的生成。在新生成项目中的 default package 可以找到 HelloWorldExample.java，双击打开文件，并可尝试给 HelloWorldExample 加上一个断点（双击编辑窗口左侧边界）。之后在外部浏览器中输入 <http://localhost:8080/examples/servlet/HelloWorldExample>，再回过头来看 Eclipse 发生了什么变化，是一个调试窗口呀！在 Eclipse 中进行的调试操作和绝大多数的 IDE 大同小异，如设置断点，单步跟踪，变量值察看等等，在这里也就不再详述了。

eclipse使用方法（转帖）

2008-10-09 21:43

目前 Java开发领域的各种集成开发环境（IDE）呈现出百花齐放的局面，从 Borland的 JBuilder,到 IBM的 Visual Age for Java WebSphere Studio,Oracle 的 JDeveloper, Sun的 Forte for Java,WebGain的 Visual Cafe, TogetherSoft 的 Together, 还有开放源代码的 Eclipse NetBeans 等，种类 10种有余。这么多种类的 IDE繁荣了 Java开发工具家族，但是也为开发人员的选择提出了难题。这些 IDE的开发环境有着较大的差别，在一种开发环境下开发的项目不能很方便地移植到另一种开发环境，这就要求更为谨慎地选择适合项目目标的开发工具。

在目前所有的 IDE中，Eclipse可以说是最有发展前途的产品之一。Eclipse最初由 OTI和 IBM两家公司的 IDE产品开发组创建，起始于 1999年 4月。IBM提供了最初的 Eclipse代码基础，包括 Platform JDT和 PDE，目前由 IBM牵头，围绕着 Eclipse项目已经发展成为了一个庞大的 Eclipse联盟，有 150多家软件公司参与到 Eclipse项目中，其中包括 Borland Rational Software Red Hat 及 Sybase, 最近 Oracle也计划加入到 Eclipse联盟中。

基本概念

在学习使用 Eclipse之前，有必要对关于这个项目的名词做一些解释。

Eclipse是一个开放源代码的软件开发项目，专注于为高度集成的工具开发提供一个全功能的、具有商业品质的工业平台。它由 Eclipse项目、Eclipse工具项目和 Eclipse技术项目三个项目组成，每一个项目由一个项目管理委员会监督，并由它的项目章程管理。每一个项目由其自身的子项目组成，并且使用 Common Public License(CPL) 版本 1.0许可协议。

Eclipse工具项目为不同的工具建造者提供一个焦点，以保证为 Eclipse Platform创建最好的工具。Eclipse工具项目的任务是为 Eclipse Platform培育广泛的工具的创建。工具项目提供单一的联系人以调和开放源代码工具建造者，从而使得覆盖和重复最小化，并保证共享的最大化和共同组件的创建，促进不同类型工具的无缝互操作。工具项目由工具开发者委员会和工具项目的项目管理委员会提议、选择和开发的子项目组成。

Eclipse技术项目的任务是为开放源代码开发者、研究者、学院和教育者提供新的管道，以参与将来 Eclipse的演化。它按照研究、培育和教育三个项目流来组织，研究项目在 Eclipse相关领域诸如编程语言、工具和开发环境方面进行探索和研究；培育项目是小型的、未正式结构化的项目，为 Eclipse软件基础添加新的能力；教育项目聚焦于教育材料的开发、教学帮助和课件。

Eclipse Platform是一个开放的可扩展的 IDE。Eclipse Platform提供建造块和构造并运行集成软件开发工具的基础。Eclipse Platform允许工具建造者独立开发与他人工具无缝集成的工具你无须分辨一个工具功能在哪里结束，而另一个工具功能在哪里开始。

Eclipse SDK(软件开发者包) 是 3个 Eclipse项目的子项目 (Platform JDT PDE) 所生产的组件合并，它们可以一次下载。这些部分在一起提供了一个具有丰富特性的开发环境，允许开发者有效地建造可以无缝集成到 Eclipse Platform中的工具。Eclipse SDK由 Eclipse项目生产的工具和来自其它开放源代码的第三方软件组合而成。Eclipse项目生产的软件以 CPL发布，第三方组件有各自自身的许可协议。

下载、安装 Eclipse

知道了这些关于 Eclipse项目的信息，我们可以下载、安装一个 Eclipse。Eclipse SDK就是我们要下载的，其中包括了开发 Java应用的所有内容和 Eclipse项目的所有成果。

进入 Eclipse项目的主页 <http://www.eclipse.org>, 点击 Downloads, 进入下载页。Eclipse最新的版本是 2.0.2版, 但是 2.0.2版的多国语言翻译包还没有出来, 所以我们可以使用一个稍早一点的版本 2.0.1。点击 2.0.1进入其下载页, 这个页面有很多下载连接, 包括 Eclipse SDK在很多平台上的版本, 这里我们只需要下载 eclipse-SDK-2.0.1-win32.zip和 NLS-SDK-2.0.1-Translations.zip 两个文件就可以了, 另外还有一个 eclipse-examples-2.0.1-win32.zip文件, 是学习《Java开发用户指南》所需要的。将这 3个文件解压缩到相同的目录, 如 D:\eclipse

在运行 Eclipse之前首先应该安装好 JDK(Sun 的 JDK或 IBM的 JDK都可以, 应该安装 1.3以上版本, 推荐使用 1.4以上版本, 因为只有使用 1.4以上版本的 JDK才可以享受到新增的 HotSwap功能对于调试带来的方便), 设置好环境变量 JAVA_HOME CLASSPATH和 PATH

为 D:\eclipse下的 eclipse.exe在桌面上建立一个快捷方式。双击快捷方式启动 Eclipse。Eclipse也可以以命令行方式启动, 在命令行上指定使用的 JVM和保存数据的 workspace, 例如:

```
D:\eclipse\eclipse.exe -vmD:\WebSphere\AppServer\java
\bin\javaw.exe -data
D:\WORK\workspace
```

其中, D:\WORK\workspace是保存项目文件的位置。

可以选择帮助 帮助内容进入 Eclipse的帮助系统。有了 Eclipse的多国语言翻译包, 所有的选单和联机帮助都已经翻译成了中文, 使用起来非常方便。

在进行下面的学习前, 建议先简单浏览一下帮助中的《工作台用户指南》, 熟悉一下透视图和视图等概念。

编写 Hello World程序

下面我们编写一个 HelloWorld程序, 步骤如下:

1. 选择文件 新建 项目;
2. 项目类别选 Java;
3. 项目列表选 Java项目;
4. 点击 下一步 ;
5. 输入项目名称, 例如: HelloWorldProject;

6. 点击 完成 ；

7. 在工具条里点击 创建 Java类 的按钮（带有一个 C 标记）见图 1；

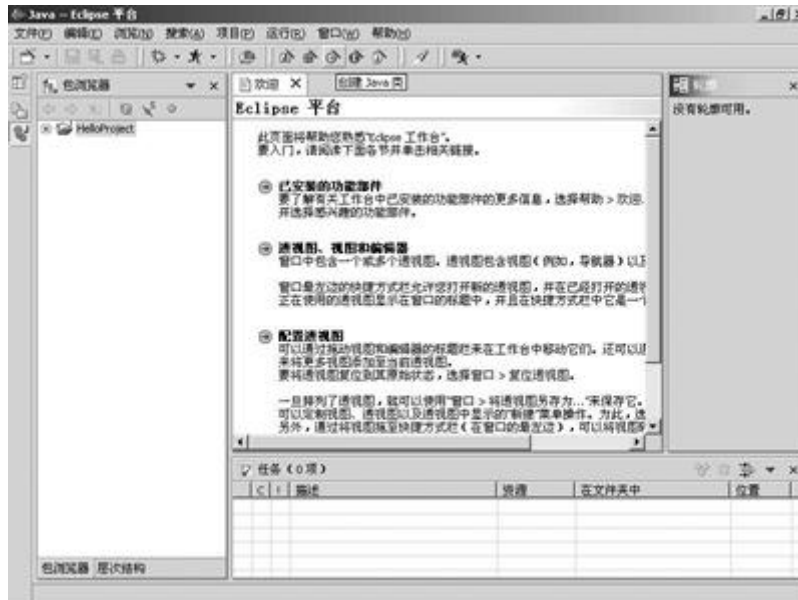


图 1 点击 创建 Java类 的按钮图

8. 在名称域输入 HelloWorld;

9. 点击 public static void main(String[] args) 的复选框，让 Eclipse 创建 main 方法（见图 2）；



图 2 Eclipse创建 main方法

10. 点击 完成 ；
11. 一个 Java编辑窗口将打开，在 main方法中输入
`System.out.println(HelloWorld)`行；
12. 使用 Ctrl-s保存，这将自动编译 HelloWorld.java；
13. 点击工具条里的 运行 按钮；
14. 选择 Java应用程序 ，然后选 新建 ；
15. 输入项目名称 Hello- Project和 main类名 HelloWorld；
16. 点击 运行 ；
17. 这时候，将会打开一个控制台窗口，一句 HelloWorld 将会显示在里面（见图 3）。

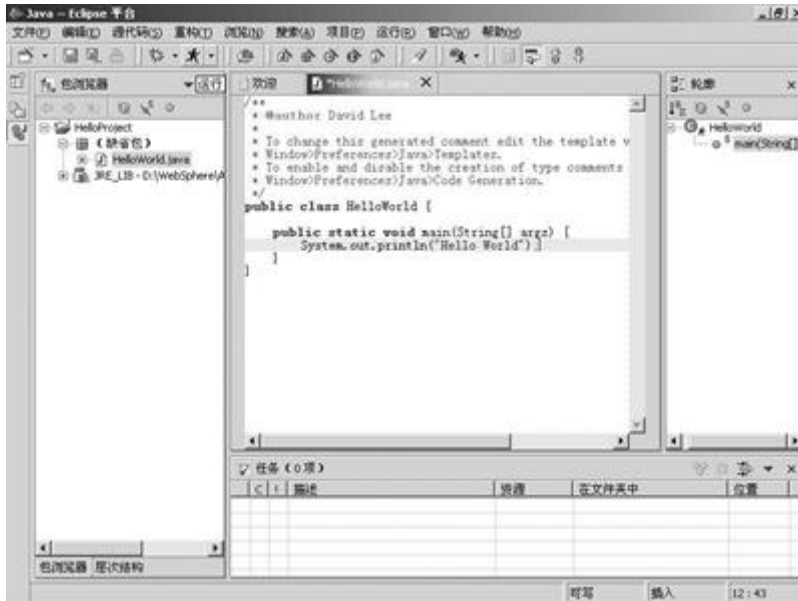


图 3 "Hello Wbrld"显示窗口

Eclipse有一个增量编译器，每次保存一个 Java文件时它就自动进行编译。这个特性被称做 自动构建 。如果不需要这个功能，可以在窗口 首选项 工作台 对资源修改执行自动构建，关闭这个特性。随后工具条中会出现 全部构建 的按钮。类似的选项也会出现在相关的选单中。

