

Java 编程利器 Eclipse

一、认识 Eclipse

Java 语言已渐渐成为当今程序开发的主流语言之一，但是方便、快捷、功能齐全的 Java 集成开发环境并不多见，这往往也就成为诸多 Java 初学者的绊脚石。本文将着重介绍一款由 IBM 出品的基于 Java IDE (Integrated Development Environment, 集成开发环境) 平台软件——Eclipse。

1. Eclipse 是什么

Eclipse 之于 Java 正如 Visual C++ 之于 C/C++。Java 的 IDE 软件很多，从 Borland 的 JBuilder，到 IBM 的 Visual Age for Java、WebSphere Studio，Oracle 的 JDeveloper，Sun 的 Forte for Java 等等，种类 10 种有余。在目前所有这些的 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。



图 1 Eclipse 的启动画面

为什么取名 Eclipse，我也不知道。有人说这是 IBM 故意和 Sun 对着干（一个是太阳，一个是日食），呵呵，但愿这不是真的。

2. Eclipse 有何优点

Eclipse 是一个开放源代码的软件开发项目，Eclipse 的优点来源于它的开放性和扩展性，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。

3. Eclipse 的安装

Eclipse 的最新版本为 Eclipse 3.0，笔者现在安装的是 Eclipse 2.1.1 中文版（见图 1）。以下为 Eclipse 的安装步骤：

(1) 安装之前，确保你的机器已安装了 JRE (Java Runtime Environment, Java 运行环境)。

(2) 进入 Eclipse 项目的主页 <http://Eclipse.org>，点击 Downloads，进入下载页。假设你安装 Windows 版本，如果我没有记错的话，文件名应该为 Eclipse-SDK-2.1.1-win32.zip。下载这个 zip 文件，直接解压至你指定的目录。

(3) Eclipse SDK 已经安装完毕了，如果你还需要安装中文版的话，再下载语言包：Eclipse2.1.1-SDK-win-LanguagePackFeature.zip，直接解压至 Eclipse 目录，覆盖该覆盖的文件。

(4) 至此，Eclipse 安装完毕。运行 Eclipse.exe，即可启动 Eclipse 工作台。



图 2 配置 Eclipse 的 JRE

正常情况下, Eclipse 会自动找到你的 JRE, 如果它找不到, 或者你同时安装了多个版本的 Java, 你可以通过菜单“窗口 首选项”来设置 JRE 路径, 如图 2 所示。

4. 如何得到更多的帮助

在使用 Eclipse 的过程中, 如果需要帮助, 可以通过以下途径获取帮助信息:

(1) 选择菜单“帮助 帮助内容”, 即可弹出主题帮助窗口, 如图 3 所示。



图 3 主题帮助窗口

呵呵, 是不是感觉很亲切? 全是中文的。

(2) 在任何地方按 F1 键, 即弹出联机帮助提示窗口。图 4 所示为在编辑器窗口弹出的提示窗口。

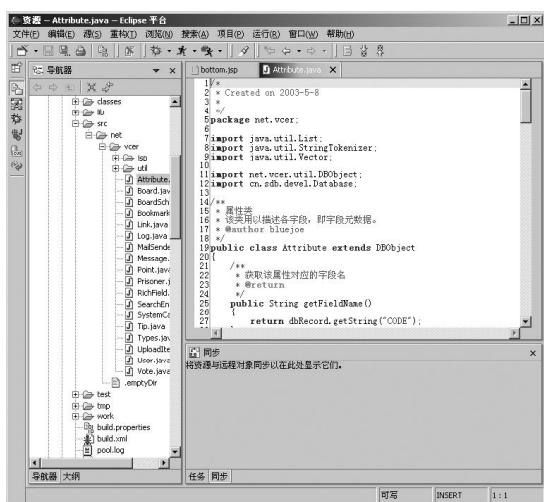


图 4 联机帮助提示窗口

(3) 你也可以浏览 Eclipse 的主页: <http://Eclipse.org>, 那里有更多的 FAQ。如果你的英文不错的话, 建议多去看看。

(4) 此外, 你还可以浏览一些国内网站, 现在使用 Eclipse 的用户越来越多了, 一般 Java 开发者论坛

里面总会提及 Eclipse。

二、认识 Eclipse 界面

1. Eclipse 界面

启动 Eclipse 后, 将显示如图 5 所示界面。

窗口由菜单栏、工具栏、工作台以及底部的状态栏组成。

通过菜单, 我们可以实现很多功能, 如前文所提及的, 我们可以通过“窗口 首选项”来配置所有 Java 项目的默认属性。

在使用过程中, 你随时可以通过点击右键弹出上下文菜单。

工作台为窗口的重要组成部分, 以下详细分解 Eclipse 工作台。

2. Eclipse 工作台

Eclipse 工作台由一个或多个透视图组成, 默认显示的是“资源”透视图。透视图包括:

