

Firefly 与 WSAD 集成的优点

Firefly 与 IBM WebSphere Studio Application Developer (WSAD) 的集成使 WSAD 用户在 IDE 集成开发环境中也可以使用 Firefly 软件配置管理工具的强大功能。本文主要介绍集成所带来的强大功能以及实现集成的一些相关底层技术，同时也将讲述一些使用上的技巧，以便帮助用户更好地解决实际问题。

关键字

Hansky

Firefly

Software Configuration Management

Version Control

WebSphere Studio Application Developer/WSAD

Eclipse

Firefly 简介

Firefly 是 Hansky 公司开发的新一代软件配置管理工具，是 Hansky 公司软件开发管理系列产品的重要组成部分。它具有安全可靠、性能卓越等特点。Firefly 不仅能够提供包括检入/检出等版本控制功能在内的一些基本操作，还可以提供各种高级的软件配置管理特性，比如：基于项目的分支管理、标记管理、并行开发支持以及基于任务的软件配置管理等。利用这些强大的功能，Firefly 可以帮助您提高工作效率、提升团队协作程度以及提高软件质量。Firefly 还具有很好的可扩展性，它可以支持从 3~5 人的小规模开发团队到企业级应用规模的所有项目，并且通过使用 ServerSync 模块还可以支持物理上分布的开发团队在一个项目上进行异地开发。

Firefly 采用了目前最先进的技术，例如：Java 和 .NET 技术。由于使用了 Java 作为主要开发语言，使得 Firefly 能够和大多数常用的操作系统兼容，如 Windows 2000、Windows XP、Windows 2003、Solaris、AIX、HP-UX、Linux、AS400、SCO Unixware 等等。Firefly 不但提供图形化的用户界面、基于命令行方式的文字界面以及基于 WEB 的界面，还能够和大多数主流的 IDE 工具集成，如 IBM WSAD /Eclipse、Microsoft Visual Studio 以及 Borland JBuilder 等。另外，Firefly 的 Server/Client 的架构还采用了 Java Remoting 技术（一种快速的真正基于 TCP/IP 的协议，无需服务器共享任何目录）。

通过 Firefly 系统和 Butterfly（变更管理）系统的集成，您可以得到一个管理整个软件开发应用生命周期的完整的和统一的软件配置管理解决方案。

Eclipse 和 WSAD

Eclipse 是一个开放的、可扩展的平台，可以用于开发诸如：网站、嵌入式 Java 应用以及 C++ 程序等多种应用的平台。WSAD 扩展了 Eclipse 的功能，使其可以支持 J2EE 等大型企业级应用开发。WSAD 提供统一的开发环境，支持开发从 WEB 应用、商业应用程序、系统集成到传统应用的多种开发模式。使用 WSAD 或 Eclipse，一个团队的所有开发人员不但可以在一个统一的环境中个性化自己的开发需求，并且可以很容易的进行协同开发。WSAD 和 Eclipse 提供的对第三方集成机制的支持是 Firefly 和 WSAD 集成的基础。

深入解析 Firefly 和 WSAD 的集成

Firefly 和 WSAD 的集成使您可以在熟悉的 IDE 开发环境中使用 Firefly 提供的强大功能。通过一个统一的用户界面就能够满足您所有的版本控制的需求，并且它的安装也如同一次复制/粘贴操作一样简单，无需多余的步骤。您在本地工作区内所做的所有变化都将自动提交到服务器上，便于和团队中的其他成员进行协同工作。Firefly 与 Eclipse/WSAD 的集成环境中的用户界面和 Firefly 客户端中的用户界面完全相同，因此熟悉 Firefly 客户端的用户无需额外的学习就可以很快的上手。对于那些只有一些入门级别软件配置管理知识的使用者来说，使用 Firefly 与 WSAD 的集成环境无疑是一个很好的选择。总之，对于一个开发团队的所有角色来说，Firefly 和 WSAD 的集成环境不但功能强大而且方便使用。