创建 CVS资源库

现在我们知道如何创建一个项目了，下一步是与某个版本服务器建立连接，将项目发布到版本服务器中。目前开源软件都是使用 CVS来做版本管理，因此我们也使用 CVS来举例。Eclipse SDK已经包括了支持 CVS的插件，要想使用其它的版本管理软件，例如 VSS,可以从 Eclipse的网站上下载相关的插件。创建一个 CVS资源库位置的步骤如下：

1. 打开 CVS资源库开发透视图（窗口 打开透视图 其它 CVS资源库开发）。
2. 在 CVS资源库视图里点右键，新建 资源库位置，见图 4

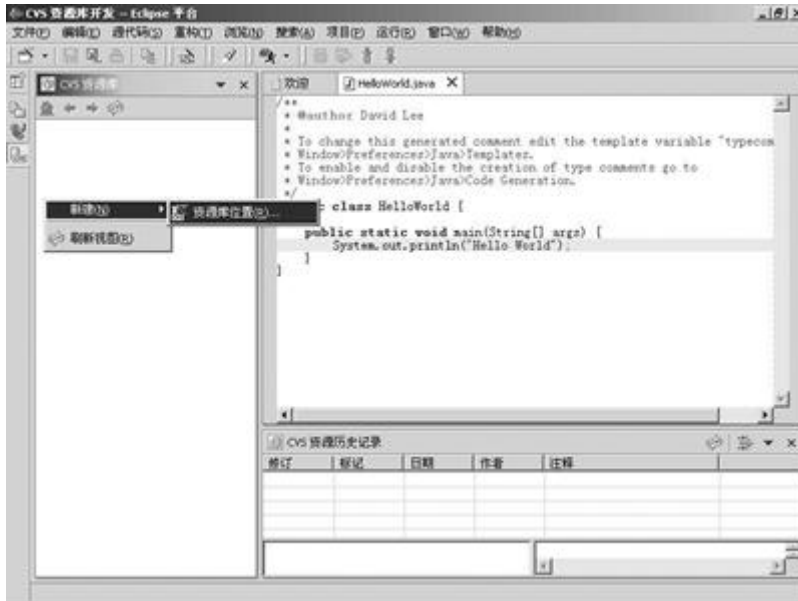


图 4 新建资源库位置

3. 在添加 CVS资源库对话框中输入 CVS Server 的详细信息。
4. 点击 完成 。
5. 一旦配置好 CVS资源库，就可以把项目输入资源库了。转到 Java透视图，在项目名称上点右键，选择小组 共享项目。
6. 选择刚才添加的资源库。
7. 点击 完成 。
8. 在同步窗口中的项目上点右键，选提交。
9. 输入为提交加入的注释，如图 5

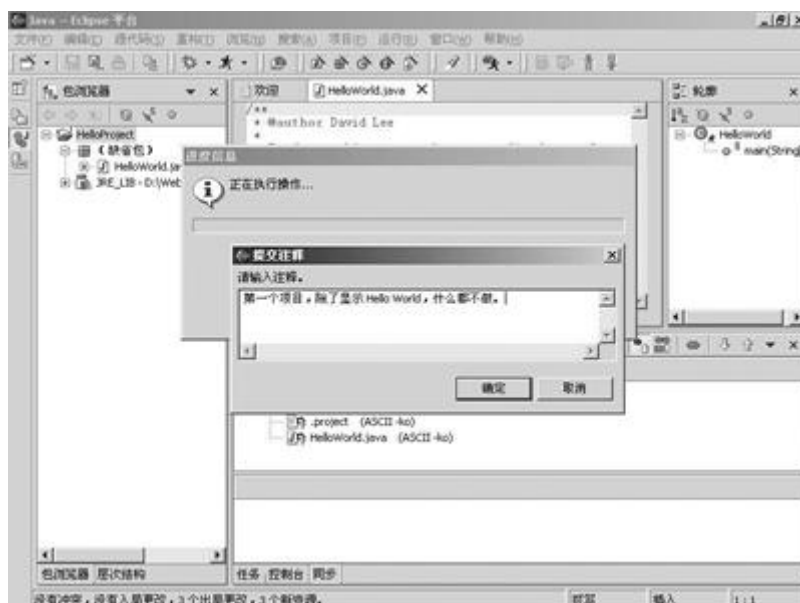


图 5 为提交加入注释

10. 点击 **确定**，则项目被发布到 CVS 资源库中。

11. 将项目共享到资源库中后，可以使用 **与资源库同步** 功能来发布代码，并且赶上别人对资源库中代码所做的修改。在项目上点右键，**同步** 与资源库同步。

下载、安装应用服务器 JBoss

为了做服务器端的开发，还需要一个应用服务器，并且把应用服务器与 Eclipse 集成起来。Eclipse 可以与多种应用服务器共同工作，包括 WebLogic、WebSphere、JRun、Orion/Oracle9iAS、JBoss、JOnAS、Resin 等。下面以目前流行的开放源代码应用服务器 JBoss 来举例。首先要下载并安装好 JBoss，我使用的是 JBoss 3.0.3，安装在 D:\jboss-3.0.3 目录。为了把 JBoss 与 Eclipse 相集成，还需要下载相关的插件，下载位置是 http://www.genuitec.com/products_easie.htm <http://www.genuitec.com/products/EASIEJBoss1.0.6.zip> 在 <http://www.genuitec.com/products/EclipseJBoss.pdf> 还有一个安装的指导。退出 Eclipse，将刚才下载的文件解压缩到 D:\eclipse 下，重新启动 Eclipse。

新安装的插件需要激活才能使用，选择窗口 **定制透视图**，将 **其它** 中的 **EASIE JBoss** 项选中。

然后配置这个插件，选择窗口 **首选项**，输入关于 JBoss 插件的信息。

以上两步执行完后，在工具条中会出现启动和停止 JBoss 的按钮。点击启动按钮启动 JBoss，如图 6

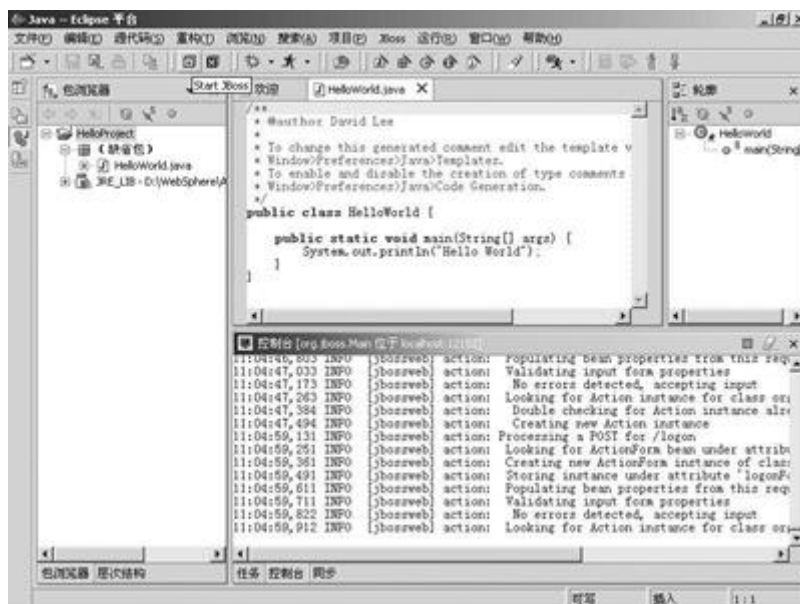


图 6 启动 JBoss

打开调试透视图，在调试透视图将会看到刚才启动的 JBoss 进程相关的信息。

停止 JBoss 可以使用启动按钮旁边的停止按钮。在这个网站上还有支持 WebLogic WebSphere Orion/Oracle 9iAS 的插件，安装方法基本上完全一样。

Eclipse 调试

到现在为止，建立一个完整的开发环境所需要的配置基本上都完成了。下面我们回到 HelloWorld 项目，看一下调试程序需要做什么事情。

用过 IDE 的人对于断点、单步运行等概念是非常熟悉的，Eclipse 具有所有 IDE 应该具有的调试功能。调试步骤如下：

1. 首先在程序中设置一个断点，双击语句左边的灰色区域，则一个断点被加到该语句上，如图 7；

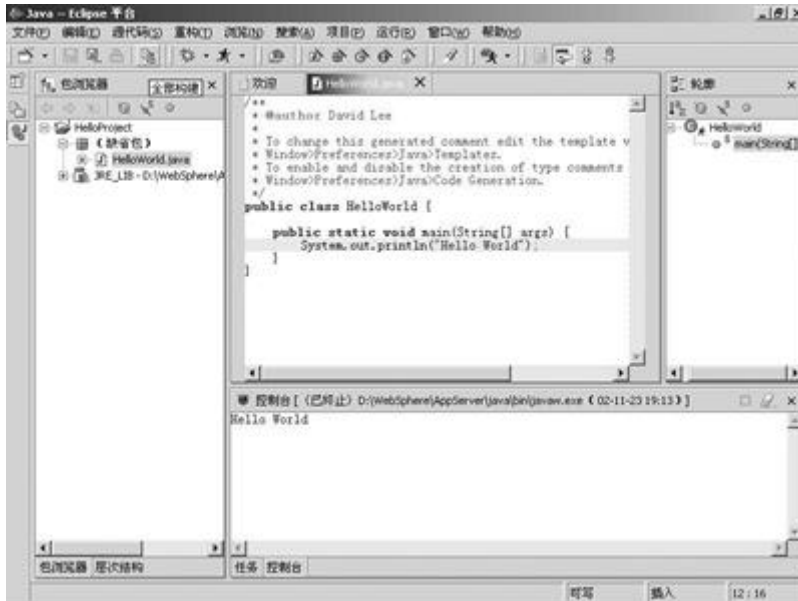


图 7 设置断点

2. 点击工具条中的调试按钮（一个 Bug）；
3. 在 Java 应用程序下选择 HelloWorld；
4. 点击 调试 ，程序将运行到刚才设置的断点处，然后停止运行；
5. 按 F6，单步执行，这时候程序运行到下一条语句；
6. 按 F8 继续运行到程序结束。

