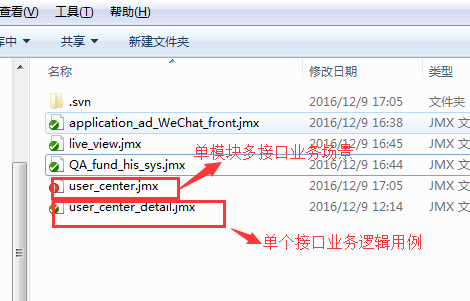
**Jmeter自动化测试用例编写规范**

1. **自动化用例整体结构划分**

用例结构的好坏，将直接影响到了用例的编写、维护、阅读、运行的效率。一个好的用例结构划分需要同时考虑业务模块覆盖和测试覆盖。从常规来说，接口测试可以分为：a.系统级的用户业务场景；b.模块级的用户业务场景；b.单接口业务场景；c.接口合法性校验测试；在jmeter中,系统级用户业务场景作为单独的一个jmx,每个模块级的用户业务场境、每个模块的各个接口业务场景、每个模块的每个接口的参数合法性校验各自做为单独的一个jmx文件存在（并且需要遵循一定的命名规范）。这样划分的目的是一来，确保在用例编写时，相对独立，不会互相干扰；二来，用例维护时，方便定位影响面和位置，提高效率，降低维护成本；三来，可以根据不同测试需求，灵活调整测试层面；

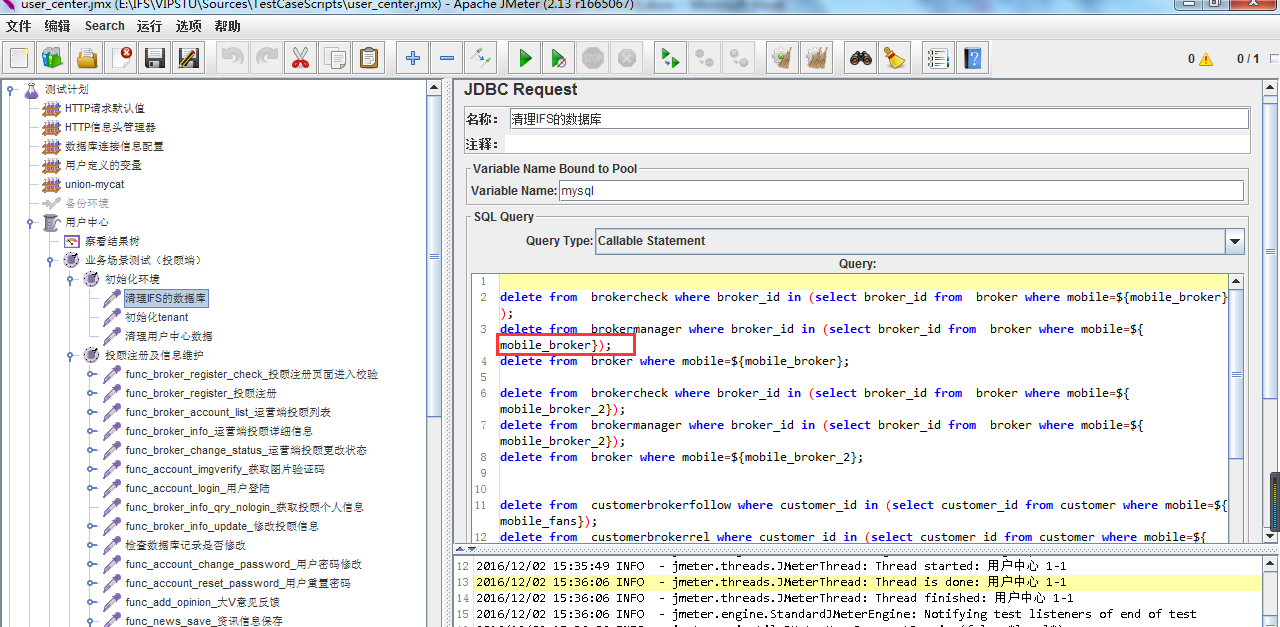
具体操作如下：用户中心业务场景命名为：user\_center.jmx;用户中心单接口业务逻辑用例：user\_center\_detail.jmx ;用户中心参数合法性校验：user\_center\_valid.jmx;



1. **自动化环境维护（手工测试+自动化测试环境统一，减少环境维护成本）；**

为减少测试环境资源、升级维护、数据准备等成本，因此，自动化测试需要直接复用手工测试环境。但这就要求，在执行测试脚本之前，需要预先清理环境，以避免数据冲突而导致脚本执行失败。Jmeter可以直接通过执行Sql语句进行环境初始化，且为保证环境的共用性，不影响他人使用；因此，删除语句需要带上条件，只删除自己即将新增的语句；