系统需求：

- WSAD 5.0/ 5.1.x
- Eclipse 2.0/ 2.1.x
- Firefly Server 2.0 或以上版本

安装：

- WSAD：将 com.hansky.hscc 目录及其目录下的所有文件拷贝到 ./eclipse/plugins 目录下
- Eclipse：将 com.hansky.hscc 目录及其目录下的所有文件拷贝到 ./plugins 目录下

什么是项目

项目的概念在 WSAD 系统和 Firefly 系统中是不一样的。在 Firefly 系统中，项目是保存软件

项目整个应用生命周期所有信息和资源的逻辑单元，这个项目可以只包含一个单一的根本分支，也可以在包含根本分支的同时包含许多子分支。

而在 WSAD 系统中，项目是一个组织目录、包以及文件的信息组织单元，用于建立程序、控制程序的版本、共享项目以及在工作区内组织项目资源。

WSAD 系统中的项目对应于 Firefly 系统中的一个分支或分支下的一个目录，并且 Firefly 系统中的一个本地工作区内可以包含多个 WSAD 项目。在本文的剩余部分，除非明确指出，项目是专指 WSAD 系统中的项目。

通过 WSAD 和 Eclipse 访问 Firefly

在 WSAD 和 Eclipse 中，Firefly 的功能通过 WSAD 菜单、工具栏以及弹出式菜单的方式提供给用户（图 1）。Firefly 的下拉式菜单提供了配置管理的所有命令。Firefly 工具栏还提供了方便的单击操作，通过单击图标按钮您就可以进行一些常用的 Firefly 操作。弹出式菜单提供了基于所选文件/目录的所有主要操作。这种集成不但使用户可以像使用 IDE 开发工具一样和 Firefly 系统进行交互，并且使 WSAD 和 Eclipse 开发者可以在常用的开发模式下高效地进行版本控制和协同开发。