导入 JUnit 例子程序

下面我们导入一个真正的项目，这个项目来自于 Eclipse JDT 的例子。

1. 选择文件 导入；
2. 选择 ZIP 文件 ；
3. 浏览并找到源代码 Jar 文件所在位置，在文件夹中输入 JUnit；
4. 点击 完成 ，将导入一个新的 JUnit 项目，如图 8

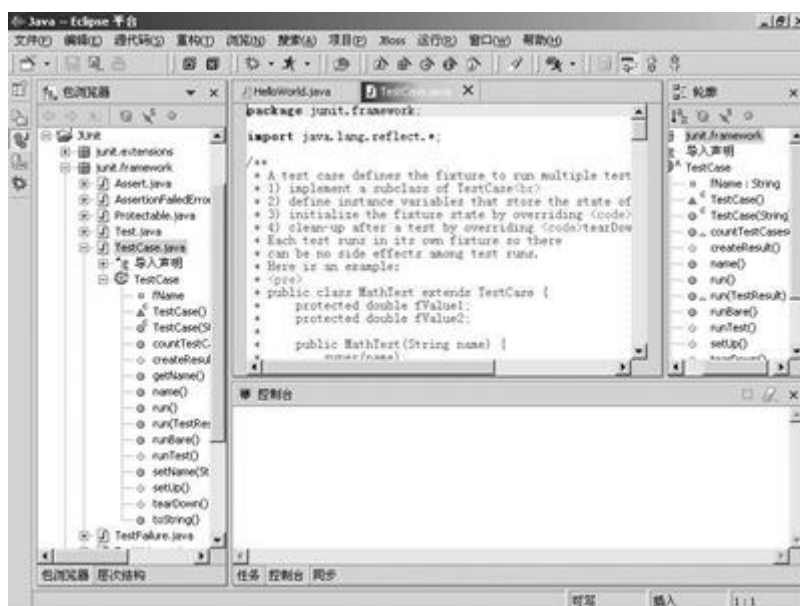


图 8 导入项目

限于篇幅，这里就不详细介绍这个项目的开发过程了。具体的开发过程请看帮助中的《Java开发用户指南》。

与其它流行的 Java IDE一样，在 Eclipse中也可以获得内容帮助，获得内容帮助的热键是 Alt+/.。

在 Eclipse SDK中已经集成了 Ant和 JUnit这些建造和测试项目的工具，还有强大的机制支持代码的重构（refactoring）。这些内容在《Java开发用户指南》中有较为详细的介绍。看了这篇 Eclipse的入门文章后，就可以直接看《Java开发用户指南》。通过对《Java开发用户指南》的学习，可以感受到 Eclipse正是我们需要的支持 XP（极限编程）快速开发方法的新一代 IDE。

Eclipse的日志文件保存在 workspace/.metadata/.log文件中，这个文件用来保存运行时的错误，其中的信息在除错时非常有用。

介绍了 Eclipse中 Java的开发过程，你可能对于使用 Eclipse开发其它语言也感兴趣。Eclipse Platform是一个通用的工具平台。它并不知道任何特定的语言，但是打开了一扇门，使得插件可以支持各种语言。Eclipse SDK包括了一个全功能的 Java IDE。这是一个如何扩展平台以支持一种特殊的语言的例子。SDK还提供 Java文件的编辑器插件和 XML文件的编辑器插件。

Eclipse 用法小全收藏

热键篇：

Template：Alt + /

修改处：窗口->喜好设定->工作台->按键->编辑->内容辅助。

个人习惯：Shift+SPACE(空白)。

简易说明：编辑程序代码时，打 sysout +Template 启动键，就会自动出现：System.out.println(); 。

设定 Template 的格式：窗口->喜好设定->Java->编辑器->模板。

程序代码自动排版：Ctrl+Shift+F

修改处：窗口->喜好设定->工作台->按键->程序代码->格式。

个人习惯：Alt+Z。

自动排版设定：窗口->喜好设定->Java->程序代码格式制作程序。

样式页面->将插入 tab(而非空格键)以内缩，该选项取消勾选，下面空格数目填 4，这样在自动编排时会以空格 4 作缩排。

快速执行程序：Ctrl + F11

个人习惯：ALT+X

修改处：窗口->喜好设定->工作台->按键->执行->启动前一次的启动作业。

简易说明：第一次执行时，它会询问您执行模式，设置好后，以后只要按这个热键，它就会快速执行。

<ALT+Z(排版完)、ATL+X(执行)>..我觉得很顺手^___^

自动汇入所需要的类别：Ctrl+Shift+O

简易说明：

假设我们没有 Import 任何类别时，当我们在程序里打入：

```
BufferedReader buf =  
new BufferedReader(new InputStreamReader(System.in));
```

此时 Eclipse 会警示说没有汇入类别，这时我们只要按下 Ctrl+Shift+O，它就会自动帮我们 Import 类别。

查看使用类别的原始码：Ctrl+鼠标左键点击

简易说明：可以看到您所使用类别的原始码。

将选取的文字批注起来：Ctrl+/

简易说明：Debug 时很方便。

修改处：窗口->喜好设定->工作台->按键->程序代码->批注

视景切换：Ctrl+F8

个人习惯：Alt+S。

修改处：窗口->喜好设定->工作台->按键->窗口->下一个视景。

简易说明：可以方便我们快速切换编辑、除错等视景。

密技篇：

一套 Eclipse 可同时切换，英文、繁体、简体显示：

1.首先要先安装完中文化包。

2.在桌面的快捷方式后面加上参数即可，

英文-> -nl "zh_US"

繁体-> -nl "zh_TW"

简体-> -nl "zh_CN"。

(其它语系以此类推)

像我 2.1.2 中文化后，我在我桌面的 Eclipse 快捷方式加入参数-nl "zh_US"。

"C:\Program Files\eclipse\eclipse.exe" -n "zh_US"

接口就会变回英文语系噜。

利用 Eclipse，在 Word 编辑文书时可不必将程序代码重新编排：

将 Eclipse 程序编辑区的程序代码整个复制下来(Ctrl+C)，直接贴(Ctrl+V)到

Word 或 WordPad 上，您将会发现在 Word 里的程序代码格式，跟 Eclipse

所设定的完全一样，包括字型、缩排、关键词颜色。我曾试过 JBuilder

、GEL、NetBeans...使用复制贴上时，只有缩排格式一样，字型、颜

色等都不会改变。

外挂篇：

外挂安装：将外挂包下载回来后，将其解压缩后，您会发现 features、

plugins 这 2 个数据夹，将里面的东西都复制或移动到 Eclipse 的 features

、plugins 数据夹内后，重新启动 Eclipse 即可。

让 Eclipse 可以像 JBuilderX 一样使用拖拉方式建构 GUI 的外挂：

1.Jigloo SWT/Swing GUI Builder：

<http://cloudgarden.com/jigloo/index.html>；

下载此版本：Jigloo plugin for Eclipse (using Java 1.4 or 1.5)

安装后即可由档案->新建->其它->GUI Form 选取要建构的 GUI 类型。

2.Eclipse Visual Editor Project：

<http://www.eclipse.org/vep/>；

点选下方 Download Page，再点选 Latest Release 0.5.0 进入下载。

除了 VE-runtime-0.5.0.zip 要下载外，以下这 2 个也要：

EMF build 1.1.1: (build page) (download zip)

GEF Build 2.1.2: (build page) (download zip)

3.0 M8 版本，请下载：

EMF build I200403250631

GEF Build I20040330

VE-runtime-1.0M1

安装成功后，便可由 File->New->Visual Class 开始 UI 设计。
安装成功后，即可由新建->Java->AWT 与 Swing 里选择所要建构的 GUI 类型开始进行设计。VE 必须配合着对应版本，才能正常使用，否则即使安装成功，使用上仍会有问题。

使用 Eclipse 来开发 JSP 程序：

外挂名称：lomboz(下载页面)

http://forge.objectweb.org/project/showfiles.php?group_id=97；

请选择适合自己版本的 lomboz 下载，lomboz.212.p1.zip 表示 2.1.2 版，lomboz.3m7.zip 表示 M7 版本....以此类推。

lomboz 安装以及设置教学：

Eclipse 开发 JSP-教学文件

Java 转 exe 篇：

实现方式：Eclipse 搭配 JSmooth(免费)。

1.先由 Eclipse 制作包含 Manifest 的 JAR。

制作教学

2.使用 JSmooth 将做好的 JAR 包装成 EXE。

JSmooth 下载页面：

<http://jsmooth.sourceforge.net/index.php>；

3.制作完成的 exe 文件，可在有装置 JRE 的 Windows 上执行。

Eclipse-Java 编辑器最佳设定：

编辑器字型设定：工作台->字型->Java 编辑器文字字型。

(建议设定 Courier New -regular 10)

编辑器相关设定：窗口->喜好设定->Java->编辑器

外观：显示行号、强调对称显示的方括号、强调显示现行行、显示打印边距，将其勾选，Tab 宽度设 4，打印编距字段设 80。

程序代码协助：采预设即可。

语法：可设定关键词、字符串等等的显示颜色。

附注：采预设即可。

输入：全部字段都勾选。

浮动说明：采预设即可。

导览：采预设即可。

使自动排版排出来的效果，最符合 Java 设计惯例的设定：

自动排版设定：窗口->喜好设定->Java->程序代码制作格式。

换行：全部不勾选。

分行：行长度上限设：80。
样式：只将强制转型后插入空白勾选。
内缩空格数目：设为 4。

Eclipse 的教学文件：

Eclipse 3.0 系列热键表 - 中英对照解说版 (by sungo) ~New~
Window+GCC+CDT 用 Eclipse 开发 C、C++ (by sungo) ~New~

本文来自 CSDN 博客，转载请标明出处：
<http://blog.csdn.net/rigger21/archive/2007/06/09/1645913.aspx>

您正在看的 JAVA 教程是:Eclipse 平台入门。

Eclipse 平台入门

作者：David Gallardo 来自：IBM

本文为您提供关于 Eclipse 平台的概述，包括其起源和体系结构。本文首先简要讨论 Eclipse 的开放源代码性质及其对多种编程语言的支持，然后通过一个简单的程序例子展示 Java 开发环境。本文还将考查以插件扩展形式可用的一些软件开发工具，并展示一个用于 UML 建模的插件扩展。

