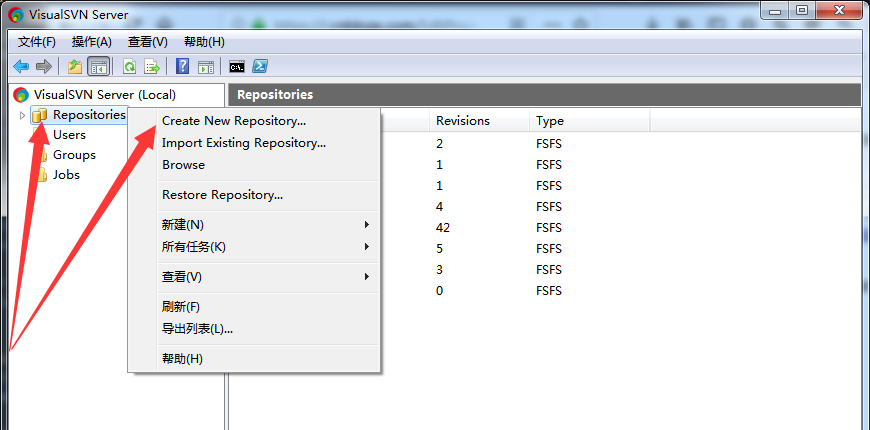
**2.1 VisualSVN创建仓库**

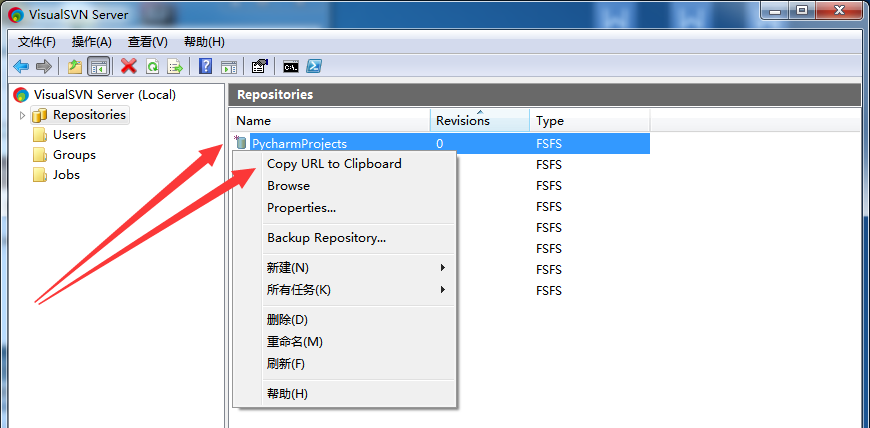
打开VisualSVN，在“Repositories”上右键，点击"Create New Repository"，然后按提示选填一下即可。我这里创建了一个叫“PycharmProjects”的仓库，允许读写的用户名密码为“ls/abcd1234”。



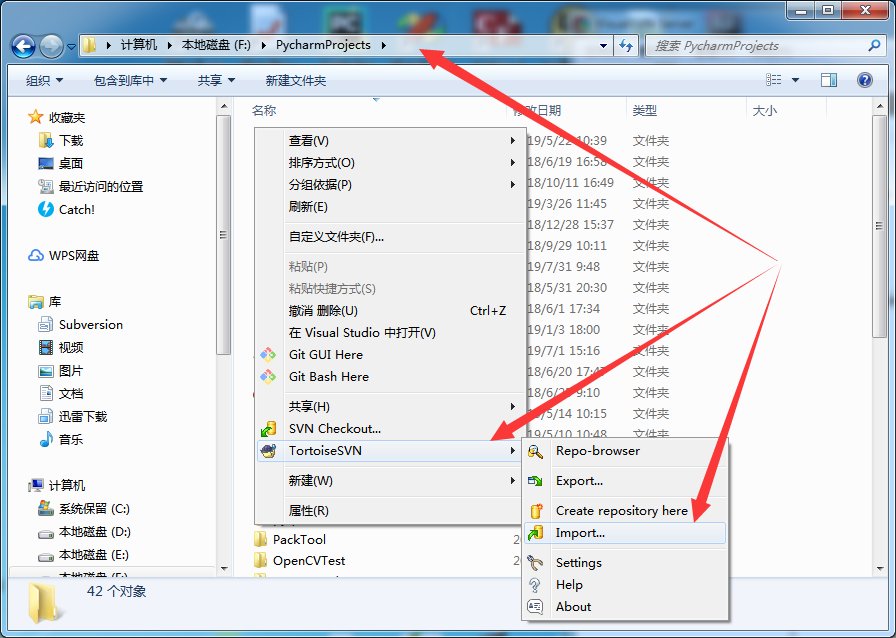
**2.2 向仓库导入文件**

注意，我们这里直接向仓库导入的文件（夹）并不会自动被svn监视，如果想要和Checkout一样被监视，那么需要在导入前先如2.3一样Checkout一把再导入（主要是把.svn给下载下来）；但严谨而言如果是这样的话那就不是Import导入了，而是通过Commit导入，Commit操作见下边2.4。