- * 资源: 用于浏览文件结构;
- * 调试: 用于调试 Java 程序;
- * Java: 用于浏览 Java 的类结构;
- * Java 浏览: 用于浏览 Java 项目结构。

如此等等, 如果安装了相关的 Eclipse 插件, 透视图还会增加, 如图 6 所示为笔者安装了 Together 插件之后的 Eclipse 透视图列表。

透视图一般由视图和编辑器组成, 如图 6 所示, “资源”透视图就包括“导航器”、“大纲”以及“任务”等视图, 图右为编辑器, 它同时显示了一个 jsp 文件和一个 Java 文件的编辑界面。



图 5 Eclipse 窗口界面



图 6 Eclipse 的透视图

三、Eclipse 中 Java 程序的开发

以上内容大致介绍了 Eclipse 的界面组成, 下面开始在 Eclipse 中编写我们的第一个 Java 程序, 为了后续内容, 我们不再选择那个著名的“Hello World”, 而是实现一个简单的计数器 Counter。

在 Eclipse 里面做 Java 程序, 开发步骤如下:

- * 新建一个 Java 项目
- * 在项目中新建包和类
- * 在类中编写类代码

* 编译运行、调试代码

1. 新建 Eclipse 项目



图 7 新建 Java 项目

通过菜单或者工具栏按钮, 选择“新建项目”, 弹出如图 7 所示的窗口。

点击“下一步”, 指定项目名称, 如: counter。

在“资源”透视图里, 我们可以

找到这个 counter 项目, 在“导航器”视图里, 可以发现它创建了两个文件: .classpath 和 .project, 这两个文件描述项目的类路径信息和项目描述信息, 用户不要轻易删除 (见图 8)。