Eclipse 是什么？

Eclipse 是一个开放源代码的、基于 Java 的可扩展开发平台。就其本身而言，它只是一个框架和一组服务，用于通过插件组件构建开发环境。幸运的是，Eclipse 附带了一个标准的插件集，包括 Java 开发工具（Java Development Tools，JDT）。

虽然大多数用户很乐于将 Eclipse 当作 Java IDE 来使用，但 Eclipse 的目标不仅限于此。Eclipse 还包括插件开发环境（Plug-in Development Environment，PDE），这个组件主要针对希望扩展 Eclipse 的软件开发人员，因为它允许他们构建与 Eclipse 环境无缝集成的工具。由于 Eclipse 中的每样东西都是插件，对于给 Eclipse 提供插件，以及给用户提供一致和统一的集成开发环境而言，所有工具开发人员都具有同等的发挥场所。

这种平等和一致性并不仅限于 Java 开发工具。尽管 Eclipse 是使用 Java 语言开发的，但它的用途并不限于 Java 语言；例如，支持诸如 C/C++、COBOL 和 Eiffel 等编程语言的插件已经可用，或预计会推出。Eclipse 框架还可用来作为与软件开发无关的其他应用程序类型的基础，比如内容管理系统。

基于 Eclipse 的应用程序的突出例子是 IBM 的 WebSphere Studio Workbench，它构成了 IBM Java 开发工具系列的基础。例如，WebSphere Studio Application Developer 添加了对 JSP、servlet、EJB、XML、Web 服务和数据库访问的支持。

Eclipse 是开放源代码的软件

开放源代码软件是这样一种软件，它们在发布时附带了旨在确保将某些权利授予用户的许可证。当然，最明显的权利就是源代码必须可用，以便用户能自由地修改和再分发该软件。这种用户权利的保护是通过一种称为 copyleft 的策略来完成的：软件许可证主张版权保护，除非明确授予用户这样的权利，否则用户不得分发该软件。copyleft 还要求同一许可证涵盖任何被再分发的软件。这实际上倒置了版权的目的——使用版权来授予用户权利，而不是为软件的开发者保留版权——copyleft 经常被描述为“保留所有版权”。

曾经四处蔓延的对开放源代码软件的许多恐惧、担忧和疑虑，都与某些 copyleft 许可证的所谓“病毒”性质有关——如果使用开放源代码软件作为您开发的程序的一部分，您将失去自己的知识产权，因为该许可证将“传染”您开发的专有部分。换句话说，该许可证可能要求与开放源代码软件一起打包的所有软件，都必须在相同的许可证之下发布。虽然这对最著名的 copyleft 许可证（即 GNU 通用公共许可证，例如 Linux 就是在该许可证之下发布的）来说可能是事实，当时还有其他许可证在商业化和社区考虑之间提供了较好的平衡。

开放源代码计划（Open Software Initiative）是一家非营利机构，它明确定义了开放源代码的含义及满足其标准的认证许可证。Eclipse 是在 OSI 认可的通用公共许可证（CPL）1.0 版之下被授予许可证的，CPL“旨在促进程序的商业化使用……”（欲获得指向通用公共许可证 1.0 版完整文本的链接，请参阅本文稍后的参考资料）。

为 Eclipse 创建插件或将 Eclipse 用作软件开发应用程序基础的开发人员，需要发布他们在 CPL 下使用或修改的任何 Eclipse 代码，但是他们可以自由决定自己添加的代码的许可证授予方式。与出自 Eclipse 的软件一起打包的专有代码不需要作为开放源代码来授予许可证，该源代码也不需要提供给用户。

尽管大多数开发人员不会使用 Eclipse 来开发插件，或创建基于 Eclipse 的新产品，但是 Eclipse 的开放源代码性质所意味的，并不只是它使得 Eclipse 免费可用（尽管便于商业化的许可证意味着插件可能要花钱）。开放源代码鼓励创新，并激励开发人员（甚至是商业开发人员）为公共开放源代码库贡献代码。对此存在许多原因，不过最本质的原因或许是为这个项目作贡献的开发人员越多，这个项目就会变得对每个人都越宝贵。随着这个项目变得

更加有用，更多的开发人员将会使用它，并围绕它形成一个社区，就像那些围绕 Apache 和 Linux 形成的社区一样。

Eclipse 是什么机构？

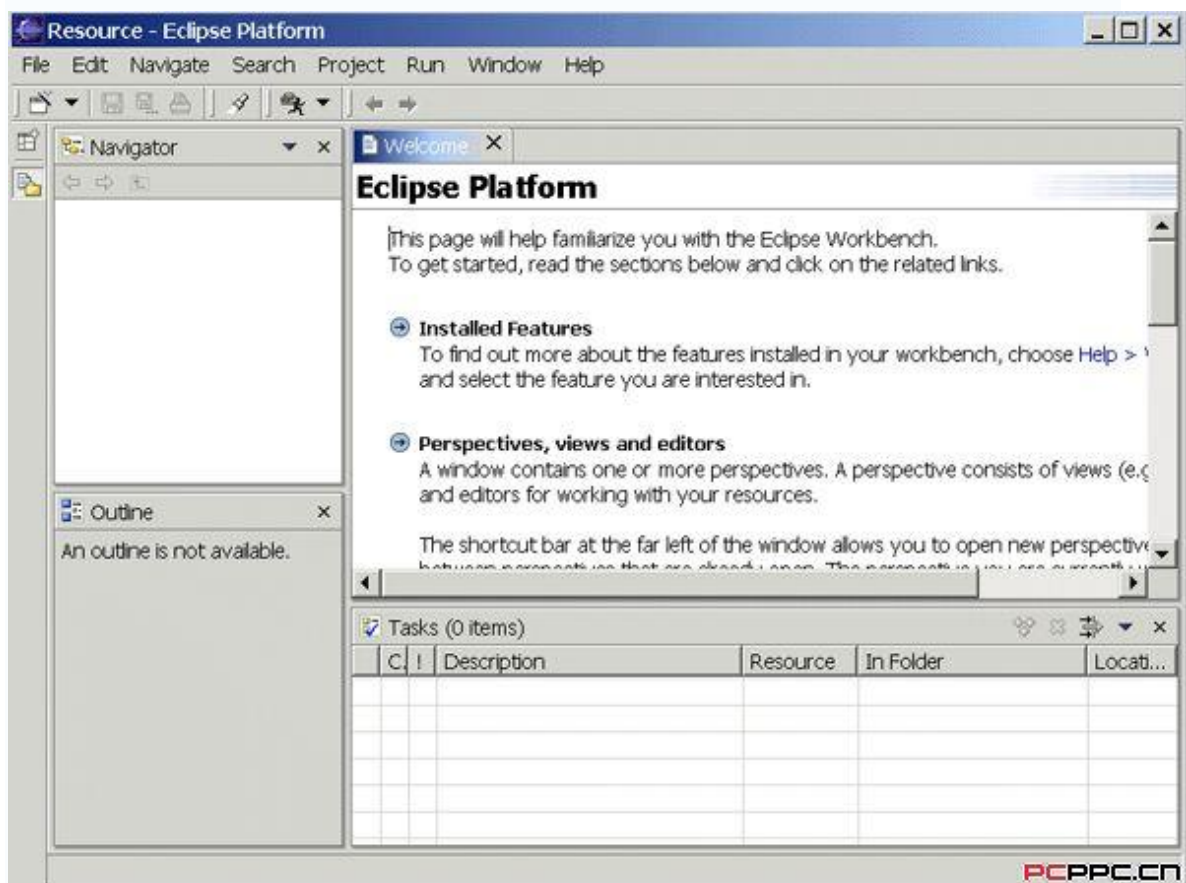
Eclipse.org 协会管理和指导 Eclipse 正在进行中的开发。在据说 IBM 花了 4000 万美元开发 Eclipse，并把它作为一个开放源代码项目发布之后，Eclipse.org 协会吸收了许多软件工具提供商，包括 Borland、Merant、Rational、RedHat、SuSE、TogetherSoft 和 QNX。从那以后还有其他公司相继加入，包括 Hewlett Packard、Fujitsu、Sybase。这些公司分别向理事会派了一名代表，这个理事会负责确定 Eclipse 项目的方向和范围。

在最高层，项目管理委员会 (Project Management Committee , PMC) 管理着 Eclipse 项目。这个项目被划分为多个子项目，每个子项目都有一名负责人。大型子项目又被划分为组，每个组也有一名负责人。目前，这其中的大多数管理角色都由最初开发 Eclipse 的 IBM 子公司 Object Technology International (OTI)的人担任，但是作为一个开放源代码的项目，它欢迎任何人的参与。任何特定部门的职责是通过该部门对项目的贡献来争取的。

现在我们已经考察了 Eclipse 背后的一些理论、历史和管理，下面让我们考察该产品本身。

Eclipse 工作台

在第一次打开 Eclipse 时，首先看到的是下面的欢迎屏幕：



图文

图 1. Eclipse 工作台

Eclipse 工作台由几个称为 视图 (view) 的窗格组成, 比如左上角的 Navigator 视图。窗格的集合称为 透视图 (perspective)。默认的透视图是 Resource 透视图, 它是一个基本的通用视图集, 用于管理项目以及查看和编辑项目中的文件。

Navigator 视图 允许您创建、选择和删除项目。Navigator 右侧的窗格是 编辑器区域。取决于 Navigator 中选定的文档类型, 一个适当的编辑器窗口将在这里打开。如果 Eclipse 没有注册用于某特定文档类型(例如, Windows 系统上的 .doc 文件)的适当编辑器, Eclipse 将设法使用外部编辑器来打开该文档。

Navigator 下面的 **Outline** 视图 在编辑器中显示文档的大纲; 这个大纲的准确性取决于编辑器和文档的类型; 对于 Java 源文件, 该大纲将显示所有已声明的类、属性和方法。

Tasks 视图 收集关于您正在操作的项目的信息; 这可以是 Eclipse 生成的信息, 比如编译错误, 也可以是您手动添加的任务。

该工作台的大多数其他特性，比如菜单和工具栏，都应该和其他那些熟悉的应用程序类似。一个便利的特性就是不同透视图的快捷方式工具栏，它显示在屏幕的左端；这些特性随上下文和历史的不同而有显著差别。Eclipse 还附带了一个健壮的帮助系统，其中包括 Eclipse 工作台以及所包括的插件（比如 Java 开发工具）的用户指南。至少浏览一遍这个帮助系统是值得的，这样可以看到有哪些可用的选项，同时也可更好地理解 Eclipse 的工作流程。

