11.4 软件版本控制工具-SVN

对于大型项目通常需要团队开发，为了能管理好项目的整合及更新记录，在项目组启动之初要建立SVN服务器，项目管理员把项目文件和公共文档放入SVN服务器，项目组成员通过SVN客户端从SVN服务器下载，完成自己的任务后提交给SVN服务器。这个过程我们通常称为是**SVN版本控制**。有关SVN的详细说明大家可参阅官网的教程或相关书籍，本章仅对本书案例介绍如何应用SVN服务器。

另外，对SVN环境的配置采用的是VisualSVN server服务端和TortoiseSVN客户端。VisualSVN server可从apache网站http://subversion.apache.org/packages.html下载。TortoiseSVN客户端的官网下载地址是http://tortoisesvn.net/downloads. html。

11.4.1 项目管理员配置SVN服务器

1.  VisualSVN server服务端配置：我们创建的版本库为BaseCode\_shoes，项目组成员有五位，分别为shoesmanager、coder1、coder2、coder3、coder4。其中shoesmanager为项目主管人员，其余四个用户为程序员用户，权限均为Read/Write。如图11-102所示：

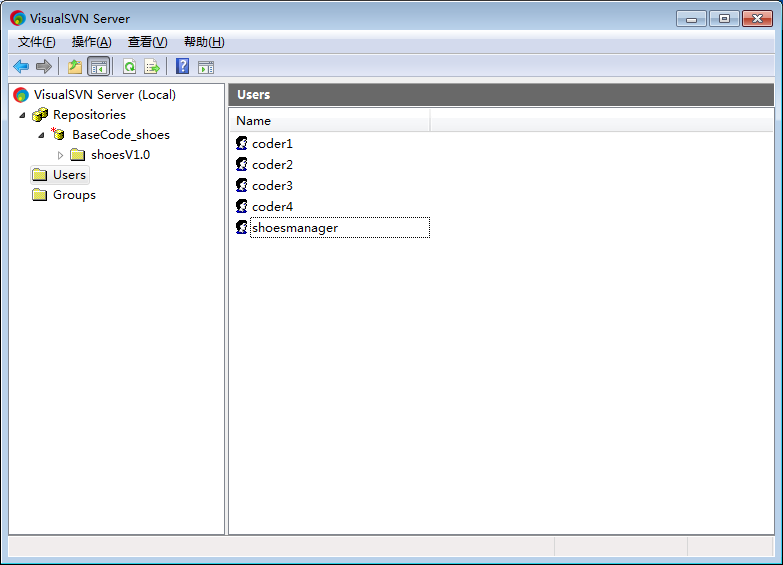


图11-102 SVN服务器代码库和用户设置

2.  shoesmanager将项目文档及框架代码整理后上传到SVN服务器。

1) 在版本库新建共享目录，如图11-103所示：

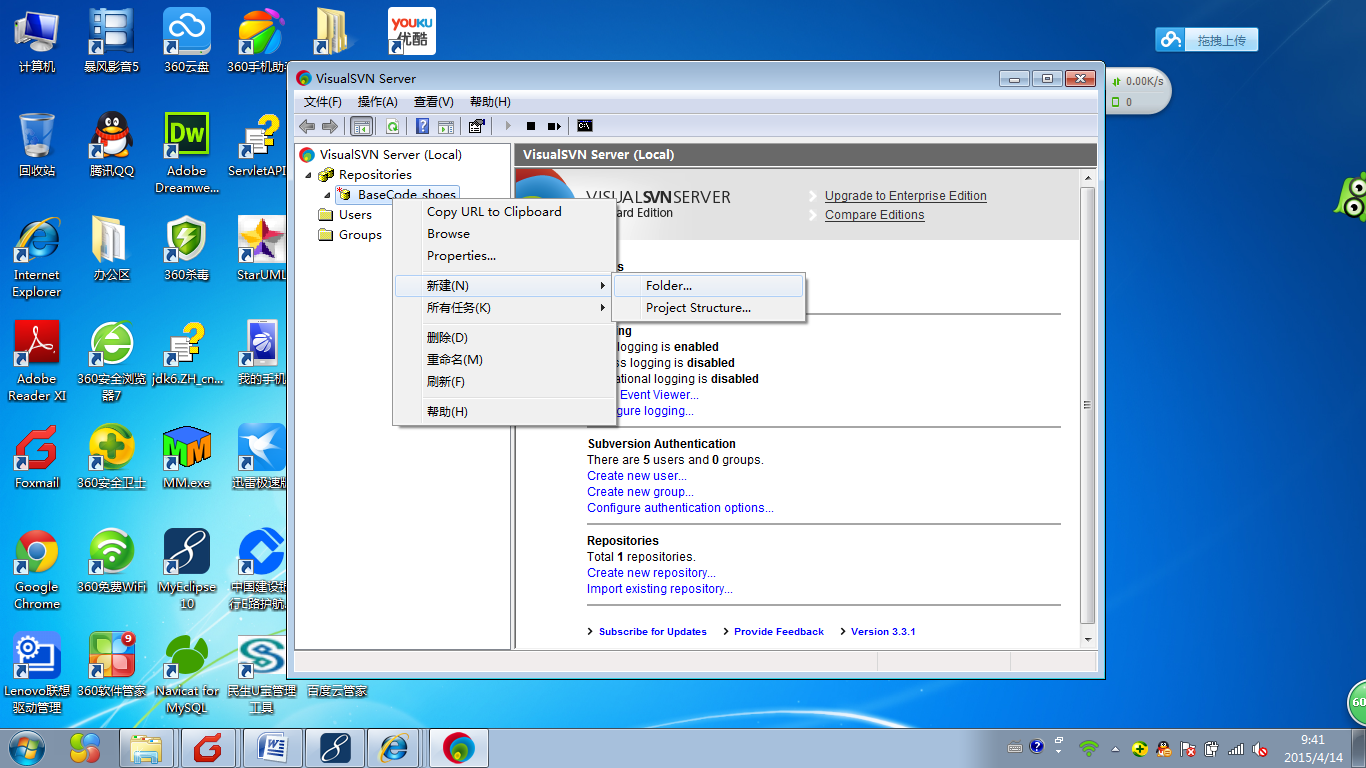


图11-103 代码库新建项目目录

将项目目录命名为“网上商城前台”，如果对后台管理，还可以创建“网上商城后台”目录。如图11-104所示：

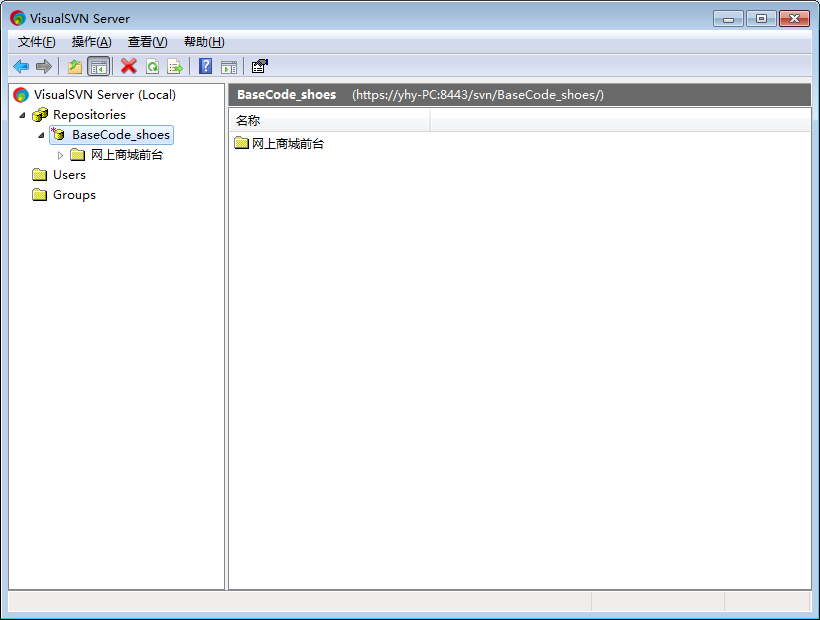


图11-104 创建管理目录“网上商城前台”

2)  shoesmanager用户检出项目目录，shoesmanager在本地计算机选择一个目录如“网上商城”，在目录中右键单击“SVN检出”。如图11-105所示：