2. 新建 Counter 类

现在我们要为这个空项目引入 Counter 类, 可以考虑先准备一个包, 如: bluejoe (当然, 你也可以不使用包而直接创建类)。

切换至“Java”透视图, 我们新建一个包 (见图 9)。

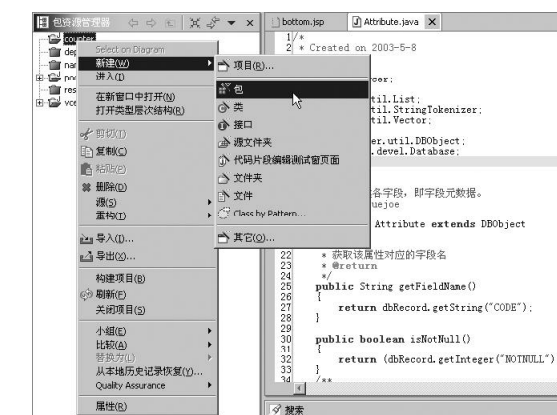


图 9 新建 Java 包

接下来, 我们指定包名为 bluejoe (见图 10)。



图 10 指定包名和路径

“完成”后, 即可在“包资源管理器”里面找到 bluejoe 包 (见图 11)。

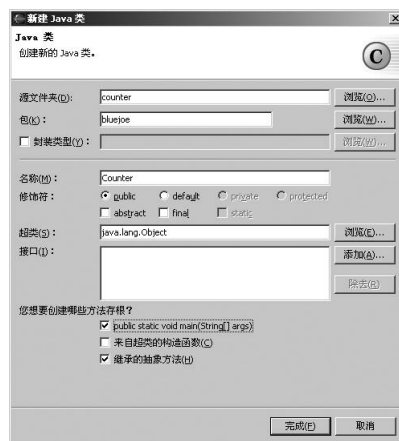


图 11 包资源管理器

灰色的图标表明这只是一个空包, 我们再加入 Counter 类 (见图 12)。

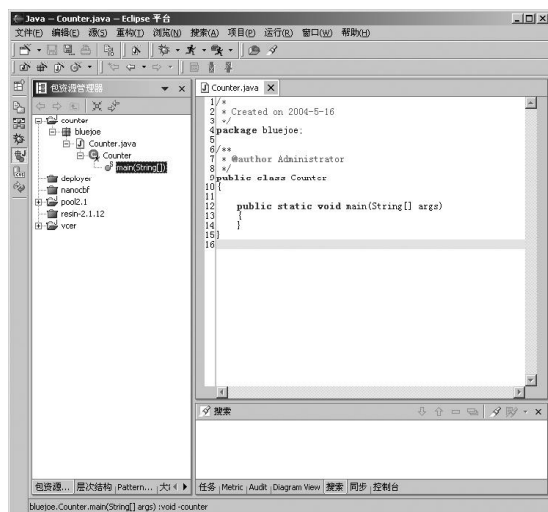


图 12 新建 Counter 类

点击“完成”, Eclipse 将自动建立一个 Counter.java, 同时我们可以在“包资源管理器”视图里看到这个类的大纲 (见图 13)。

上面我们只不过是点了点鼠标, 一个 Java 程序的大框架就写好了。

3. 编辑 Counter 类

我们再为 Counter 类填充一些代码, 填充之后的 Counter.java 内容如下:

```
public class Counter
{
    private long counter;
```



图 13 Counter 类

```

Counter()
{
    counter = 0;
}

Counter(long beginning)
{
    counter = beginning;
}

public void add()
{
    counter++;
}

public static void main(String[] args)
{
}

```

这个时候，我们在 Eclipse 里可以：

(1) 为类和方法添加注释：在“包资源浏览器”视图里选择类或者方法，选择右键菜单命令“源 添加 Javadoc 注释”，即可添加相应的 Javadoc 注释模板。

(2) 为 counter 变量添加 get 方法：选择 counter 变量，选择右键菜单命令“源 生成 Getter 和 Setter...”，将弹出如图 14 所示的窗口。

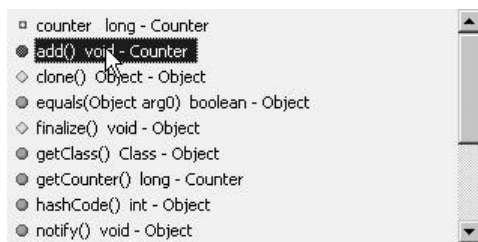


图 14 为类属性添加 Getter 和 Setter

点击“确定”，Eclipse 即自动生成了 getCounter

() 方法：

```

/**
 * @return
 */
public long getCounter()
{
    return counter;
}

```

(3) 填充 main 方法，用以测试我们的类：代码如下：

```

public static void main(String[] args)
{
}

```

```

Counter myCounter = new Counter(100);
myCounter.add();

System.out.print(myCounter.getCounter());
}

```

在代码输入过程中，即可领略到 Eclipse 的智能提示功能，如图 15 所示，编写代码的过程中，Eclipse 自动列出了 myCounter 的成员列表。

① 创建局部变量“stranger”
② 创建字段“stranger”
③ 创建参数“stranger”

图 15 Eclipse 的智能提示功能

(4) 代码格式化：有的时候，手动输入的代码并不是很好看，那么就可以使用 Eclipse 的代码格式化功能来排版你的代码，在编辑器的上下文菜单命令中可以找到“源 格式化”选项。

(5) “组织 import”和“添加 import”：如果你输入了一个类，如 File，你就可以使用 Eclipse 的“添加 import”功能，自动加入 import java.io.File 语句；不同于“添加 import”，“组织 import”则完成对所有 import 语句的批量添加。

此外，Eclipse 的编辑器还提供错误提示、快速修正等功能。例如：当你在某个方法里使用了一个未声明的变量 stranger，编辑器将报错：“不能解析 stranger”，此时点击代码行前面的灯泡图标，即可弹出如图 16 所示的修改建议。