为继续这个短暂的

您正在看的 JAVA 教程是:Eclipse 平台入门。 Eclipse 之旅，我们将在 Navigator 中创建一个项目。右键单击 Navigator 视图，然后选择 **New=>Project**。当 New Project 对话框出现时，选择左面的 Java。标准 Eclipse 只有一种 Java 项目类型，名为“Java Project”。如果安装了插件来提供 JSP 和 servlet 支持，我们会从这里看到一个用于 Web 应用程序的附加选项。眼下，请选择 Java Project，在提示项目名称时输入“Hello”，然后按 Finish。

接下来，我们将检查一下 Java 透视图。取决于您喜欢的屏幕管理方式，您可以通过选择 **Window=>Open Perspective=>Java** 来改变当前窗口中的透视图，也可以通过选择 **Window=>New Window**，然后再选择这个新的透视图，从而打开一个新的窗口。

正如您可能预期的那样，Java 透视图包含一组更适合于 Java 开发的视图。其中之一就是左上角的视图，它是一个包含各种 Java 包、类、jar 和其他文件的层次结构。这个视图称为 **Package Explorer**。还要注意主菜单已经展开了——并且出现了两个新的菜单项：Source 和 Refactor。

Java 开发环境（JDE）

为试验一下 Java 开发环境，我们将创建并运行一个“Hello, world”应用程序。使用 Java 透视图，右键单击“Hello”项目，选择 **New=>Class**，如图 2 所示。在随后出现的对话框中，键入“Hello”作为类名称。在“Which method stubs would you like to create?”下面，选中“public static void main(String[] args)”复选框，然后按 Finish。

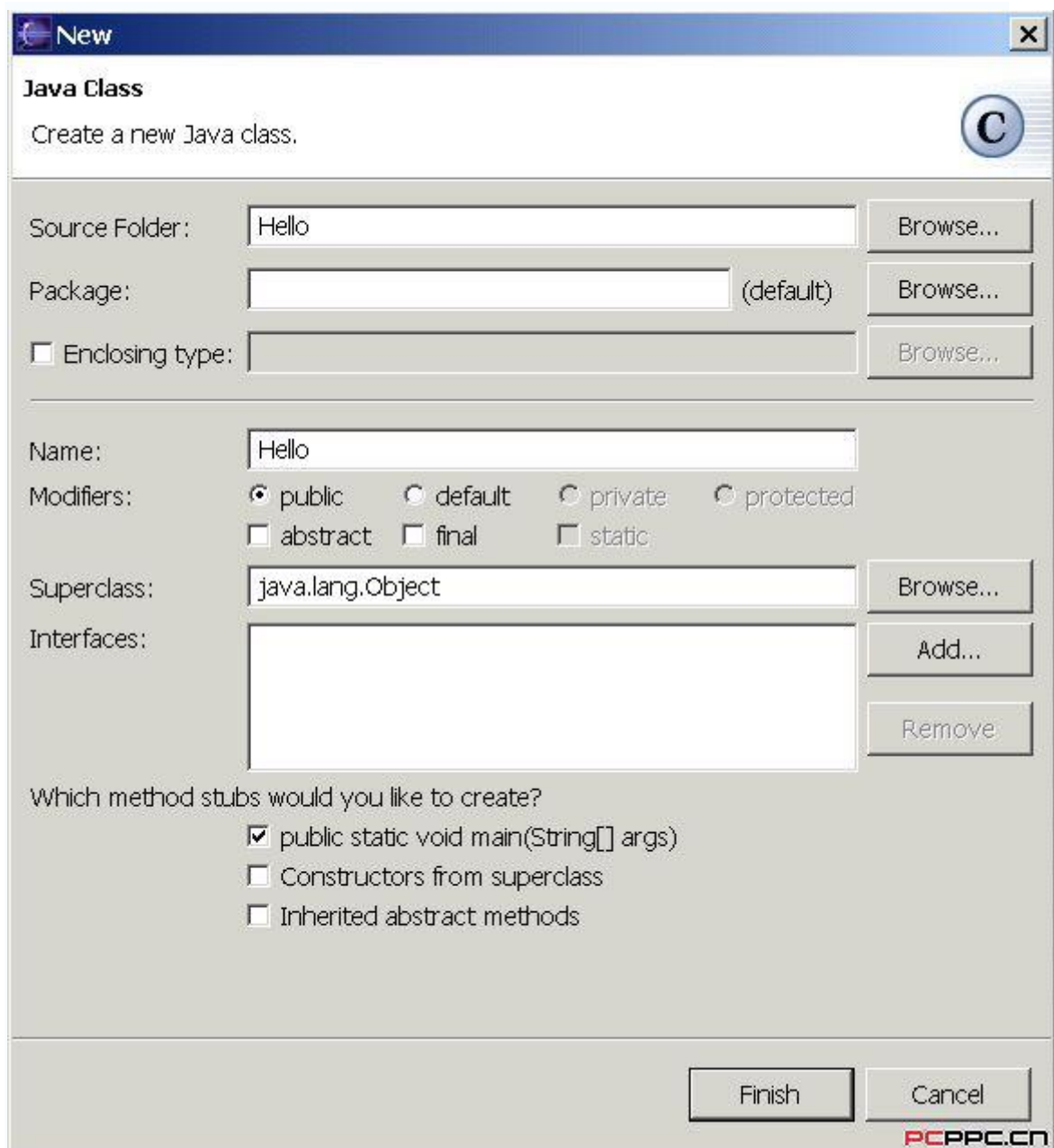
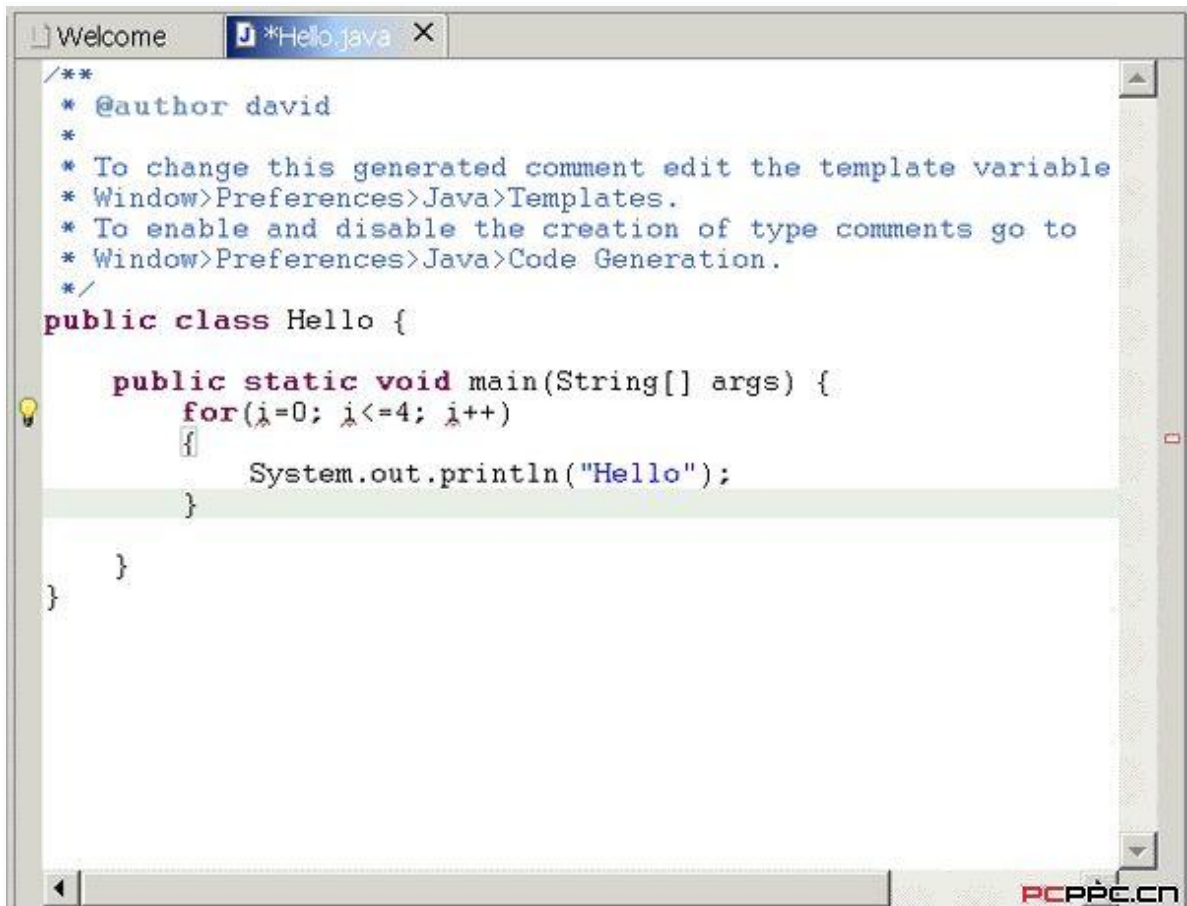


图 2. 在 Java 透视图中新建类

这样将在编辑器区域创建一个包含 Hello 类和空的 main() 方法的 .java 文件,如图 3 所示。然后向该方法添加如下代码(注意其中 i 的声明是有意省略了的):