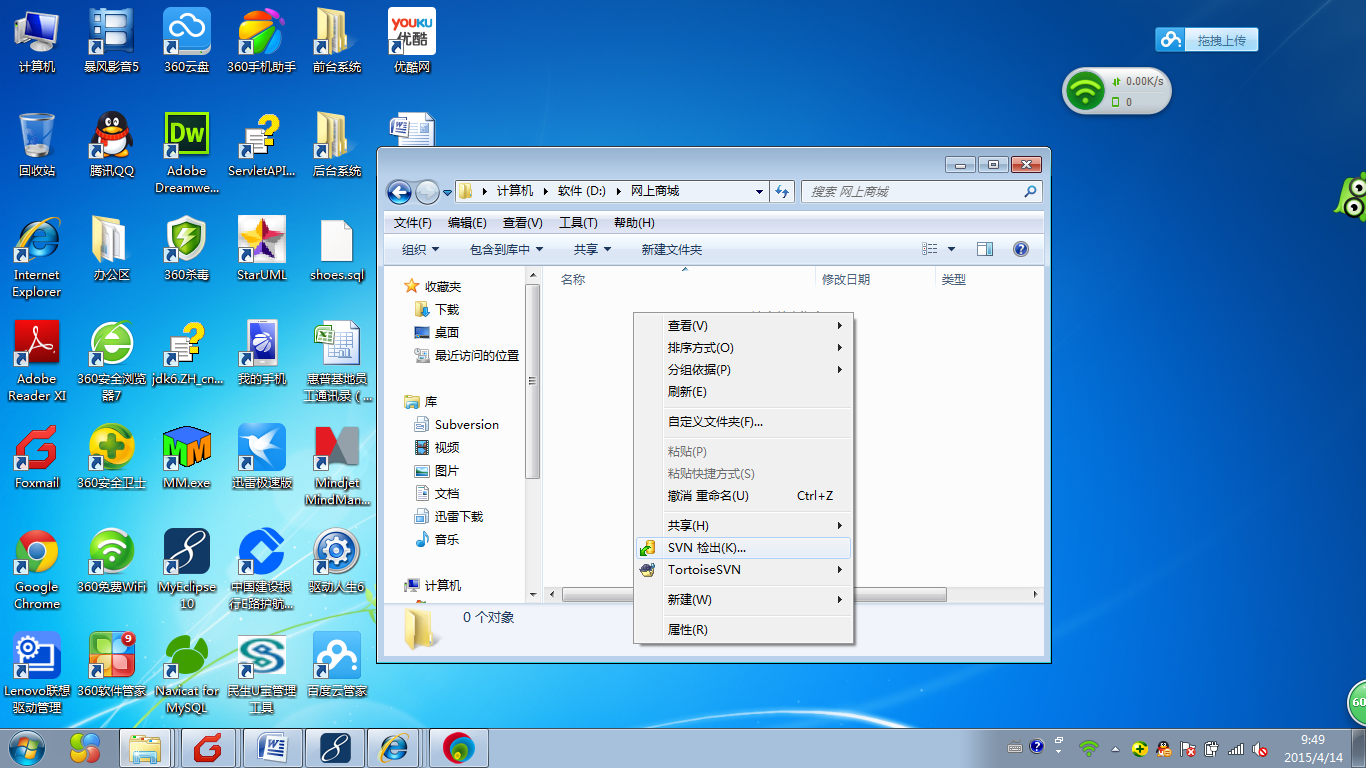


图11-105 检出操作

3) 检出“网上商城前台”到本地目录。因为是一台机器上配置，所以IP地址写成：127.0.0.1。如果SVN服务器在另一单独的机器上，则需要写入那台机器的IP地址。如图11-106所示：

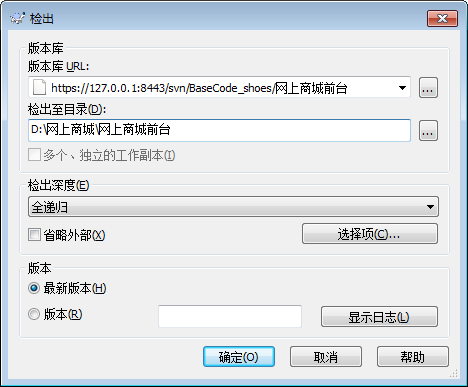


图11-106 “检出”对话框设置

4) 进行身份认证。如图11-107所示：

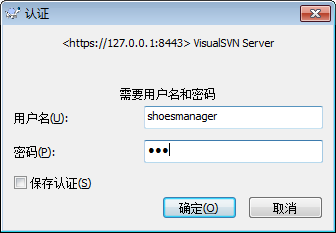


图11-107 身份认证

5) 检出完成。如图11-108所示：

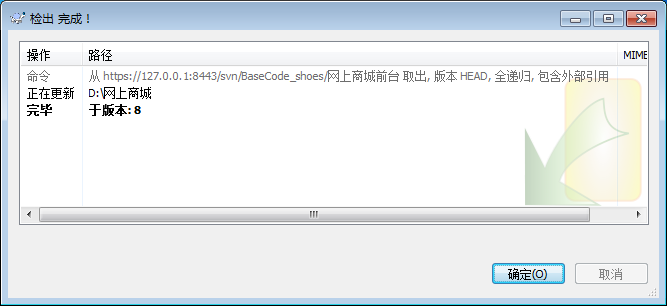


图11-108 检出完成

完成后，shoesmanager用户本地目录就会出现“网上商城前台”目录。如图11-109所示。

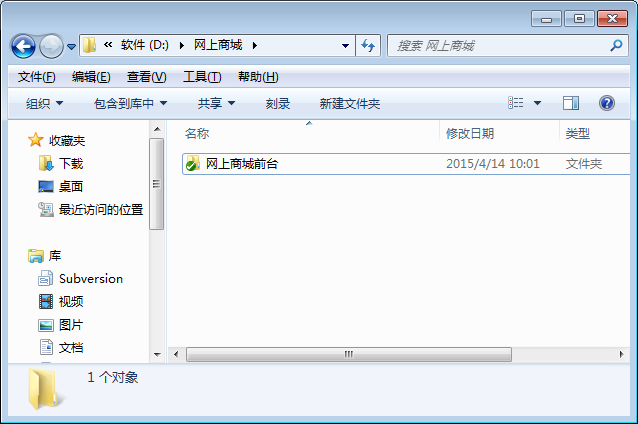


图11-109 shoesmanager用户本地目录

6)  shoesmanager用户配置项目文件。将搭建好的项目框架和与本项目相关的文档放入“网上商城前台”，以供项目组其他程序员检出。新增的文件以蓝色问号标识。如图11-110所示。

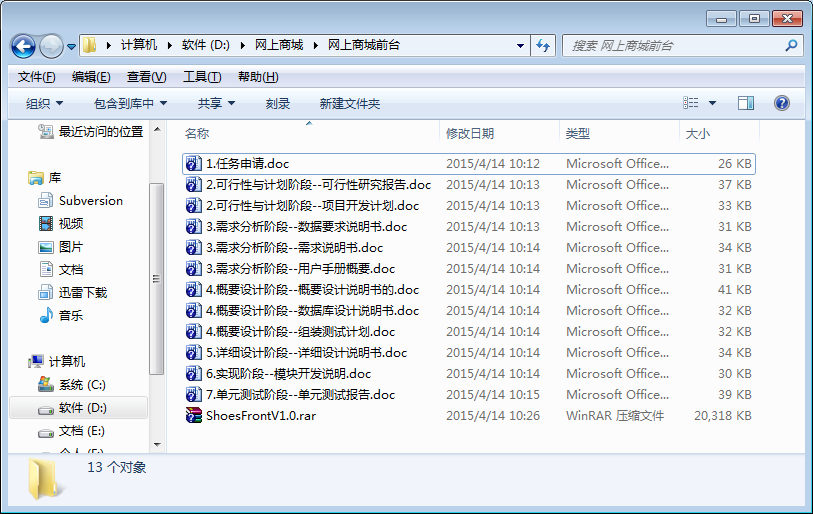


图11-110 shoesmanager配置服务器文件

7)  shoesmanager用户提交服务器。shoesmanager用户放入“网上商城前台”目录的文件会以蓝色问号进行标识，说明这些文件还没有提交给SVN服务器，那么其他用户就无法下载这些文件。右击“网上商城前台”→“SVN提交”。如图11-111所示。