图 16 Eclipse 的快速修正功能

此时选择“创建字段 Stranger”，即自动为类创建了一个 Stranger 属性字段。

呵呵，是不是感觉很智能啊？Eclipse 在你的编辑过程中就完成了语法检查，这样可以保证你总能写出语法正确的代码。

4. 运行 Counter 类

选择菜单“运行 运行行为 Java 应用程序”，即完成 Counter 的执行，输出信息将自动打印至控制台窗口，如图 17 所示。

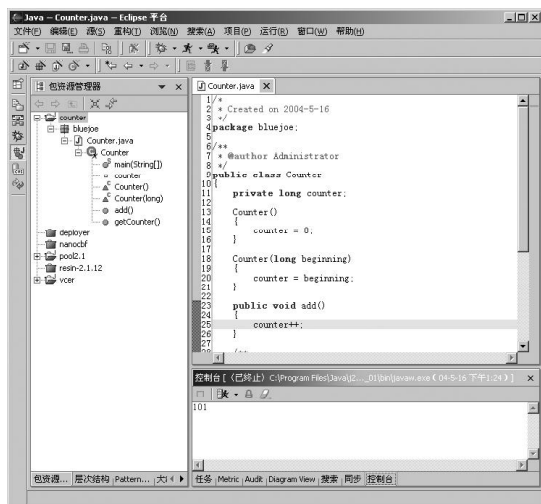


图 17 在 Eclipse 中运行程序

也可以新建一个“运行”项，以备以后使用。选择“运行 运行...”菜单项，弹出如图 18 所示的窗口。

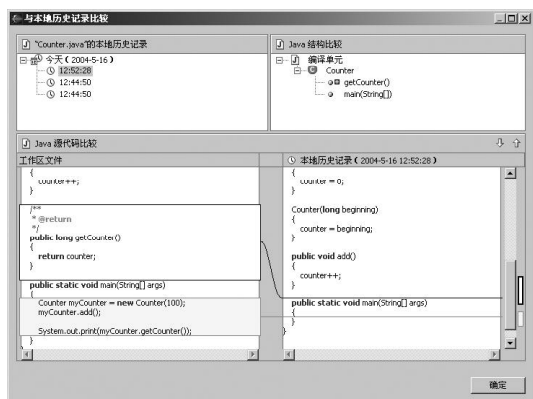


图 18 配置运行项

选择“Java 应用程序”，点击“新建”，然后指定名称和 Main 类即可。以后只要选择该运行项，即可启动该 Java 应用程序。

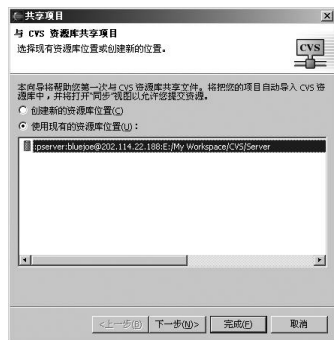


图 19 在 Eclipse 中调试程序

5. 调试 Counter 类

Eclipse 中，调试的操作与运行很相似。你可以使用断点让程序停下来，并观察其中的变量，如图 19 所示。

调试时，以下快捷键经常会使用到：

- * F5 Step Into：进入方法体；
- * F6 Step Over：执行当前语句；
- * F7 Step Out：跳出方法体；
- * F8 Resume：继续执行；
- * Ctrl+R Run to cursor：运行至当前行。

调试时，Eclipse 将自动切换至“调试”透视图，它包含以下视图：

- * 调试：显示调试进程、线程信息；
- * 变量：显示当前变量；
- * 断点：显示断点信息；
- * 表达式：允许表达式输入表达式，查看结果；
- * 显示：允许表达式输入表达式，输出结果；
- * 控制台：显示输出信息。

此外，Eclipse 还提供了远程调试功能，在此不再详述。

6. 使用本地历史记录

在 Eclipse 里面编写代码非常安全，因为它保存了不同版本的历史记录，你可以随时查看各历史版本，比较文件内容，并选择“替换”或者“恢复”操作。如图 20 所示就是历史版本比较的界面窗口。



图 20 比较本地历史纪录

四、Eclipse 高级应用

1. CVS 版本管理

现如今，大多数应用程序是由多人组成的团队开发的。即使只涉及几个开发人员的小项目，也需要对源代码的更改进行严格控制。CVS 是一种通用的版本管理系统，很幸运的是，Eclipse 内嵌了一个 CVS 客户端工具。当你编写本地代码时，很容易就可以实现和远程资源库的同步。

使用版本管理的前提是你拥有一台 CVS 服务器，这个很简单，下载一个 CVS Server，配置一下即可。接下来，你就可以在 Eclipse 中配置 Server 的位置，选择我们的 counter 项目，在上下文菜单命令中选择“小组 共享项目”，将弹出如图 21 所示的窗口。