图文

图 3. Java 编辑器中的 Hello 类

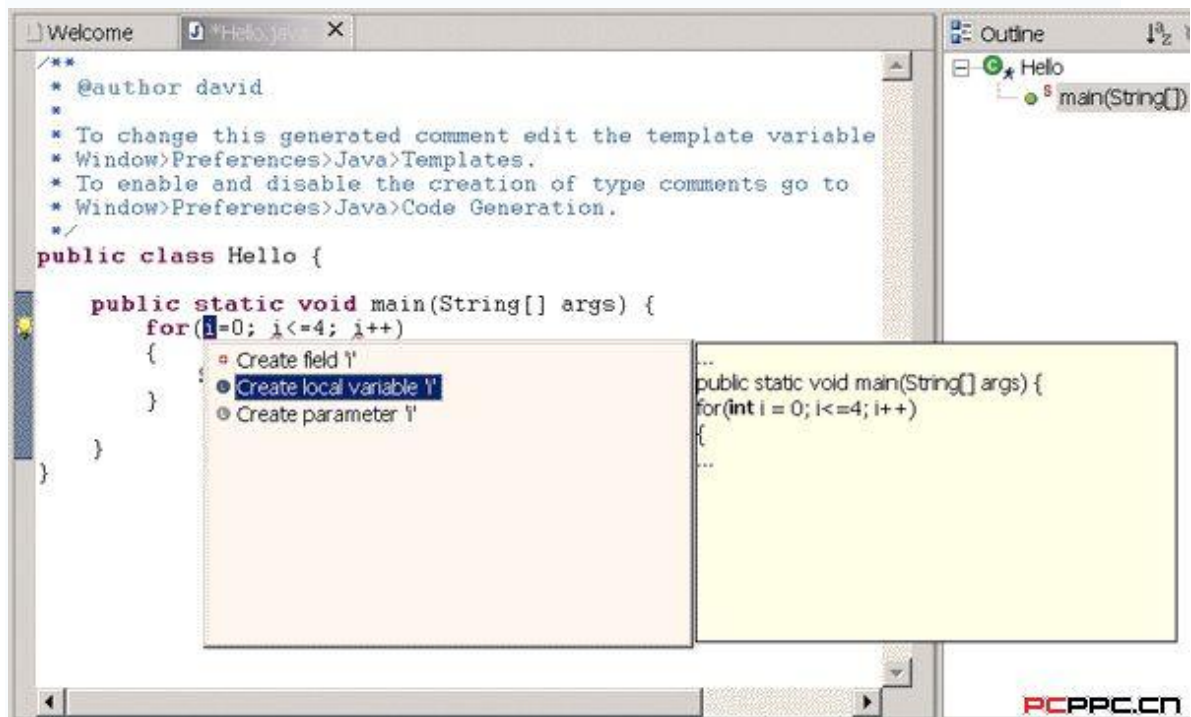
您会在键入时注意到 Eclipse 编辑器的一些特性，包括语法检查和代码自动完成。在 2.1 版（我曾下载 M2 版来试用过）中，当您键入开括号或双引号时，Eclipse 会自动提供配对的符号，并将光标置于符号对之内。

在其他情况下，您可以通过按 Ctrl-Space 来调用代码自动完成功能。代码自动完成提供了上下文敏感的建议列表，您可通过键盘或鼠标来从列表中选择。这些建议可以是针对某个特定对象的方法列表，也可以是基于不同的关键字（比如 for 或 while）来展开的代码片段。

语法检查依赖增量编译。每当您保存代码，它就在后台接受编译和语法检查。默认情况下，语法错误将以红色下划线显示，一个带白“X”的红点将出现在左边沿。其他错误在编辑器的左边沿通过灯泡状的图标来指示，这些就是编辑器或许能为您修复的问题——即所谓的 Quick Fix（快速修复）特性。

上面的代码例子在 for 语句后面有一个灯泡状图标，因为 i 的声明被省略了。双击该图标将调出建议的修复列表。在此例中，它将提供创建一个类字段 i、一个局部变量 i 或

一个方法参数 `i` 的建议；单击其中的每一个建议都会显示将要生成的代码。图 4 显示了该建议列表和建议创建一个局部变量之后生成的代码。

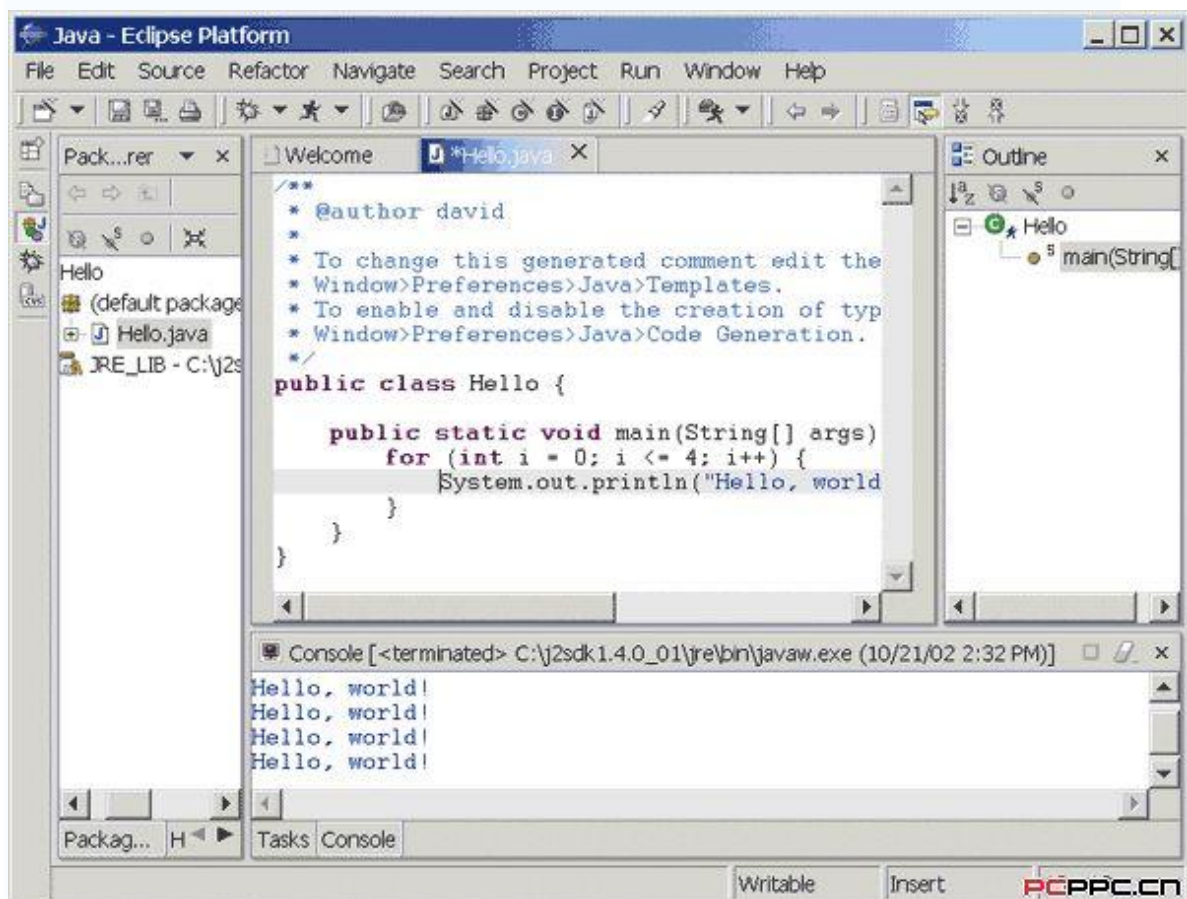


图文

图 4. Quick Fix 建议

双击该建议就会把建议代码插入到代码中的恰当位置。

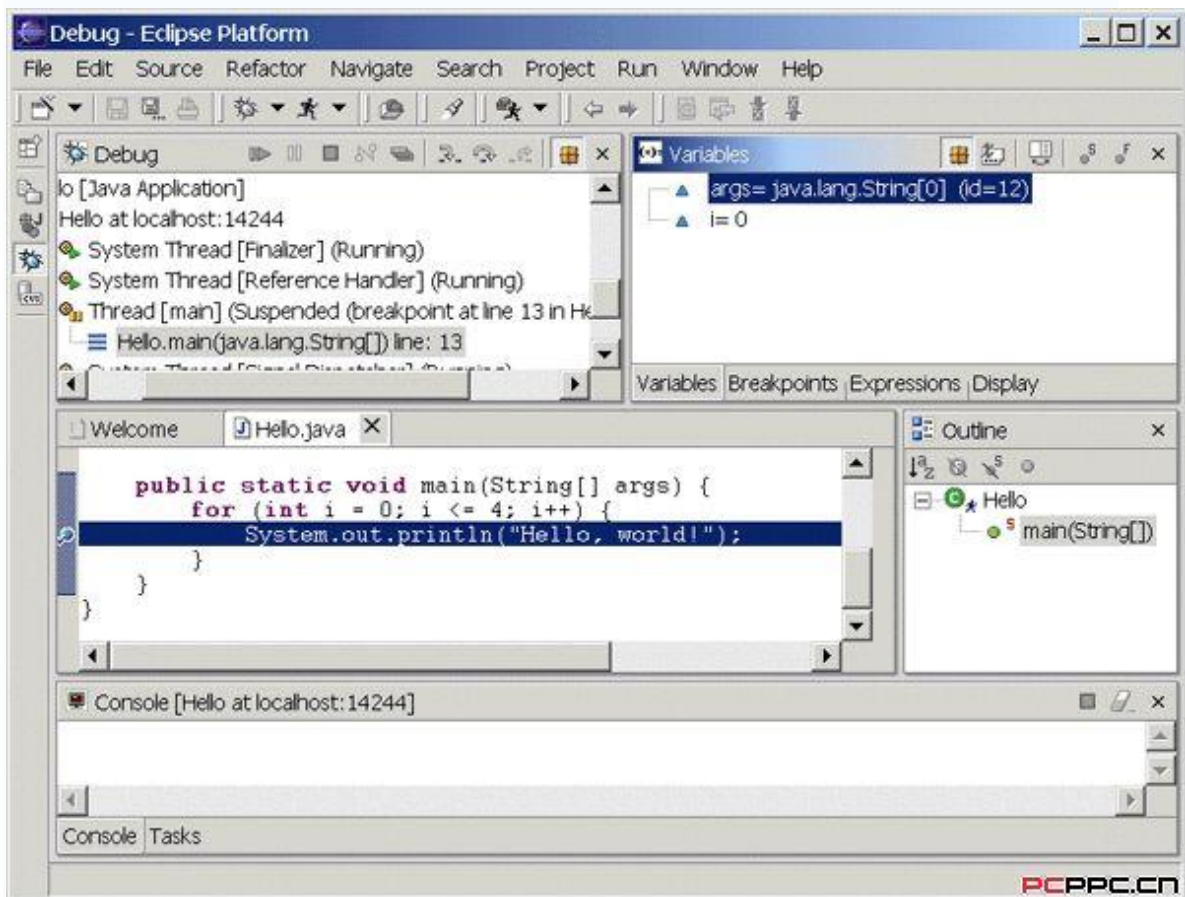
一旦代码无错误地编译完成，您就能够从 Eclipse 菜单上选择 Run 来执行该程序（注意这里不存在单独的编译步骤，因为编译是在您保存代码时进行的。如果代码没有语法错误，它就可以运行了）。这时会出现一个具有适当默认设置的 Launch Configurations 对话框；请按右上角的 Run 按钮。一个新的选项卡式窗格将出现在下面的窗格（控制台）中，其中显示了程序的输出，如图 5 所示。



图文

图 5. 程序的输出

也可以在 Java 调试器中运行程序。首先双击编辑器视图左端的灰色边沿，从而在调用 System.out.println() 之后的 main() System.out.println() 中设置一个断点。一个蓝色的点将会出现在那里。然后从 Run 菜单上选择 Debug。正如上面描述的，这时会出现一个 Launch Configurations 对话框。请选择 Run。透视图将自动切换到 Debug 透视图，其中具有许多有趣的新视图，如图 6 所示：