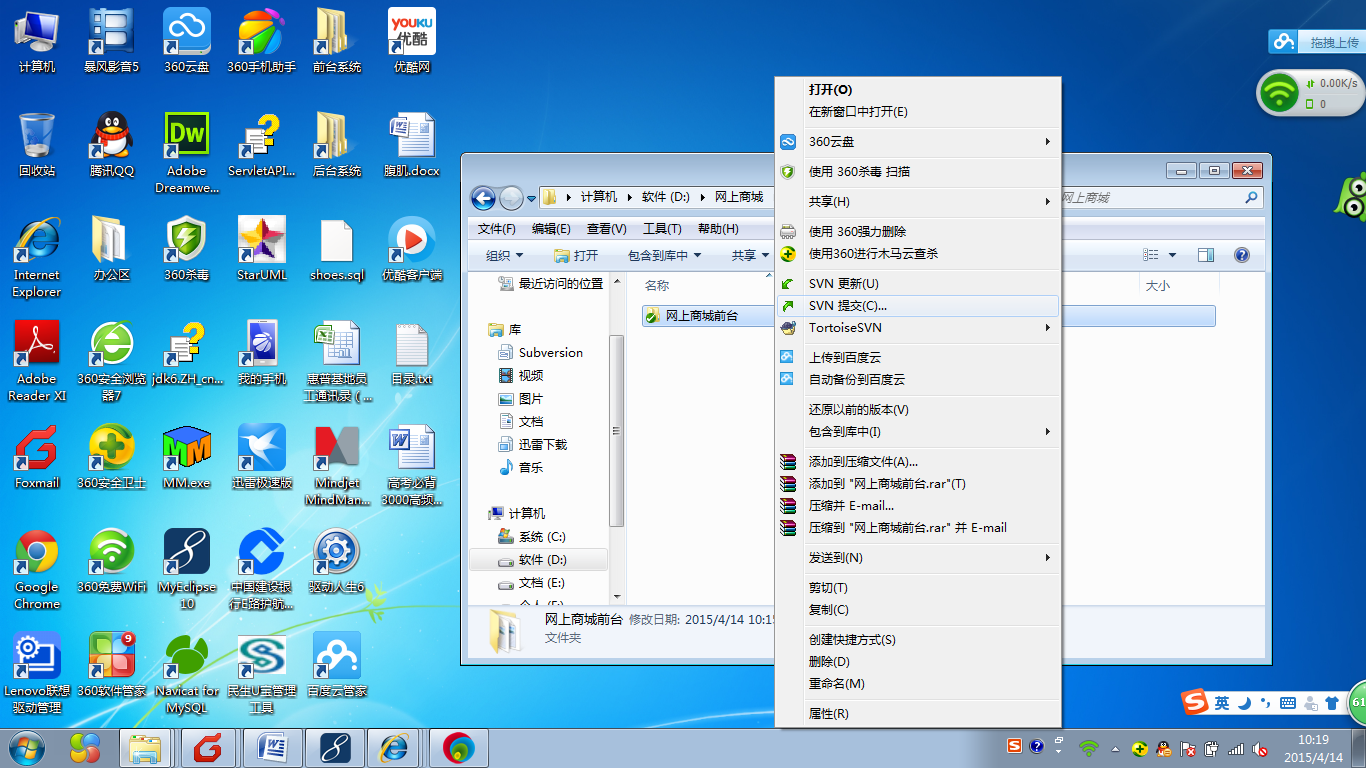


图11-111 文件提交给SVN服务器

提交设置对话框，设置提交哪些文件，如图11-112所示。

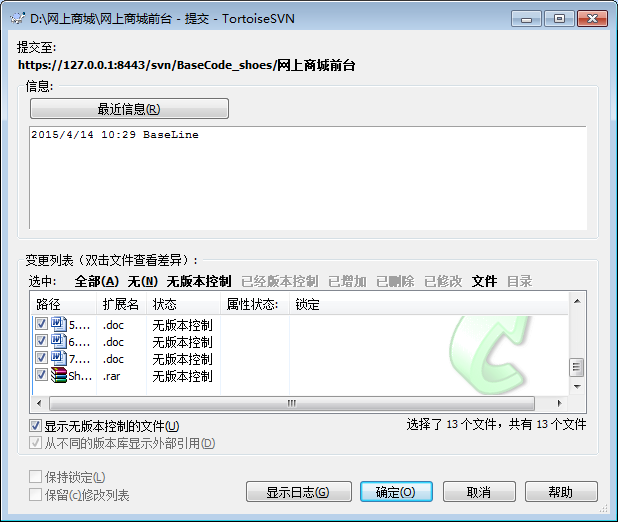


图11-112 提交设置

选好提交文件后，需要再次输入用户名和密码。提交成功后会弹出对话框，如图11-113所示：

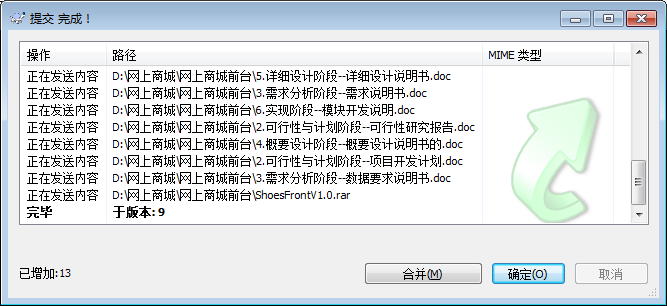


图11-113 提交成功

确定后，shoesmanager用户的本地目录文件标识改成了绿色对号，表示本地文件和服务器文件是同步的。如图11-114所示。

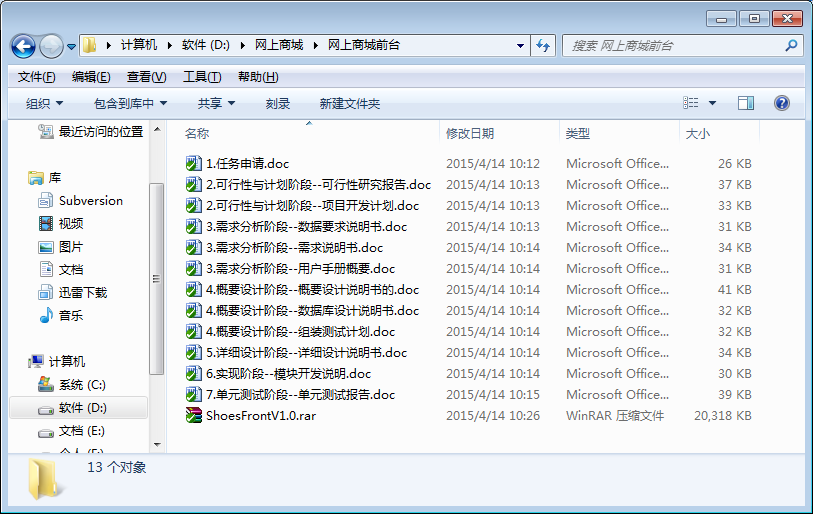


图11-114 shoesmanager用户的本地目录

至此，SVN服务器文件就配置完毕了，当然在实际的应用中要更加复杂，比如给不同的用户分配不同的权限，管理人员的职责等，这些需要大家在实际工作中积累经验。

11.4.2 程序员使用SVN客户端

1. 检出文件到本地，以coder1用户为例，如图11-115所示：

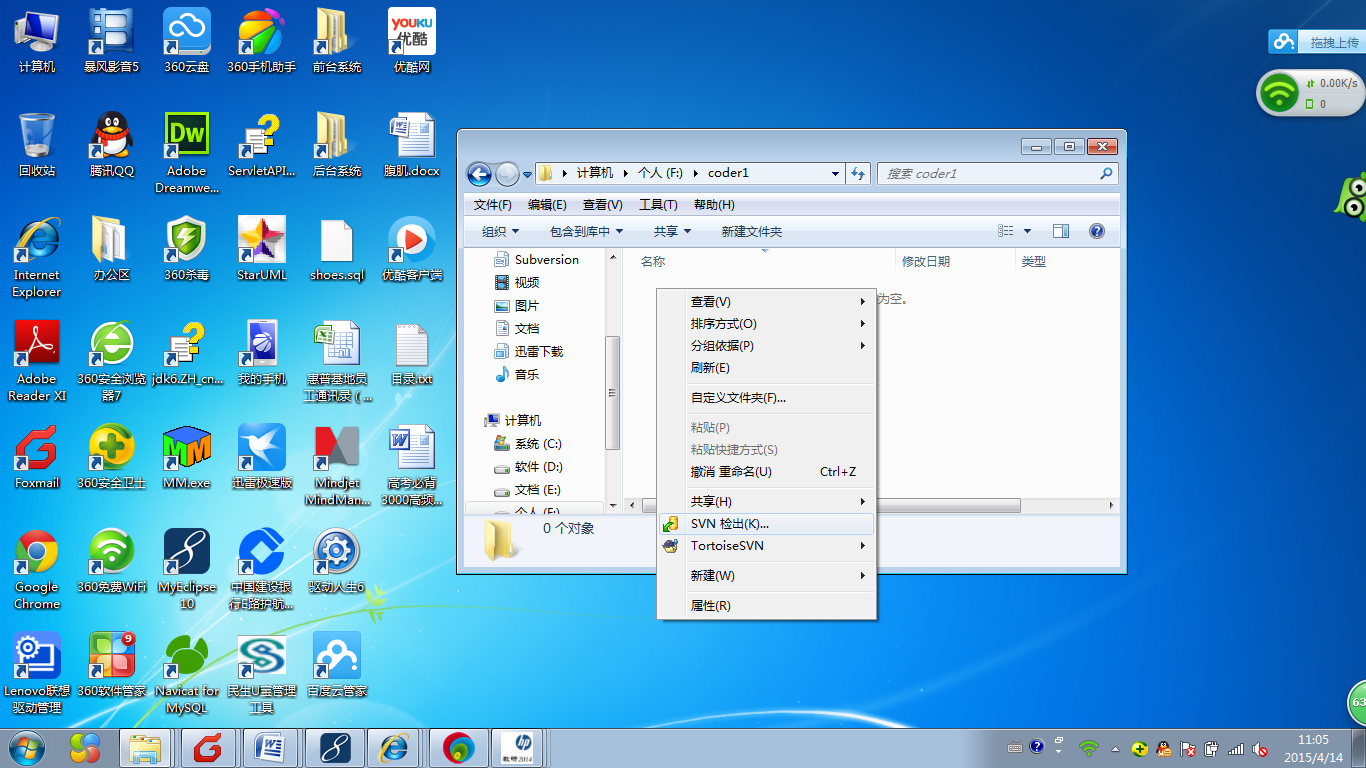


图11-115 程序员coder1检出文件

选择版本库URL及检出目录，如图11-116所示：

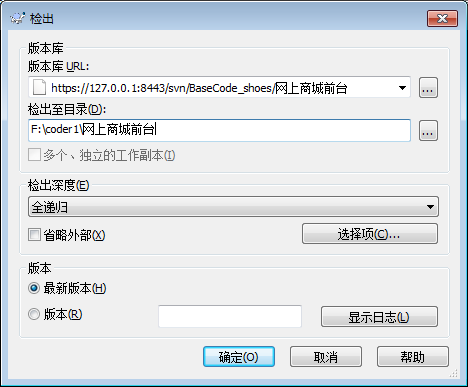


图11-116 选择版本库和检出目录

确定后要求用户输入shoesmanager分配的用户名和密码，并要求该用户具有访问SVN服务器的权限，如图11-117所示：

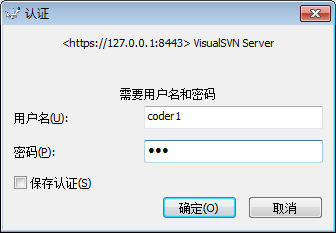


图11-117 输入密码

确定后开始从SVN服务器下载文件，下载成功后，用户本地目录中会保存从SVN服务器所配置文件的副本，文件用绿色对号标识。如图11-118所示：

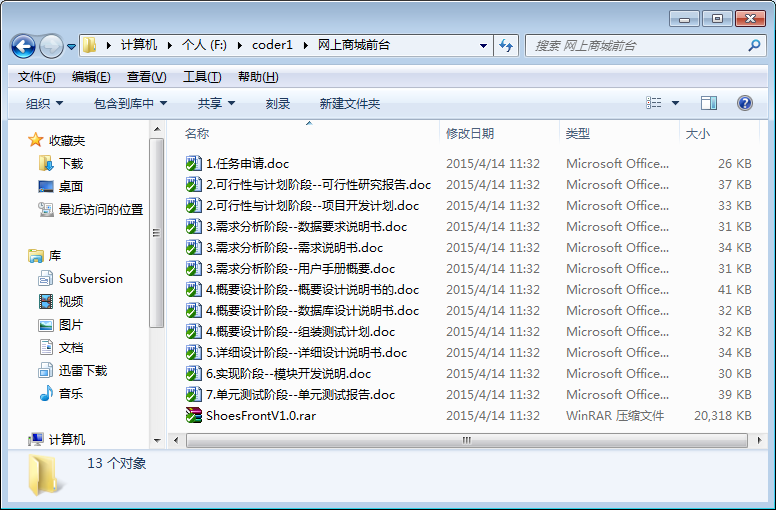


图11-118 coder1用户本地目录

2. 为了进一步说明SVN客户端的常用操作，使用coder2用户也从SVN服务下载“网上商城前台”文件，步骤参考coder1，最终coder2本地目录如图11-119所示：