图 21 共享项目设置

你可以选择已有的 CVS 资源库配置，也可以新建一个配置项（如图 22 所示）。

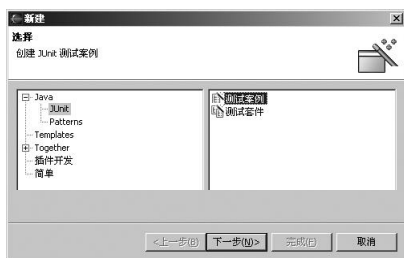


图 22 新建共享项目设置

配置完成之后，“小组”菜单项将会展开，如图 23 所示。



图 23 版本管理菜单

你可以：

- * 使用“与资源库同步...”完成与远程资源的比较；
- * 使用“更新”获取远程资源的最新版本；
- * 使用“提交...”提交你本地的修改工作。



图 24 使用 CVS 进行项目的协同开发

如此等等，和“本地历史记录功能”相似，你也可以比较本地与远程版本的文件内容，并选择“替换”或者“恢复”操作。如图 24 所示是笔者使用 CVS 和 V Cer.net 项目

2. JUnit 单元测试

单元测试对于程序设计来说也很重要，没有通过测试的代码无疑和随手抓住一把降落伞跳下去一样恐怖。

Eclipse很好地集成了JUnit 测试框架，但需要你额外下载一个JUnit 的包文件。你可以在官方网站 junit.org 找到 JUnit.jar。

下面我们使用 JUnit 来测试我们前面写出来的 Counter 类。

(1) 首先将 JUnit.jar 加入构建路径，让 Eclipse 能找到 JUnit 的包类。选择 counter 项目，右键菜单选择“属性...”弹出属性菜单，并在“Java 构建路径”里引入 JUnit.jar。

(2) 新建测试案例。选择 counter 项目，右键菜单选择“新建 其他...”，在如图 25 所示窗口中选择“JUnit 测试案例”。

单击“下一步”按钮进入下一个界面（见图 26）。

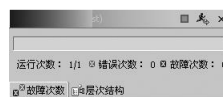


图 25 新建测试案例 Step 1

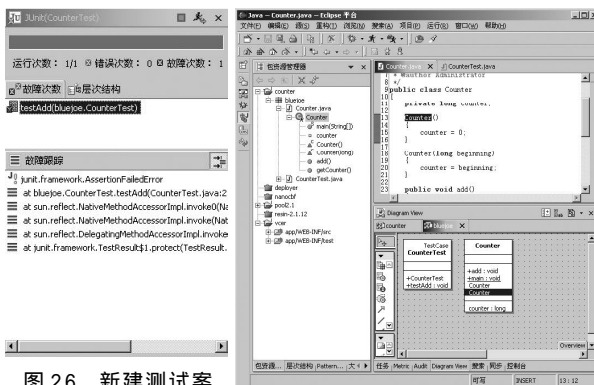


图 26 新建测试案例 Step 2

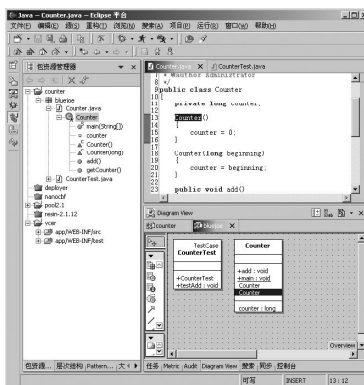


图 27 新建测试案例 Step 3

填写好各项信息，单击“下一步”，选择待测试的 Counter 的方法（见图 27）。

我们仅选择 add，点击“完成”按钮。CounterTest 就生成了，内容如下：

```
package bluejoe;

import junit.framework.TestCase;

/**
 * @author Administrator
 */
public class CounterTest extends TestCase
{
    /**
     * Constructor for CounterTest.
     * @param arg0
     */
}
```

```

public CounterTest(String arg0)
{
    super(arg0);
}

public void testAdd()
{
}

```

(3) 我们现在的任务就是在 testAdd 里面加入我们具体的测试代码, 如下所示:

```

public void testAdd()
{
    Counter myCounter = new Counter(100);
    myCounter.add();

    assertTrue(myCounter.getCounter() == 101);
}