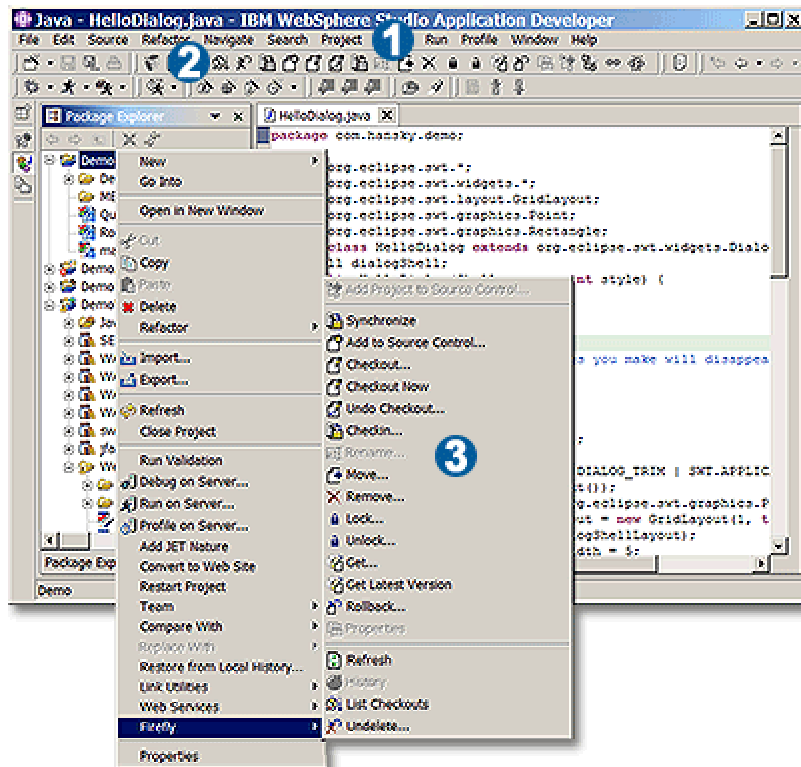


图 1 Firefly 中的操作

1. Firefly 下拉式菜单
2. Firefly 工具栏

3. Firefly 弹出式菜单

表 1 WSAD 中常用的 Firefly 命令

Menu options	Context menu options	Toolbar icons
Config Firefly	Synchronize	Config Firefly
Add Project to Source Control	Add to Source Control	Refresh
Open Project from Source Control	Checkout ...	History
Bind Project with Source Control	Checkout Now	List Checkouts
Unbind Project from Source Control	Undo Checkout	Undelete
Synchronize	Checkin	Synchronize
Add to Source Control	Rename	Add to Source Control
Checkout ...	Move	Checkout Now
Checkout Now	Remove	Undo Checkout
Undo Checkout	Lock	Checkin
Checkin	Unlock	Rename
Rename	Get	Move
Move	Get Latest Version	Remove
Remove	Rollback	Lock
Lock	Properties	Unlock
Unlock	Refresh	Get Latest Version
Get	History	Rollback
Get Latest Version	List Checkouts	Properties
Rollback	Undelete	Add Project to Source Control
Properties		Open Project from Source Control
Refresh		Bind Project with Source Control
History		Unbind Project from Source Control
List Checkouts		
Undelete		

实现基础

Firefly 系统与 WSAD 工具的集成采用纯 Java 技术实现，其工作方式就好像在您熟悉的 IDE 开发环境中一样。虽然我们推荐安装 Firefly 客户端，但采用 Firefly 与 WSAD 集成的方式，您不需要独立的客户端就可以完成软件配置管理的所有操作。当使用集成环境时，所有的操作都将在与 IDE 开发环境相同的 Java 运行环境中进行。因此不需要调用其他别的应用或独立的进程，您就可以在 WSAD 环境中完成所有关于软件配置管理的操作。集成的好处也显而易见：首先是快速，您无需等待开启一个新的进程而后等待相关数据的传输；其次是高效，也就是有效的节省了系统资源；最后也是最重要的，该方式可以更紧密的与 IDE 开发工具进行交互。

另外，Firefly 还扩展了 team、actionSet、popupMenus 以及其他底层的 extension point，实现了 Firefly 服务器和客户端通信的所有协议。这保证了 Firefly 和 WSAD 的集成不但继承了 Firefly 客户端的所有功能，而且更重要的是可以使其能够更加通畅的与 IDE 工具进行交互。

Java Remoting 技术

Firefly 服务器与客户端以及集成环境的通信协议均采用 Java Remoting 技术实现。它基于成熟的 TCP/IP 协议，完全由 Hansky 公司自行研发和优化。这是一个完全面向对象、基于事务、轻量级和快速的协议，其独一无二的优点如下：

安全

Firefly 服务器和客户端的通信完全基于 TCP/IP 协议，其实现不依赖任何网络文件系统也无需服务器端共享文件目录。服务器唯一要做的就是建立 socket 服务端用以监听客户端的请求。这样有两个优点：一是防止对共享文件系统的恶意拷贝和更改；二是防止病毒的攻击。

另外，采用 Java Remoting 技术可以使您能够建立 SSL 安全信道在 internet 上进行数据安全传输。

高效

因为采用了轻量的对象序列化协议,使得 Java Remoting 成为一种轻量级的通信协议，使用 Java Romating 技术比使用 SUN Java RMI 等重量级序列化协议更加节省网络负载，因此可以高效地利用网络资源。

基于流的技术使 Firefly 比同类的没有采用流技术的产品要高效的多。它使服务器和客户端的通信所占的通信带宽降低到了最小，特别是在服务器和客户端传输大文件时 Firefly 要比其他同类产品快的多。

基于项目的操作

基于项目的操作方式使共享项目资源比以往要容易的多。“Add Project to Source Control”操作可以将整个项目加载到版本控制当中。一旦一个项目纳入到 Firefly 版本控制当中，其他用户可以通过“Open Project from Source Control”命令从 Firefly 服务器中获取这个项目。您可以在本地工作区内生成这个项目的的一个拷贝，这种机制使一个团队内的所有成员可以共享一个项目的所有资源和属性。