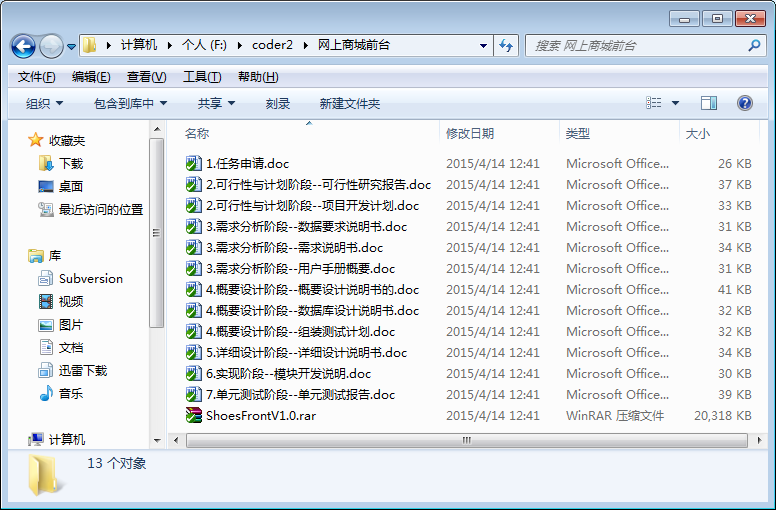


图11-119 coder2用户本地目录

3.  coder1对需求说明书做了修订，如图11-120所示：

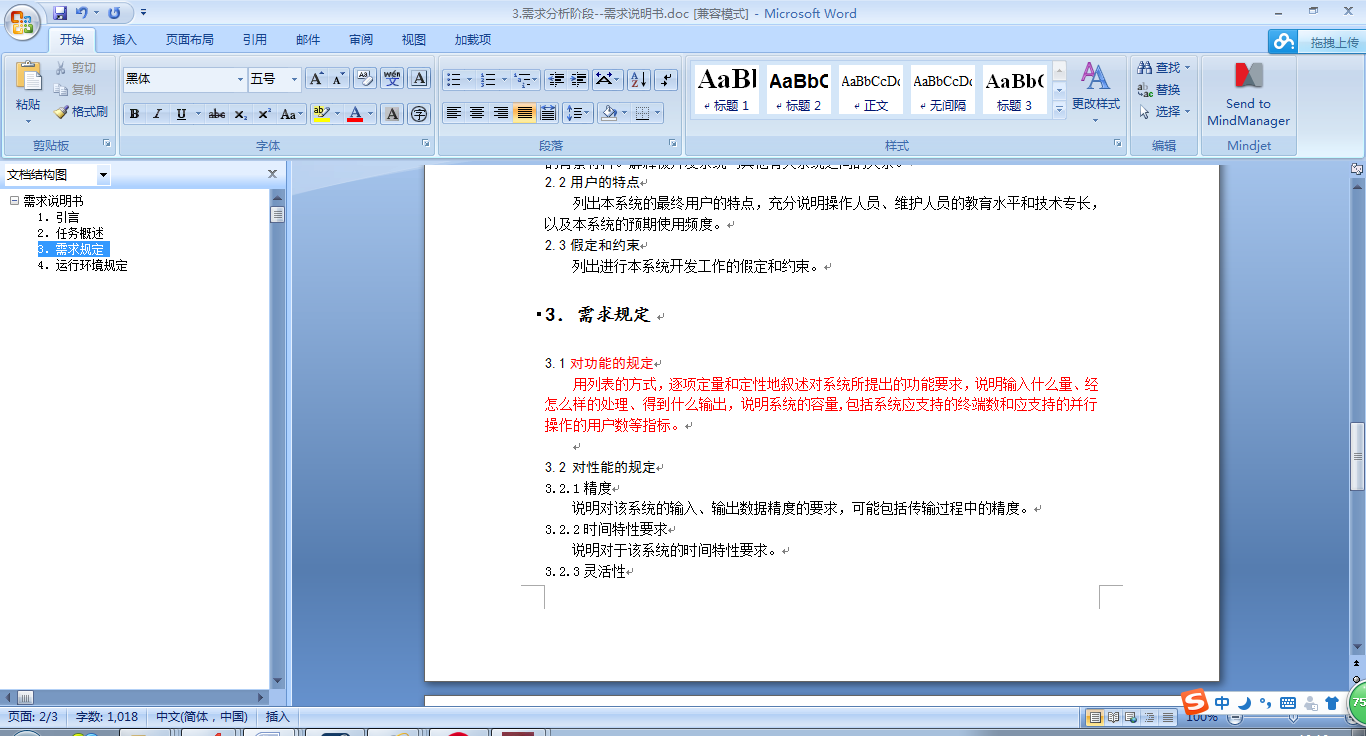


图11-120 模拟coder1用户修改文件

文件一旦做了修改，其标识就变成红色感叹号，表示该文件被修改过。如图11-121所示：

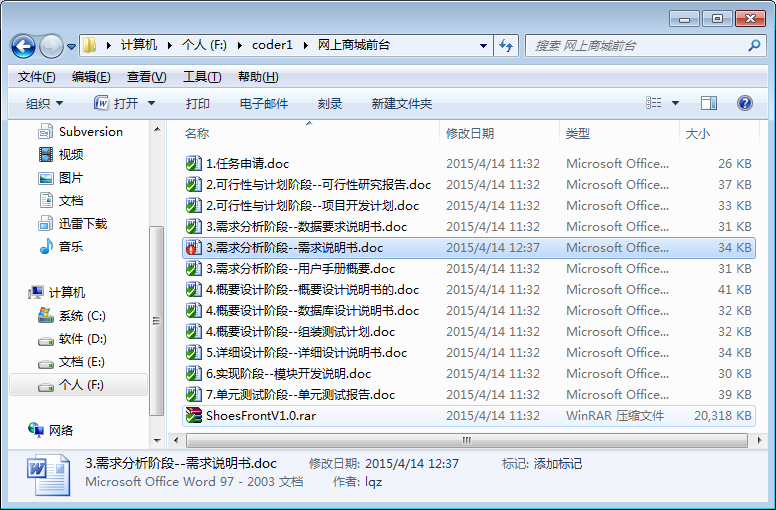


图11-121 coder1用户未更新文件

4.  coder1用户将修改的文件提交到服务器。如图11-122所示：

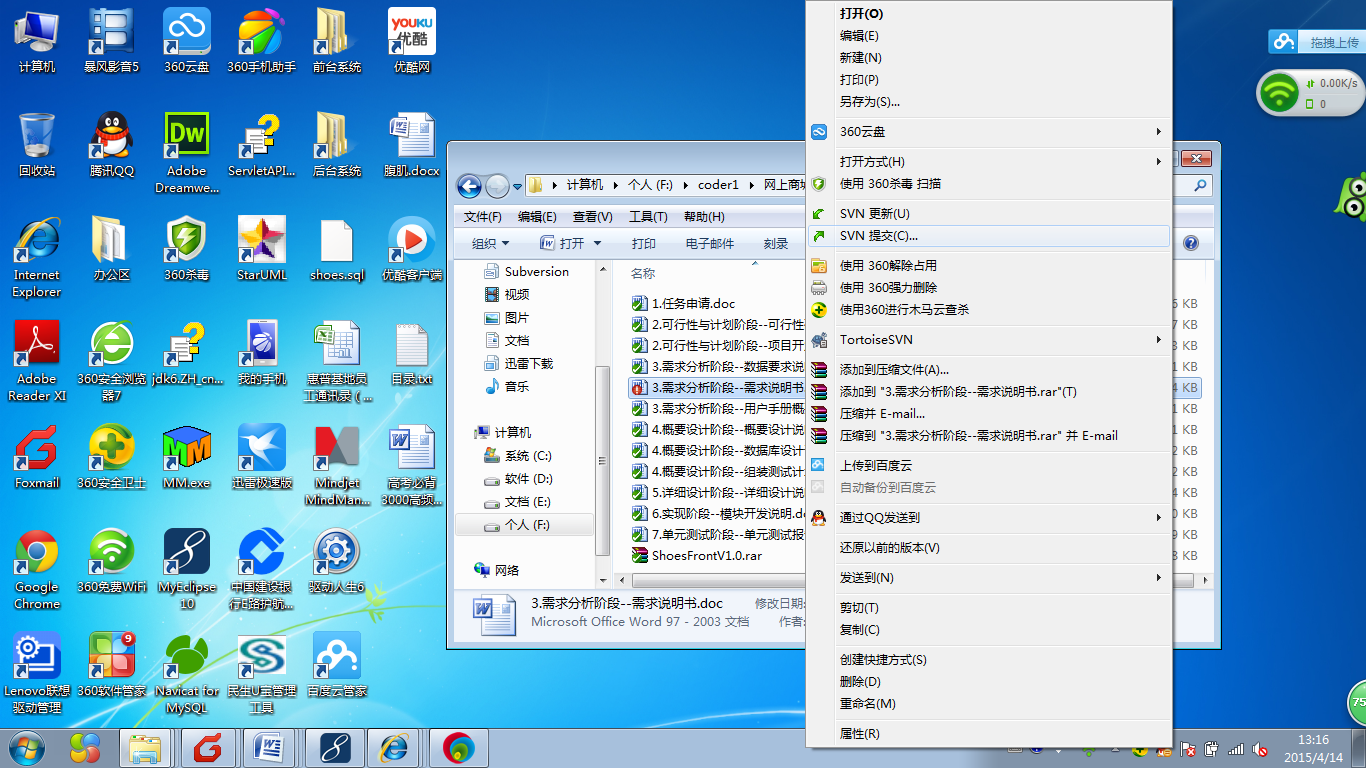


图11-122 coder1用户提交修改后文件

coder1设置提交选项，如图11-123所示：

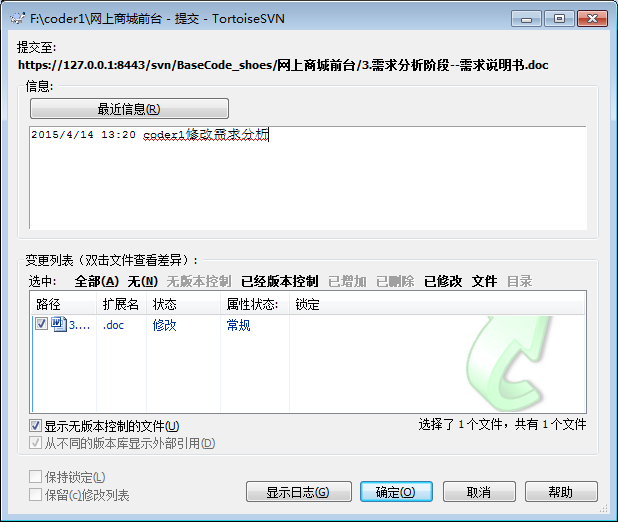


图11-123 coder1设置提交选项

提交完成后给出提示信息，如图11-124所示：

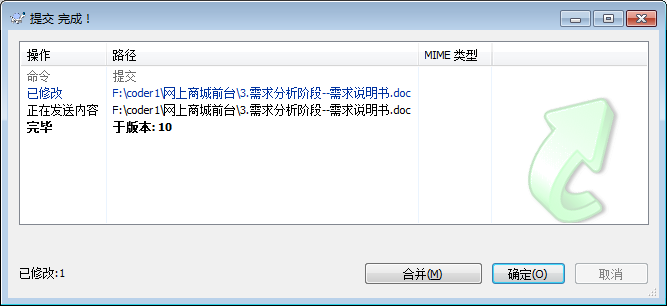


图11-124 修改文件提交完成

5. 现在需求分析文档已被coder1用户做了修改，所以其他用户需要在SVN客户端进行更新才会和coder1同步。如图11-125所示。

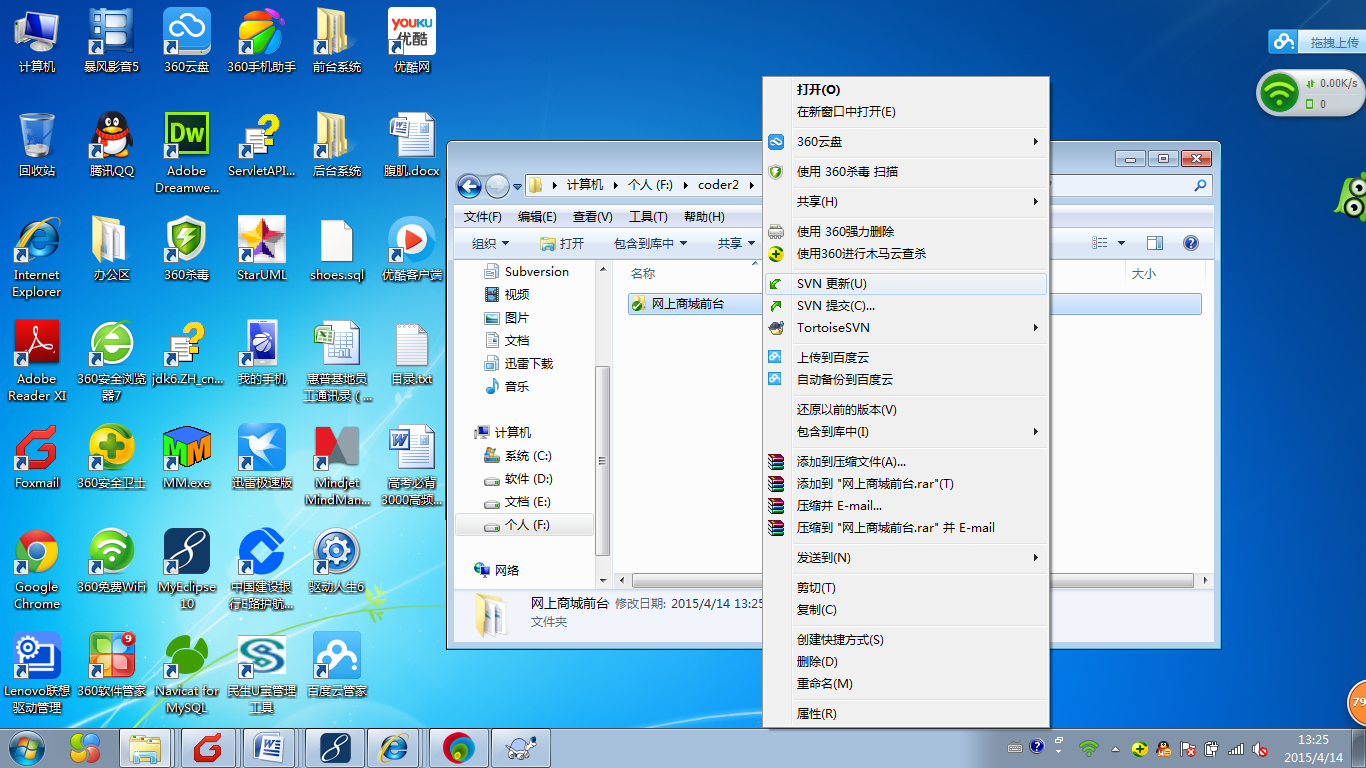


图11-125 coder2用户更新文件

coder2用户输入正确的用户名和密码后，程序开始更新，更新成功弹出如下信息，如图11-126所示：

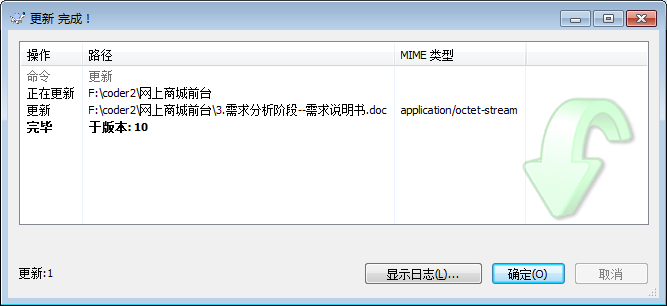


图11-126 coder2更新成功