```

这段代码的意思是说: 我创建一个计数器, 起始值为 100, 现在计数一次, 现在我断言它的计数应该是 101。这是个非常简单的测试逻辑, 但不要小看它, 实际上它和“我现在发射神州五号, 我断言它在多少秒以后的速度应该是多少 km/s”是一样的逻辑。

assertTrue 是 TestCase 的方法, 同样的方法还很多, 参见 JUnit 详细手册。

(4) 接下来我们开始“点火发射”, 来测试这个 Counter。选择菜单“运行 运行” (注意, 是运行“JUnit 测试”), 我们就可以发现“JUnit 测试”视图 (见图 28)。

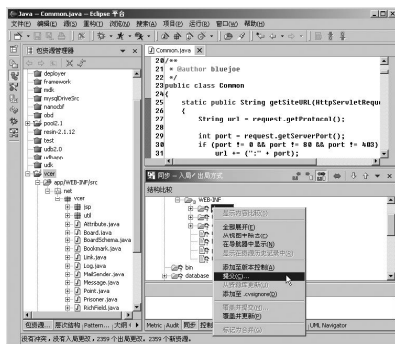


图 28 JUnit 运行结果

绿色告诉我们, 我们的 Counter 工作正常。

现在我们改一下代码, 假设粗心的你将 Counter.add() 写成:

```

public void add()
{
    counter - - ;
}

```

这时候运行 CounterTest 案例。

很显然, 这个结果告诉我们, Counter 类出问题了。你下一步的工作就是在 add() 方法里面查找故障。

JUnit 的优点不再展开, Eclipse 集成了 JUnit 框架, 无疑会为你选择 Eclipse 增加筹码。

3. Eclipse 插件

Eclipse 的一个很突出的特点就是它可以集成更多的插件(plugin), Eclipse 本身也提供了开发插件的功能。一般安装 Eclipse 插件的步骤很简单, 将插件解压至 Eclipse 的 plugins 目录下即可, 下一次 Eclipse 启动时会自动装载这些插件。

以下简单介绍几款有用的插件:

(1) Together for Eclipse

Borland Together 是著名的 UML 建模工具, 它可以完成代码和模块之间的及时同步。Borland Together 除了为 Microsoft .Net 提供了插件之外, 也提供了 Eclipse 的插件, 安装了 Together for Eclipse, 用户完全可以在 Java 项目里面完成各种 UML 建模。

下载地址: http://borland.com/products/downloads/download_together.html。

(2) Lombok j2ee plugin

严格的说 Lombok 是 Eclipse 的一个 J2EE 的插件, 它将很多 Java 应用服务器、J2EE 组件和 Web 应用开发集成到 Eclipse 中, 可以帮助 Java 开发者使用 Eclipse 建立、测试、部署 J2EE 应用。对于 JSP 的调试, Lombok 的原理则是对要调试的 JSP 页面产生的 Java 代码进行调试。整个调试过程与 Java 应用程序的调试过程一样, 这样不论是 JSP 本身的调试还是被 JSP 调用的对象的调试便无缝地联结到了一块。Lombok 插件会让你早日结束使用 UltraEdit 编辑器编写 JSP 代码的恶梦。

下载地址: <http://www.objectlearn.com/products/download.jsp>。

(3) XMLBuddy

XMLBuddy 为 Eclipse 增添了 XML 编辑能力, 其中包括对用户可配置的语法着色、DTD 驱动的代码辅助、验证以及同步的提纲视图。XML Buddy 还为 Workspace 添加 XML 透视图, 为 XML 文档和 DTD 添加新的项目模板。

下载地址: <http://www.xmlbuddy.com/>。

(4) VSS 插件

Eclipse 本身已经提供了对著名的开放源码的版本控制服务器 CVS 的支持, 但是有的用户还是习惯于 Visual SourceSafe (VSS) 的 Check Out-Check In, 你可以下载 VSS 插件, 即可在 Eclipse 实现 SourceSafe 的版本管理。

下载地址: <http://vssplugin.sourceforge.net/>。

正是如此多的插件丰富了 Eclipse, 让你永远不感到寂寞。也正是 Eclipse 对诸多插件的支持, 坚定了你选择 Eclipse 的信念。

Eclipse 入门教程到此结束, 希望你能快速使用上 Eclipse 并喜欢上它。一个好的 IDE 对于学习一门程序语言是非常重要的, 笔者也希望你能熟练掌握 Java 语言, 把代码写得更好!

(文 / 白乔)