如果您通过 Firefly 图形客户端或者命令行方式已经初始化了本地工作区，那么通过将项目绑定到源代码控制，您就可以直接针对该项目中的文件进行版本控制操作。简而言之，该操作就是通知 WSAD “这个项目是通过 Firefly 进行版本控制的”。这个操作也可以用于从服务器上打开一个 J2EE 项目，这个使用方法我们将在本文的稍后部分进行讨论。

“Unbind from Source Control” 可以让您选择是否将您的项目从 Firefly 版本控制中释放出来。当一个项目被释放出来后，在本地的项目拷贝依然存在，只是和服务器的链接已经被删除。但是您可以再次将项目加载到版本控制中，或者通过 Firefly 客户端访问已经存在的本地工作区。

自动映射

无需执行 Putback 或 Bringover 操作，在 WSAD/Eclipse 中，针对 Firefly 的任何操作都将立即反映到服务器上，Firefly 与 WSAD 的集成采用的这种模式比 Firefly 客户端采用的变更集的模式更直接和易于掌握。如果采用 Firefly 客户端，所有在本地工作区内所作的改动都将首先在变更集内进行排队，只有当您进行了 Putback 操作，这些排队的变更才会被提交到服务器上，这种模式需要额外的操作步骤但更利于组织和管理本地工作任务。

复查

Firefly 会对 “Delete”，“Refactor->Move” 和 “Refactor->Rename” 三个操作进行监听。用户无论在何时进行这些操作，IDE 工具都将调用 Firefly 命令来进行相应的处理（图 2）。

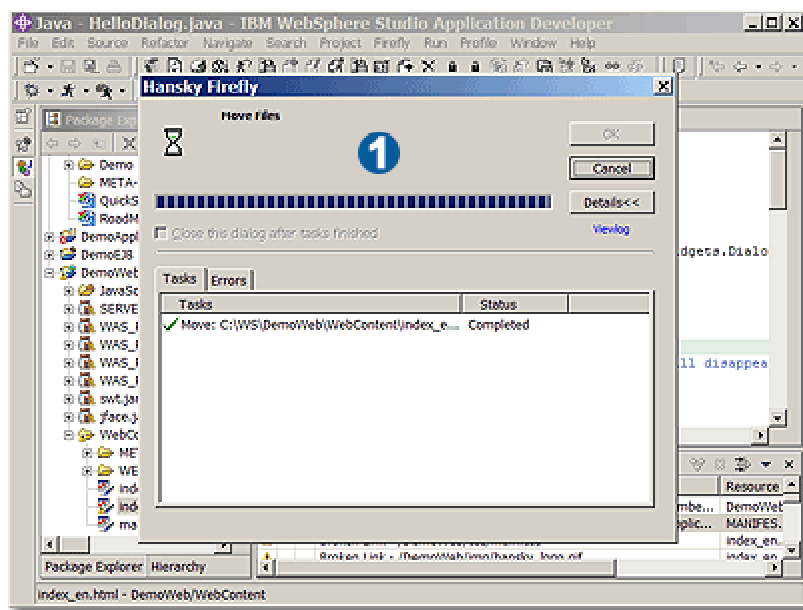


图 2 回调：当在 WSAD 中针对某一文件进行了移动操作后，相应的 Firefly 操作就会被调用，从而将相应的变化自动提交到 Firefly 服务器上。

基于目录的操作

Firefly 与 WSAD 的集成不像其他产品，您可以方便地选择在一个或几个目录上进行相应的操作，比如：“Add to Source Control”，这样可以使您在将文件加入到受控环境时无需在一