更新成功后，coder2用户本地目录中的文件就和coder1用户的同步了。在以后的开发过程中，各个用户会不断重复提交或更新。提交是将自己修改的文件上传到服务器，而更新则是将服务器中最新版本下载到本地，如图11-127所示：



图11-127 用户与SVN服务的关系

11.4.3 冲突的解决

项目组成员在利用SVN进行提交和更新时，会有以下两种情况出现。第一种情况：各自修订或新增的是不同的文件，如项目组成员在完成各自模块的过程中，如果各自模块功能独立，则对各自模块的实现类进行修改或新增时不影响其他用户。第二种情况：多个用户同时修订一个文件，如做SSH框架项目时需要对同一个struts.xml和applicationContext.xml进行配置。那么此时就会产生冲突，如果用户之间不做沟通，后提交的用户提交的文件就会把之前用户提交的文件给覆盖掉，对整个项目会产生较大的麻烦。解决冲突最好的办法就是项目组成员要沟通及时，后提交用户需要手动把之前提交的文件内容与自己修改的文件进行合并，然后再提交。另外在提交时注意编写提交信息和查看最近信息，如图11-128所示：

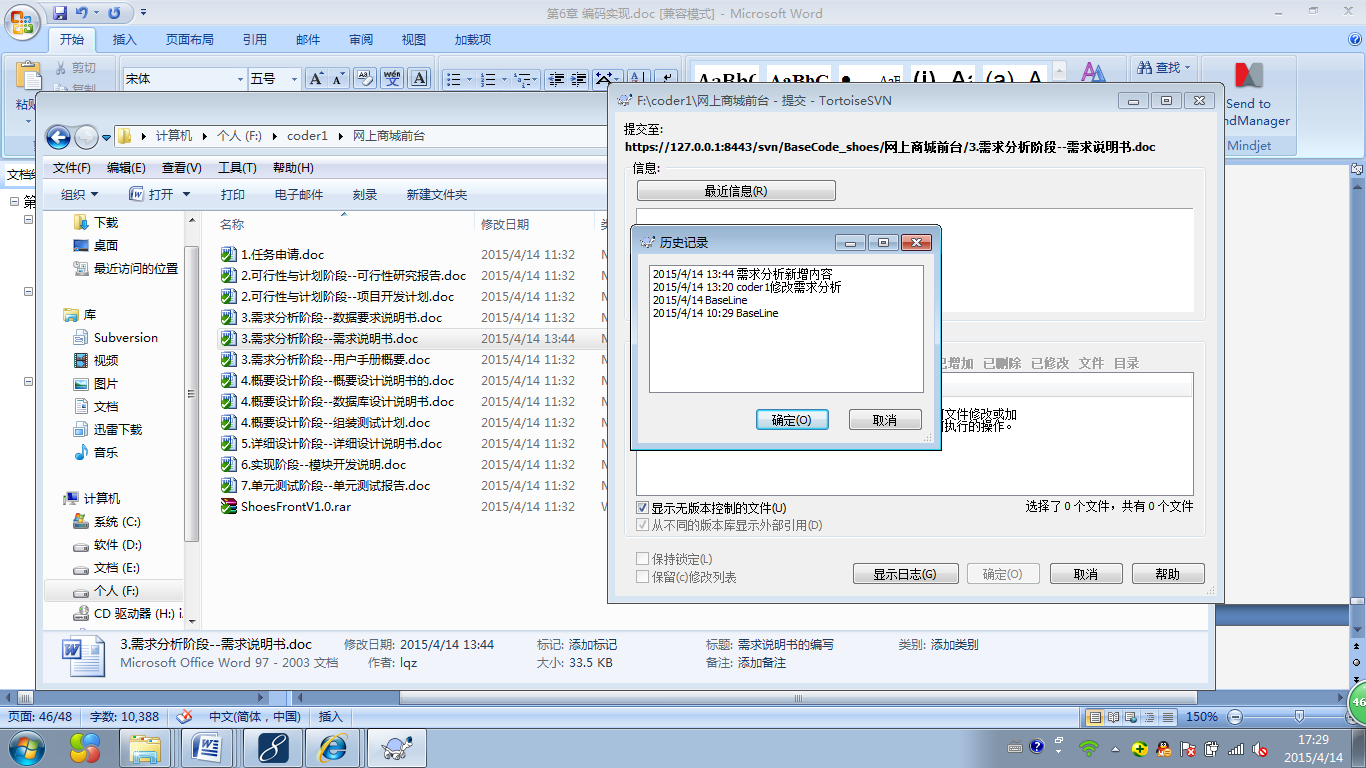


图11-128 用户提交时查看最近信息

11.4.4 MyEclipse SVN插件使用

程序员在使用MyEclipse或Eclipse开发工具做开发时，可以通过安装SVN插件，在IDE环境中直接使用SVN版本管理工具，给开发过程带来便利。所以在开发阶段，程序员通常通过MyEclipse或Eclipse的SVN插件subclipse对所开发的项目进行版本控制。关于插件，大家可以从<http://subclipse.tigris.org/servlets/ProjectDocumentList?folderID=2240>地址进行下载。本书采用的插件版本为1.8.22，如图11-129所示：