图文

图 6. Debug 透视图

首先，请注意该透视图左上角的 Debug 视图。这个视图显示调用堆栈，并且标题栏中有一个工具栏，它允许您控制程序的执行，包括继续、挂起或终止程序、跟踪下一个语句、单步执行下一个语句，或者从方法返回。

右上角的窗格包含许多选项卡式的视图，包括 Variables、Breakpoints、Expressions 和 Display。这里我单击了 Variables 视图，以便我们能够看到 i 的当前值。

可以通过上下文敏感的帮助，获得关于这些视图的更多信息：单击视图的标题，然后按 F1。

附加插件

除了像 JDT 这样用于编辑、编译和调试应用程序的插件外，还有些可用的插件支持从建模、生成自动化、单元测试、性能测试、版本控制到配置管理的完整开发过程。

Eclipse 标准地附带了配合 CVS 使用的插件, CVS 是用于源代码控制的开放源代码并发版本系统 (Concurrent Versions System)。Team 插件连接到 CVS 服务器, 允许开发团队的成员操作一组源代码文件, 却不会相互覆盖其他人的更改。这里不打算进一步探讨如何从 Eclipse 内部进行源代码控制, 因为这需要安装 CVS 服务器, 不过支持开发团队而不只是独立的开发, 这是 Eclipse 的一个重要的必备特性。

已经可用或已宣布要推出的一些第三方插件包括：

版本控制和配置管理

- CVS
- Merant PVCS
- Rational ClearCase

UML 建模

- OMONDO EclipseUML
- Rational XDE (代替 Rose)
- Together WebSphere Studio Edition

图形

您正在看的 JAVA 教程是: Eclipse 平台入门。

- Batik SVG
- Macromedia Flash

Web 开发、HTML、XML

- Macromedia Dreamweaver
- XMLBuddy

应用服务器集成

- Sysdeo Tomcat launcher

欲了解可用插件的更完整列表, 请参阅 Eclipse 插件列表。

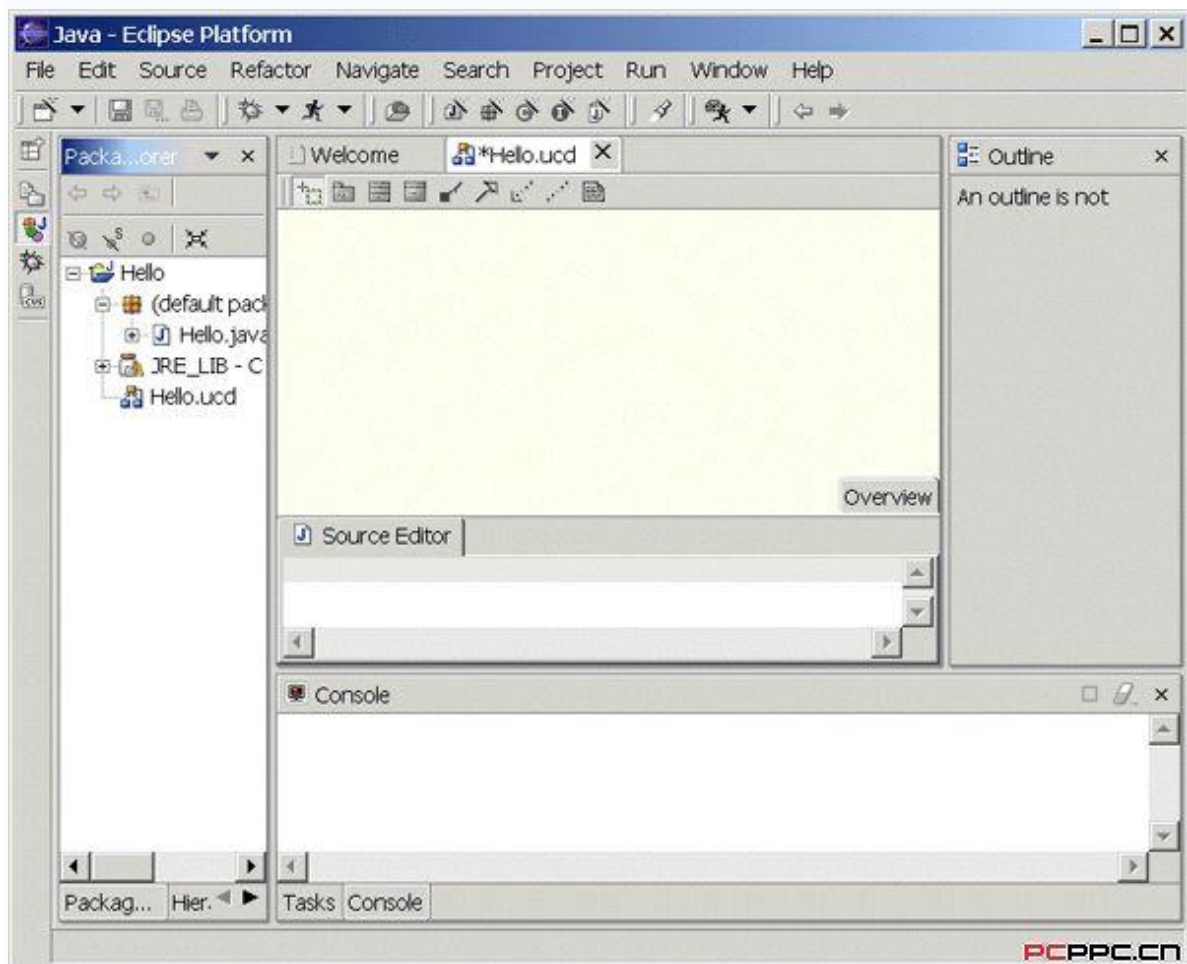
例子：一个用于 UML 建模的插件

要查看插件的例子, 以及查看它是如何与 Eclipse 集成的, 请下载流行的 OMONDO EclipseUML；您需要注册, 不过该插件是免费的。这个插件依赖 GEF, 即 Graphical Editor Framework, 这是另一个 Eclipse 插件。GEF 是 Tools 子项目的一部分。要下载 GEF, 请

转到 Eclipse Web 站点，选择“downloads”，然后单击“Tools PMC downloads page”链接。注意您需要下载 OMONDO 推荐的 GEF 版本（针对 OMONDO 1.0.2 的是 GEF 2.0 版）。

下载之后，插件的安装通常是通过解压缩下载文件，并将其内容复制到 Eclipse 插件目录来完成的。在此例中，GEF 需要解压缩到 Eclipse 目录（它将自动从该目录进入插件目录）。为安全起见，您可能想将它解压缩到某个临时目录，再相应地从那里复制相关目录。如果 Eclipse 正在运行，您需要停止它然后再重新启动它，这样它才能识别新安装的插件。

一旦 EclipseUML（以及 GEF）安装完成，您就能够像创建一个 Java 类文件一样创建一个类图。在 Java 透视图中，右键单击 Package Explorer 中的“Hello”项目，然后从弹出菜单上选择 **New=>Other**。New 对话框的左边窗格中将会有一个用于 UML 的新选项。EclipseUML 的免费版本仅支持类图，因此右侧的惟一选项是 UML Class Diagram。请选择 UML Class Diagram，然后为该类图键入一个名称，比如“Hello”：



图文

图 7. Class Diagram 编辑器

编辑器区域中将会出现一个图形编辑器，它带有用于绘制类图的画布。您可以通过两种方式创建类图：通过将 Java 文件从 Package Explorer 拖放到类图上，从而对现有代码进行逆向工程；或者使用空白类图上面工具栏中可用的绘制工具。要试验第一种方法，请创建一个名为 Person 的新类（使用 **File=>New=>Class**），然后赋予它下面列出的两个私有属性：

```
/** Person.java
 * @author david
 */
public class Person {
    private String name;
    private Address address;

    /**
     * Returns the address.
     * @return Address
     */
    public Address getAddress() {
        return address;
    }

    /**
     * Returns the name.
     * @return String
     */
    public String getName() {
        return name;
    }

    /**
     * Sets the address.
     * @param address The address to set
     */
    public void setAddress(Address address) {
        this.address = address;
    }

    /**
     * Sets the name.
     * @param name The name to set
```

```

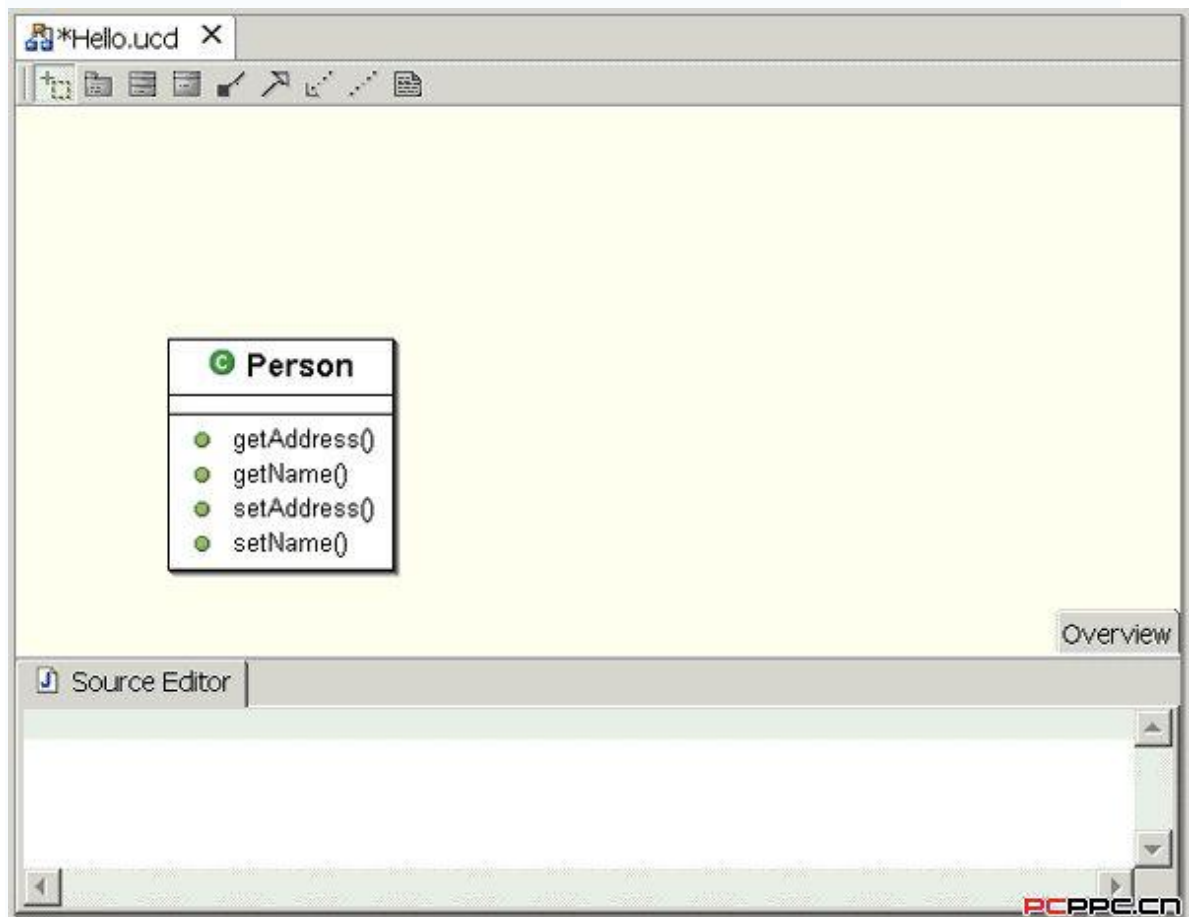
*/
public void setName(String name) {
    this.name = name;
}

}