个目录下进行大量的文件选择工作，而只需右键点击相应的目录然后执行“Add to Source Control”操作就可以了（图 3）。

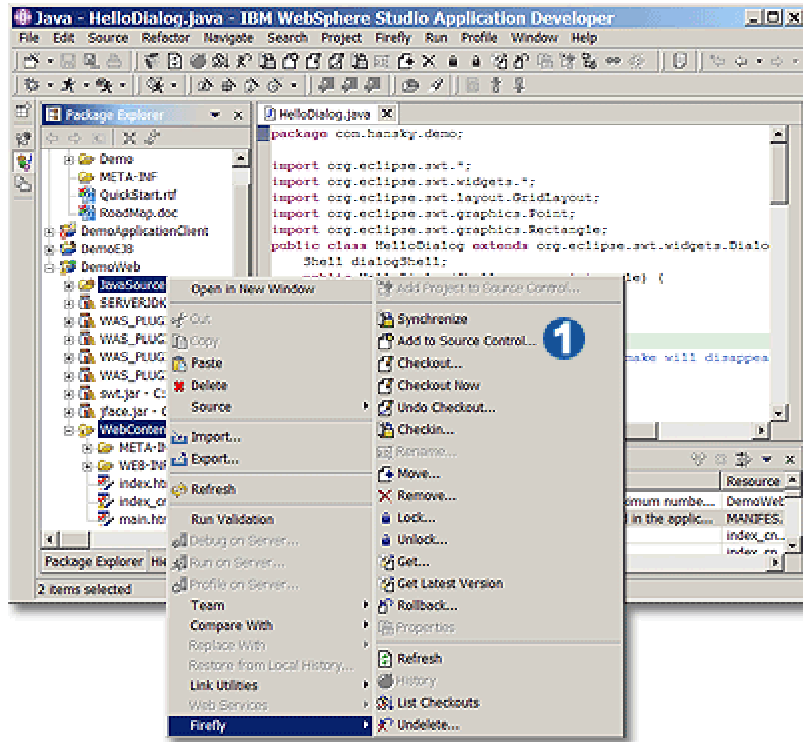


图 3 基于目录的操作：弹出式菜单列出了对于选中的两个目录的所有可以执行的操作，包括“Add to Source Control”。

全面的命令集

Firefly 与 WSAD 集成环境中的命令集既全面又简单易懂，使进行配置管理比以往要方便得多。比如在进行检出操作时，您可以选择是否将检出的文件进行锁定，而且您还可以对以前的版本或标记版本进行检出。当您点击“确定”后，Firefly 将显示过程对话框，而不是像其他产品一样显示 WSAD 缺省的进度条对话框。通过该对话框，您可以查看您正在提交的所有任务、目前的状态以及错误信息。当然如果您不想查看这个过程，您也可以点击“立即检出”按钮，这个操作将缺省检出最新版本。

“同步”可以使您得到文件的最新版本或者由其他人新近添加的文件；“回滚”可以使您得到文件的历史版本；“下载最新版本”可以使您跳过选择条件而直接得到文件的最新版本。还有许多其他的和易于使用的操作，您可以通过 Firefly 和 WSAD 集成环境的在线帮助得到这些命令的详细信息。

对于移动/重命名的支持

在 Firefly 系统中，移动和重命名操作将不被简单地看作删除和新建操作的组合，从而避免了历史版本的丢失。Firefly 集成环境将完整地保存一个文件的所有历史即使这个文件被移到了其他目录（移动操作本身也会作为文件的一个历史版本被保留下来），这样将使跟踪文件历史变得更加容易，不仅如此，Firefly 还支持对非源码文件的移动和重命名（图 4）。

Revision	Path	Date	User	Action	Description
Editing	DemoWeb/WebContent/index_cn.html	04-05-25 11:20	admin	Editing	Editing
1.6	DemoWeb/WebContent/index_cn.html	04-05-25 11:20	admin	Move	
1.5	DemoWeb/WebContent/index_cn.html	04-05-25 11:20	admin	Move	
1.4	DemoWeb/WebContent/index_cn.html	04-05-25 11:18	admin	Move	
1.3	DemoWeb/WebContent/index_en.html	04-05-19 15:05	admin	Move	
1.2	DemoWeb/index_en.html	04-05-19 15:05	admin	Move	
1.1	DemoWeb/index.html	04-05-19 15:05	admin	Update	
1.0	DemoWeb/index.html	04-05-19 14:45	admin	Create	