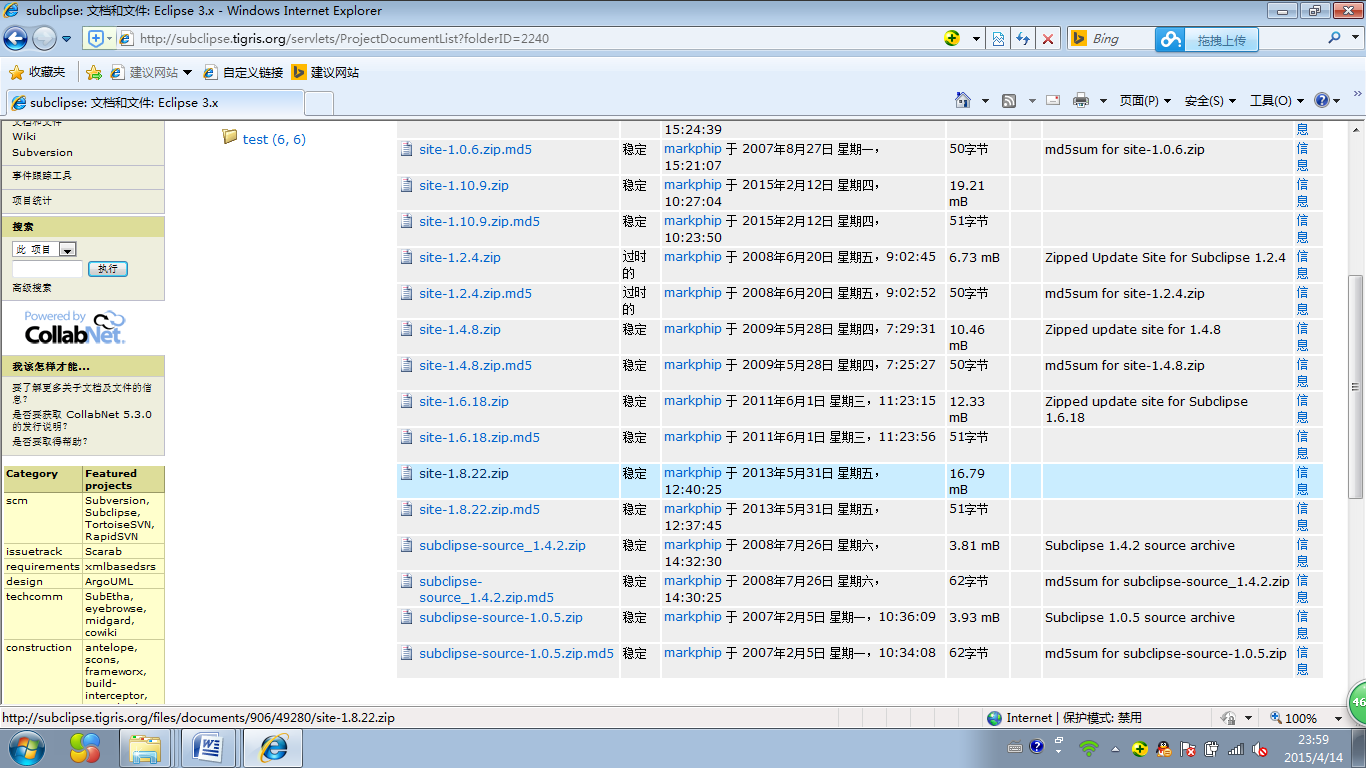


图11-129 subclipse下载版本

1. 解压下载的sit-1.8.22压缩包，如图11-130所示：

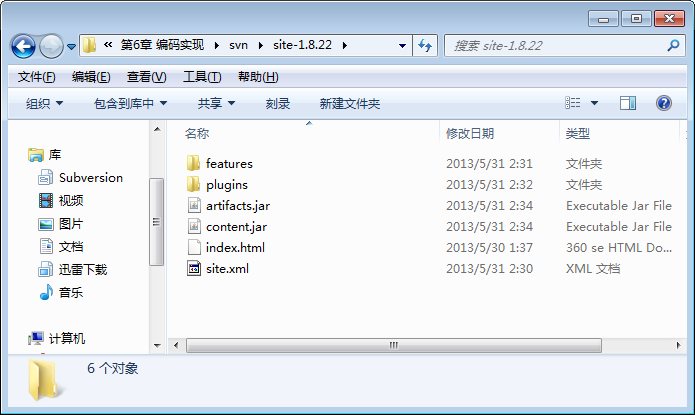


图11-130 sit-1.8.22包

2. 打开MyEclipse10安装目录中的dropins文件夹，新建SVN目录，将sit-1.8.22包中的features和plugins复制到SVN目录中。如图11-131所示：

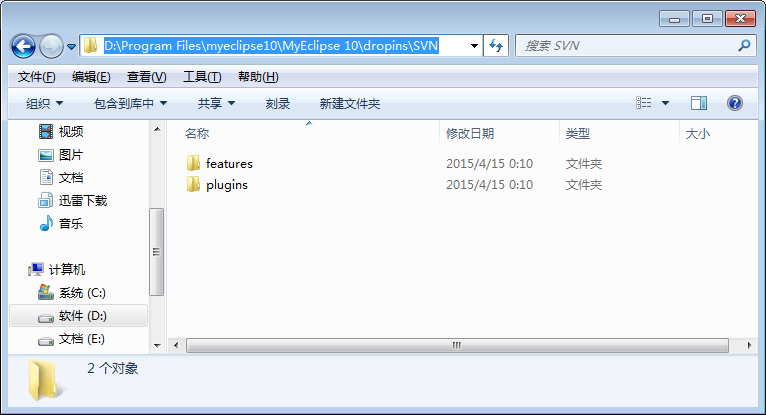


图11-131 在MyEclipse10中安装好SVN插件

3.  MyEclipse使用SVN插件，安装好SVN插件后，启动MyEclipse，这时启动会比较慢，因为MyEclipse要去下载一些相关联的Jar文件，当MyEclipse启动之后进入到workspace时，就会自动弹出提示相关联文件已经自动下载完成的对话框，如图11-132所示：