```

（应该承认，我仅键入了针对 name 和 address 的行。getter 和 setter 方法是通过 Eclipse 自动生成的，即右键单击源代码，然后从弹出菜单上选择 **Source=>Generate Getter and Setter** 。

请保存并关闭 Person.java Hello.ucd。



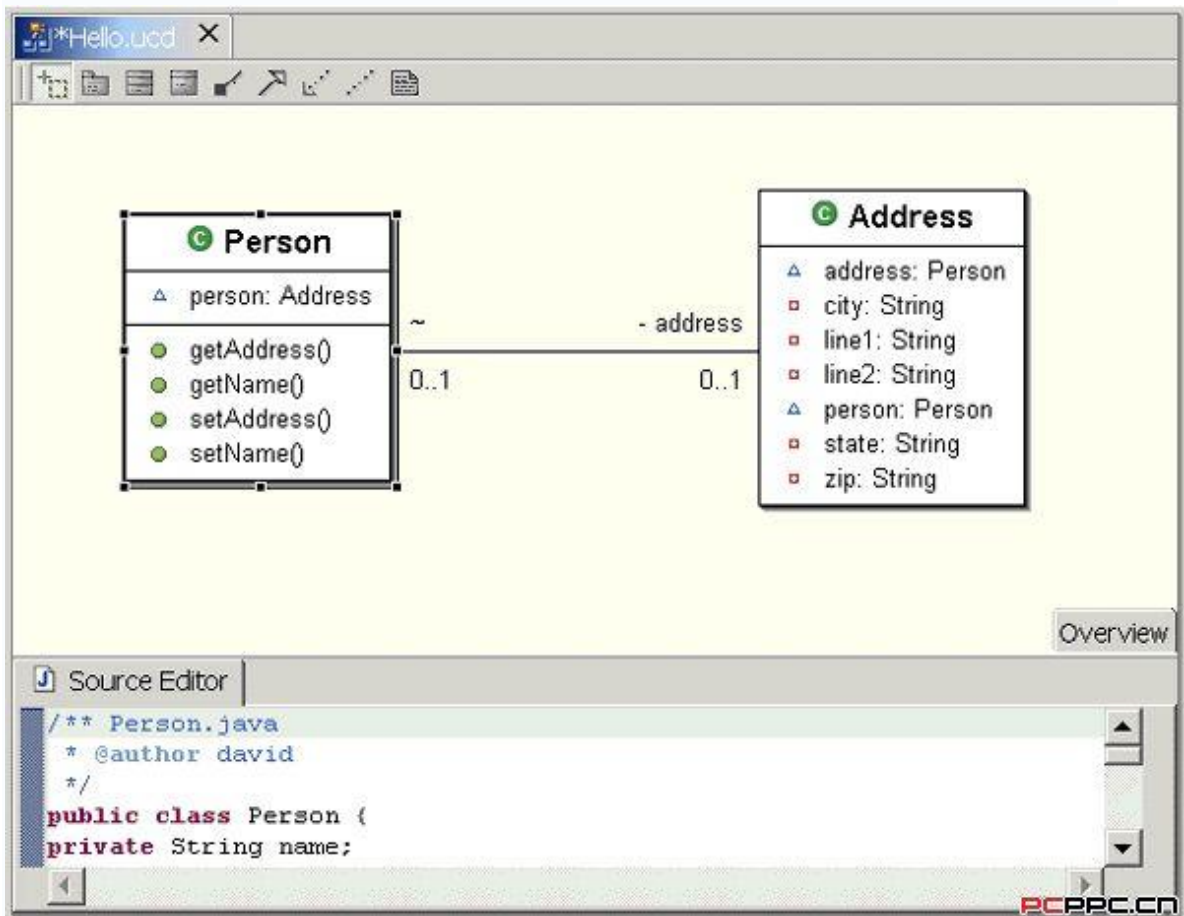
图文

图 8. Person 类图

要从 UML 创建 Java 类，请单击类图窗口顶部工具栏上的“New class”按钮，即左起第三个按钮，然后单击类图。当 New 类向导打开时，请键入 Adress 作为类名称，然后按 Finish。

您可以右键单击类名称并选择 **New=>Attribute**，从而给类添加属性。在 New 属性对话框中，请输入属性名称、类型和可见性。然后右键单击类名称并选择 **New=>Method** 来添加方法。

当您更改类图时，图下面的 Source Editor 窗口将反映所做的更改。最后，您可以单击 Association 按钮（左起第五个），绘制一条从 Person 类指向 Address 类的线段，从而绘制这两个类之间的关系图。这样会调出另外一个对话框，您可以在其中输入关联属性（请参考 EclipseUML 帮助，以了解关于必需信息的更多内容）。完成后的图应该类似如下：



图文

图 9. 关联

这个 UML 插件展示了 Eclipse 插件的几个典型特点。首先，它展示了工具之间的紧密集成。表面上绝对无法看出有多个组件在工作；与 Eclipse 平台和 JDT 的集成是无缝的。例如，当 **Person** 类被创建时，它显示语法错误是因为它的一个属性 **Address** 没有定义。一旦 **Address** 类在 UML 图中创建完成，这些组件就会分开显示出来。

另一个特点是 EclipseUML 利用其他插件提供的功能的能力——在此例中是 GEF 插件，它提供用于开发可视化编辑器的工具。

还有另一个特点涉及 EclipseUML 插件使用多层次功能来分发的方式。支持类图的基本插件是免费的，但是更成熟的版本要付费才能使用。

Eclipse 平台体系结构

Eclipse 平台是一个具有一组强大服务的框架，这些服务支持插件，比如 JDT 和插件开发环境（PDE）。它由几个主要的部分构成：平台运行库、工作区、工作台、团队支持和帮助。

您正在看的 JAVA 教程是:Eclipse 平台入门。

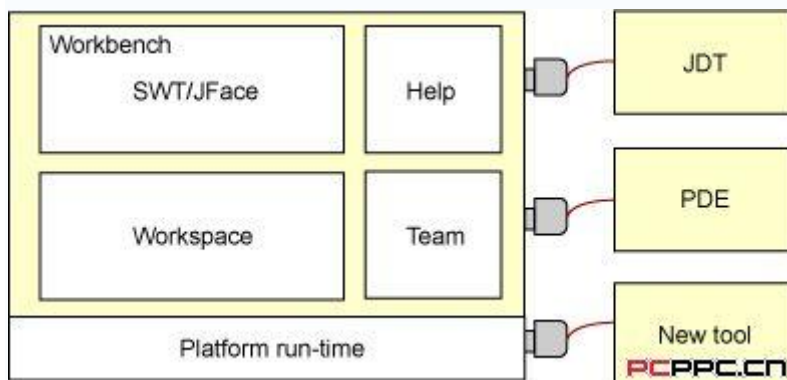


图 10. Eclipse 平台体系结构

平台

平台运行库是内核，它在启动时检查已安装了哪些插件，并创建关于它们的注册表信息。为降低启动时间和资源使用，它在实际需要任何插件时才加载该插件。除了内核外，其他每样东西都是作为插件来实现的。

工作区

工作区是负责管理用户资源的插件。这包括用户创建的项目、那些项目中的文件，以及文件变更和其他资源。工作区还负责通知其他插件关于资源变更的信息，比如文件创建、删除或更改。

工作台

工作台为 Eclipse 提供用户界面。它是使用标准窗口工具包（SWT）和一个更高级的 API（JFace）来构建的；SWT 是 Java 的 Swing/AWT GUI API 的非标准替代者，JFace 则建立在 SWT 基础上，提供用户界面组件。

SWT 已被证明是 Eclipse 最具争议的部分。SWT 比 Swing 或 SWT 更紧密地映射到底层操作系统的本机图形功能，这不仅使得 SWT 更快速，而且使得 Java 程序具有更像本

机应用程序的外观和感觉。使用这个新的 GUI API 可能会限制 Eclipse 工作台的可移植性，不过针对大多数流行操作系统的 SWT 移植版本已经可用。

Eclipse 对 SWT 的使用只会影响 Eclipse 自身的可移植性——使用 Eclipse 构建的任何 Java 应用程序都不会受到影响，除非它们使用 SWT 而不是使用 Swing/AWT。

团队支持

团队支持组件负责提供版本控制和配置管理支持。它根据需要添加视图，以允许用户与所使用的任何版本控制系统(如果有的话)交互。大多数插件都不需要与团队支持组件交互，除非它们提供版本控制服务。

帮助

帮助组件具有与 Eclipse 平台本身相当的可扩展能力。与插件向 Eclipse 添加功能相同，帮助提供一个附加的导航结构，允许工具以 HTML 文件的形式添加文档。

Eclipse 的前景

围绕 Eclipse 的开发正处于关键阶段。主要软件工具提供商都参与进来了，并且开放源代码 Eclipse 插件项目的数量正在与日俱增。

可移植、可扩展、开放源代码的框架并不是个新思想（您会想起 Emacs），但是由于它成熟、健壮和优雅的设计，Eclipse 带来了全新的动力。IBM 价值 4000 万美元的世界级软件在开放源代码领域的发布，给业界带来了久违的震撼。