图 4 对移动和重命名的支持：文件的移动和重命名也将被看作文件历史的一部分而保存下来。这样即便文件被重命名您也可以查看到该文件的整个历史。

注意事项

- Unix 相关操作系统允许文件名包括一些特殊字符，但 Windows 操作系统不允许。因此如果您同时使用这两个开发环境，请注意文件的命名。
- 在 Unix 相关操作系统中一行以“/n”结束同时另起新的一行，然而在 Windows 系统中却是以“\r\n”表示，因此当您打开在 Unix 相关系统上建立的文件时请选用适合与 Unix 格式的编辑器。
- J2EE 项目：

如果在一个 J2EE 应用中有多于一个项目，那么下面的技巧将帮助您更好的管理他们。

- (1) 用户 A 在目录“C:\J2EE”下建立了一个名为“Demo”的 J2EE 项目，缺省的情况下 WSAD 会建立一个新目录“C:\J2EE\Demo”并将“.Project”文件放在该目录下。并且 WSAD 会根据您的要求建立三个标准组件项目：DemoClient, DemoEJB 和 DemoWeb，其相应的目录为：C:\J2EE\DemoClient, C:\J2EE\DemoEJB, C:\J2EE\DemoWeb。
 - (2) 用户 A 选择了一个项目 A，然后选择“Firefly -> Add Project to Source Control”，当在初始化过程中选择本地路径时，用户 A 应该选择目录“C:\J2EE”为根目录，而不是缺省的“C:\J2EE\Demo”目录。
 - (3) 用户 A 在将 WSAD 中的标准组件项目添加到版本受控环境中时，所有的和这个项目相关的四个标准组件项目都将被添加到版本受控环境中。
 - (4) 用户 B 选择“Firefly -> Open from Source Control”从受控环境中打开一个项目并在目录“C:\WS”下建立本地工作区时，所有的和这个项目相关的四个标准组件项目都会被下载到用户 B 的本地磁盘上，虽然这时只显示“Demo”一个项目。
 - (5) 用户 B 导入其余的三个项目分别到：“C:\WS\DemoClient”，“C:\WS\DemoEJB”和“C:\WS\DemoWeb”目录下。
 - (6) 用户 B 将三个标准组件项目绑定到版本控制环境中以后，就可以用 Firefly 管理这四个项目了。
- 如果 Firefly 的下拉菜单无法显示，请做如下操作：在菜单栏上选择“Window -> Customize Perspective”，然后在对话框内展开其他的窗口，选择“Source Code Control”，然后点击“确定”按钮。

- 缺省情况下，WSAD 将编译项目下的所有做过修改的原文件，同样在您从 Firefly 系统中打开一个项目也是如此。因此，如果在您最后一次打开这个项目后修改了一定数量的文件，WSAD 将会花费比较长的时间来编译这些文件，但如果您想避免这种情况的发生可以做如下操作：选择“Window -> Preference”，然后取消对自动 Build 修改资源的选择。
- 请确认当您将一个项目添加到受控环境中时选择了“.Project”文件。这样可以保证当其他人从受控环境中打开这个项目时可以共享项目的资源和属性。
- 建立一个新的项目时，请为您的源文件建立一个“Src”子目录，并同时指定 WSAD 在另外一个目录下生成.class 文件。这样可以帮助 Firefly 系统避免将*.class 文件也加入到受控环境中。这样做的好处是您当然不希望*.class 从受控环境中检出后就变成只读的了，当然您也可以将*.class 文件添加到文件过滤器中以实现这种目的。

问题和限制

- Firefly 不处理空目录，因此当您删除了一个目录下的所有文件，这个目录也会自动消失。
- 当冲突产生又不能自动解决时，您需要 VDM（可视化的版本比较和归并工具），因此如果您打算使用 VDM 工具，您必须安装 Firefly 客户端。

更多信息

如果想得到关于 Hansky Firefly 工具以及其他软件开发管理工具的信息，请访问：

Hansky.com/en/products/index.html

如果想得到关于使用 Firefly 与 WSAD 集成环境的详细信息，请参考 WebSphere Studio Application Developer 的帮助。

关于 WebSphere Studio Application Developer 的详细信息，请访问：

IBM.com/software/awdtools/studioappdev/