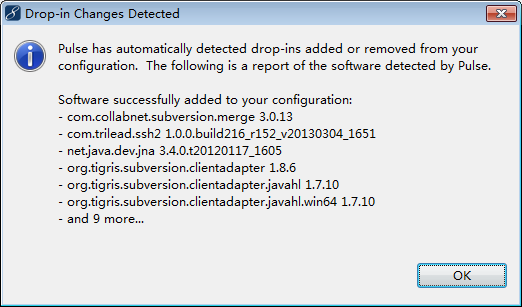


图11-132 关联文件下载完成提示

1) 查看安装好的SVN

单击菜单上的Window→Show View→Other…，如图11-133所示：

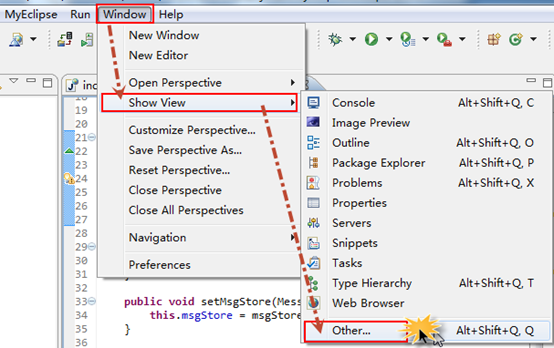


图11-133 查看SVN

2) 在弹出的Show View对话框中就可以看到已经安装好的SVN插件，如图11-134所示：

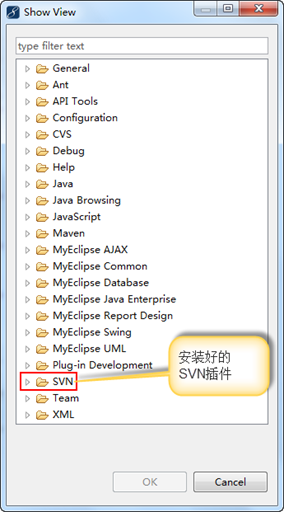


图11-134 MyEclipse中安装好的SVN插件

3) 项目搭建管理员将搭建好的项目上传到SVN服务器。右击项目-Team-Share Project，如图11-135所示：

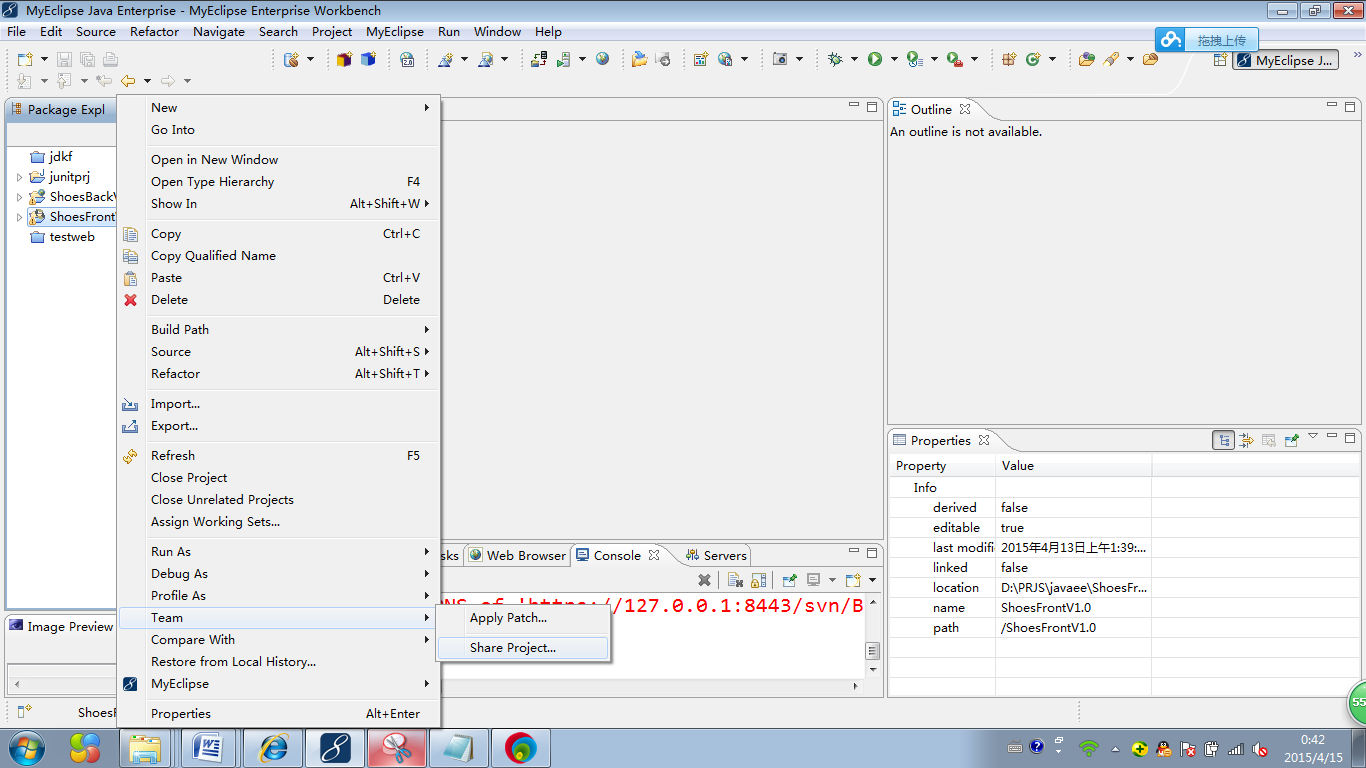


图11-135 共享项目

4) 在弹出的对话框中选择资源库类型，这里选择SVN，如图11-136所示：

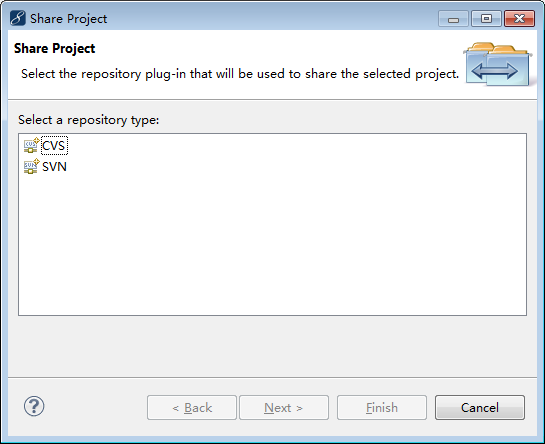


图11-136 选择资源库类型

5) 选择使用已有的资源库位置，就是SVN服务的位置。如图11-137所示：



图11-137 选择资源库位置