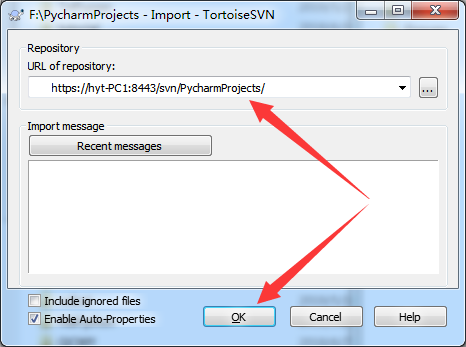
获取仓库URL，选中仓库，右键，Copy URL to Clipboard。



 进入到要导入仓库的目录下，右键，展开TortoiseSVN，选择Import。

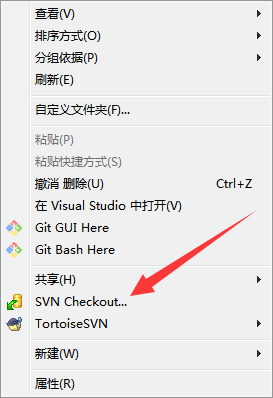


 在“URL of repository”中填入我们在VisualSVN中创建的仓库的URL，然后点击OK即可。

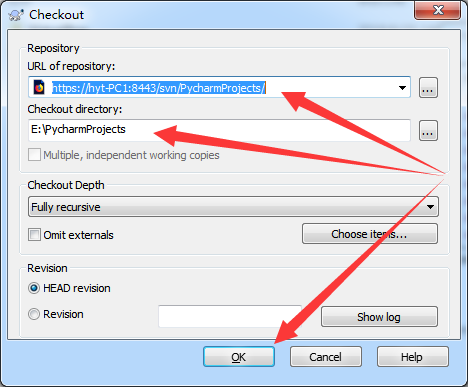


**2.3 从仓库下载代码**

在要下载代码的目录，右键，点击SVN Checkout

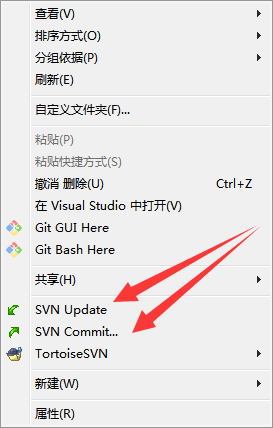


 填好要导出的仓库，要导出到的目录，然后点击OK即可。这里要注意我们向仓库导入的是“PycharmProjects”目录下的文件，是不包括“PycharmProjects”本身的，所以导出时如果仍想导到“PycharmProjects”文件夹，那一定要导出来“PycharmProjects”文件夹。



**2.4 提交修改**

修改文件之后在文件夹下右键，选择SVN Commit即可；但服务端文件可能已被别人先通过SVN Commit修改了，所以建议先使用SVN Update同步服务端文件，如果有冲突先处理冲突，再SVN Commit。



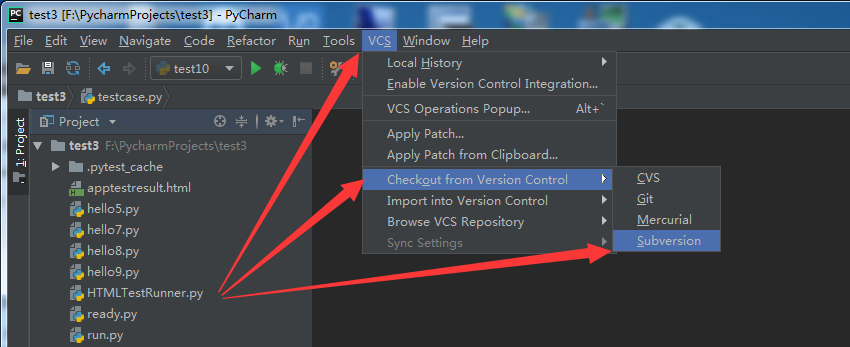
**三、基于PyCharm的仓库创建和使用**

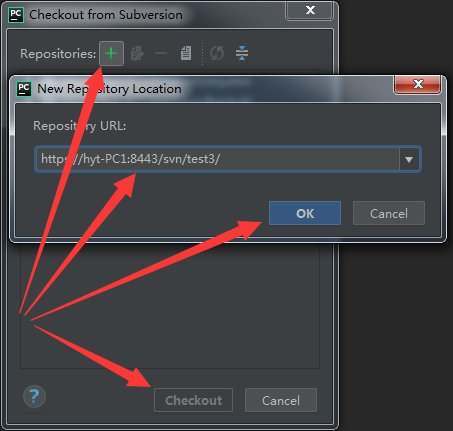
首先要明确，PyCharm使用VisualSVN，他也能且只能遵守VisualSVN的操作，所以PyCharm的仓库创建和使用和没使用PyCharm时的仓库的创建和使用，形式上可以稍加变通但本质上必是一样的。