6) 选择使用项目名称作为文件夹名。如图11-138所示：

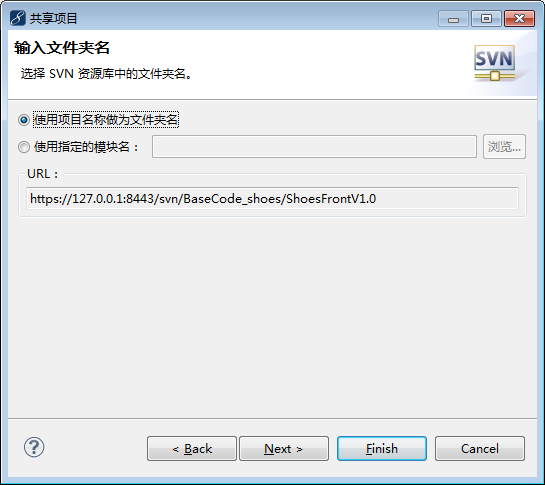


图11-138 设置上传文件夹名

7) 单击Finish后输入用户名和密码。如图11-139所示：

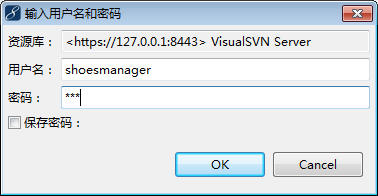


图11-139 输入用户名和密码

8) 单击OK后转到开发环境，发现项目前出现星号标识，需要再提交到服务器。如果某个文件进行了更改，则该文件也用星号标识。如图11-140所示：

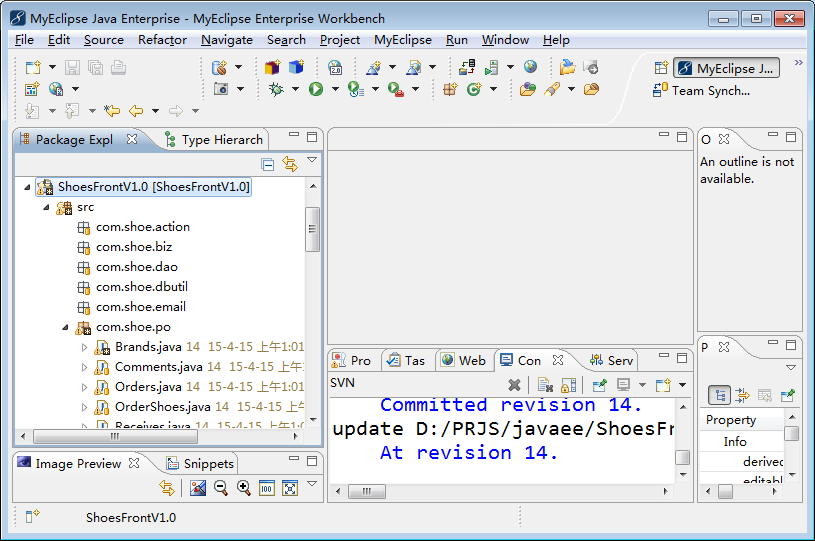


图11-140 MyEclipse团队开发界面

9) 对修改的文件提交到服务器，右击文件，如图11-141所示：

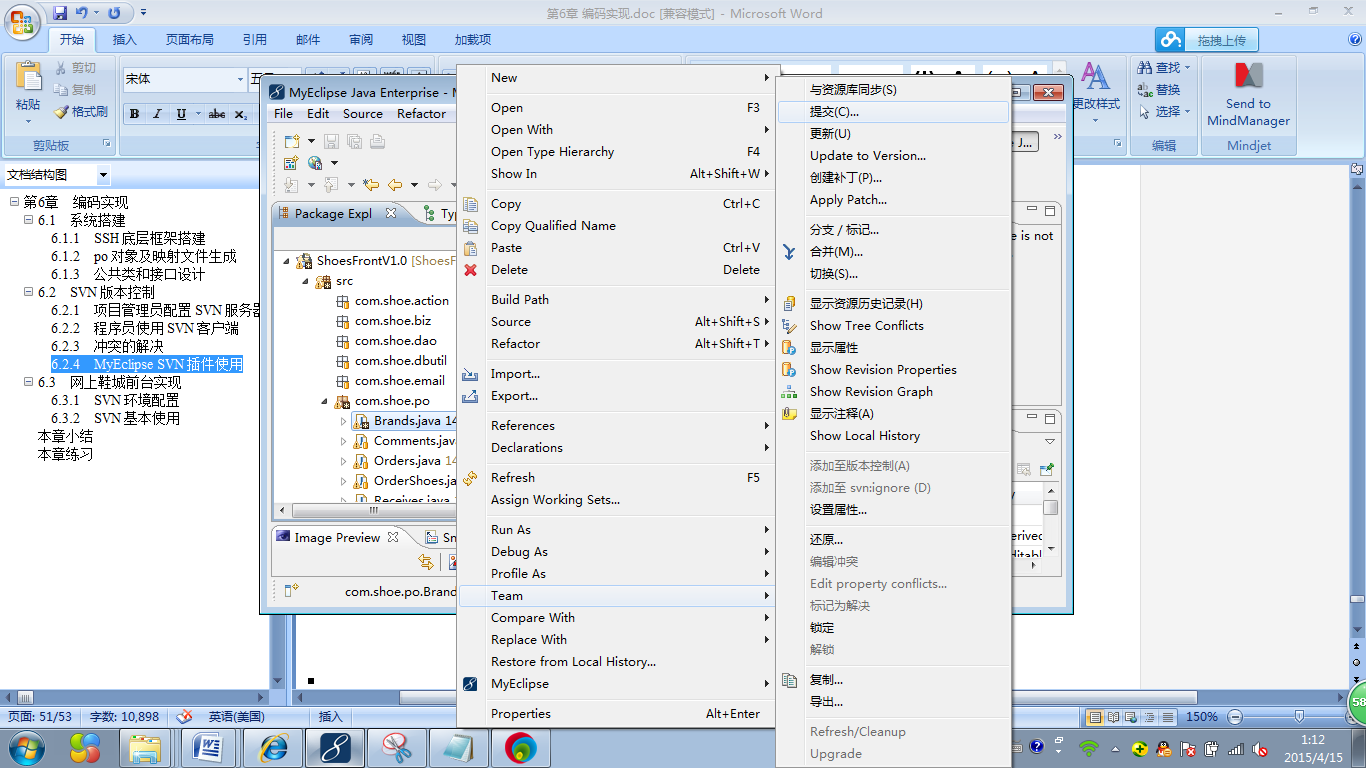


图11-141 提交修改的文件

10) 编写注释后提交，如图11-142所示：

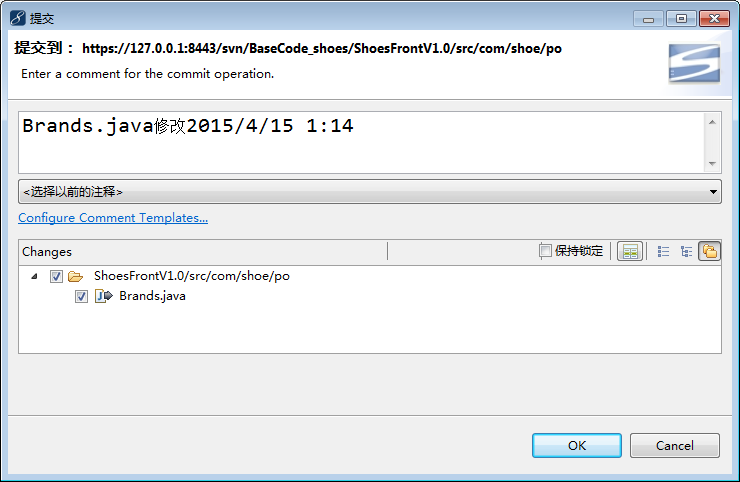


图11-142 提交

至此，我们就完成了在MyEclipse中使用SVN插件，项目组成员接下来的工作就是更新和提交自己负责的代码。当然仍要注意冲突问题，大家可参考11.4.3一节的冲突解决方案。