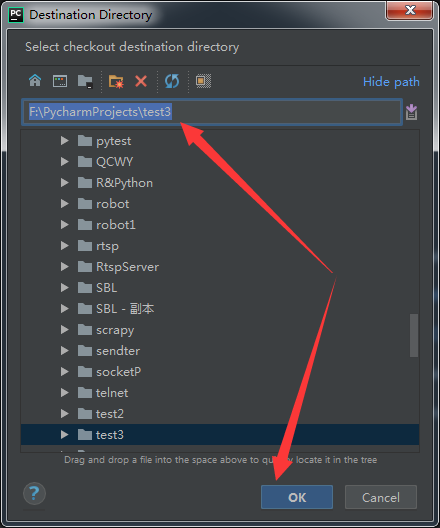
典型的场景是：我们写了一个项目，然后想导入到一个仓库中去，供其他同事下载和修改。我们这里来实现这一需求，假设项目叫test3。

首先，第一步仍是像2.1一样在VisualSVN中仓建一个空的仓库。我这里创建的仓库叫test3。

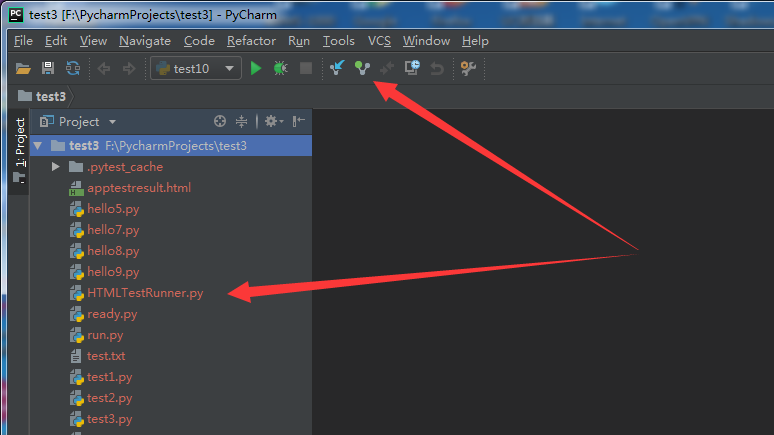
然后，第二步我们在2.2中说过，如果是使用Import那我们本地的后续修改是不被SVN监控的也不能通过Commit同步到仓库中去，所以我们需要先把test3仓库Checkout到当前项目下



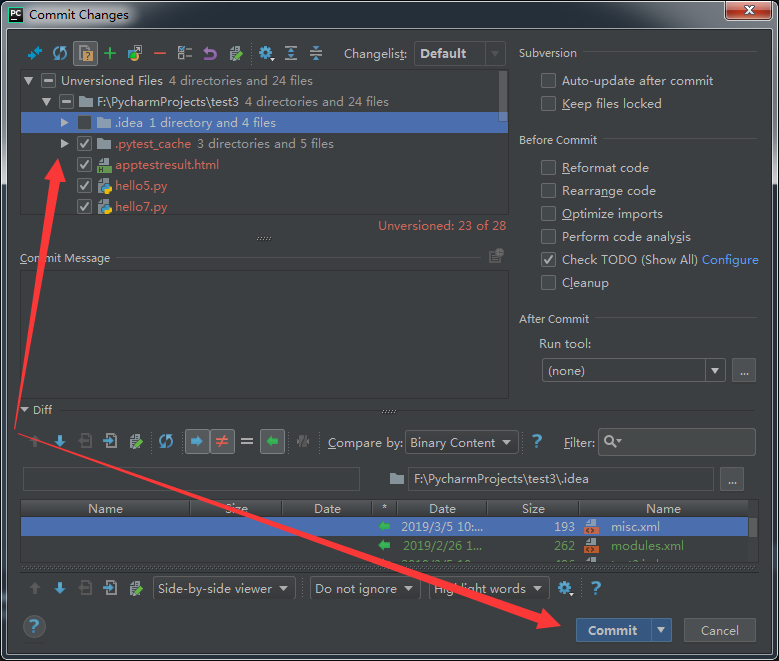




 此时关闭项目再重新打开，就可以看到项目文件被着色（当前红色表示文件未被加入仓库），同时工具栏上多了个版本控制的图标，可点击该图标使用Commit将要加入仓库的文件加入仓库。



 .idea存放的是PyCharm的配置，每个人的配置是不一样的，所以一般不提交到仓库。



后续，其他人想下载代码，直接Checkout或在PyCharm中Checkout即可，就不多说了。

转自https://www.cnblogs.com/lsdb/p/11386874.html