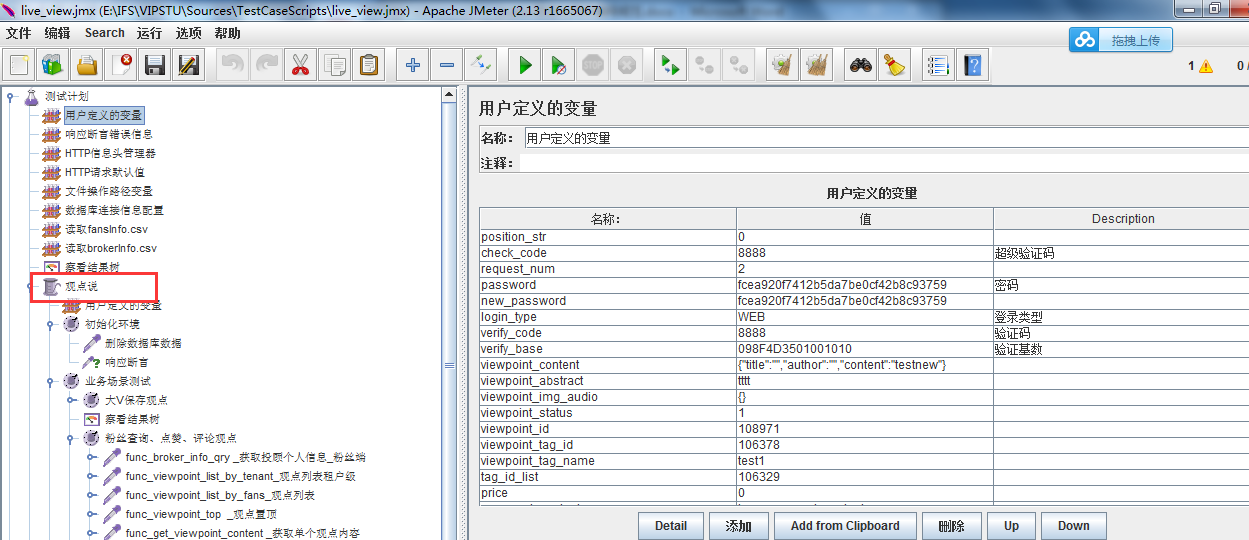
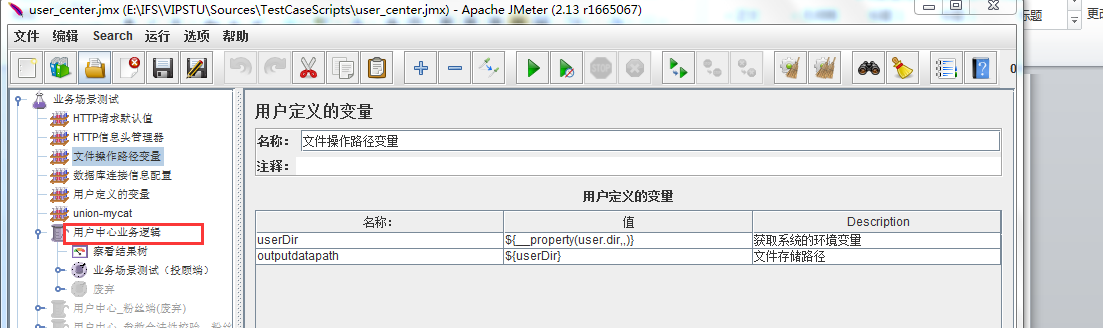
例如，账号注册，唯一索引是电话号码，因此删除语句需要带上电话号码=测试输入数据；如图所示：



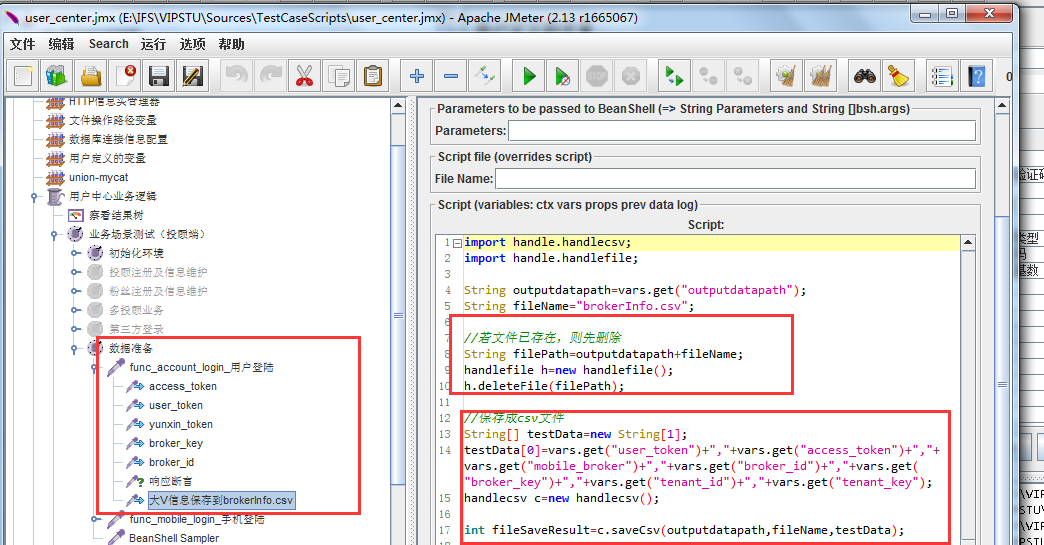
1. **模块间的数据依赖处理；**

涉及到模块之间的数据相互依赖情况，例如需要在客户模块进行开户，才能到交易模块进行交易；需要在被依赖的测试，再通过csv文件或者其他文件格式进行不同模块（线程组）之间的数据传递；具体执行步骤如下：

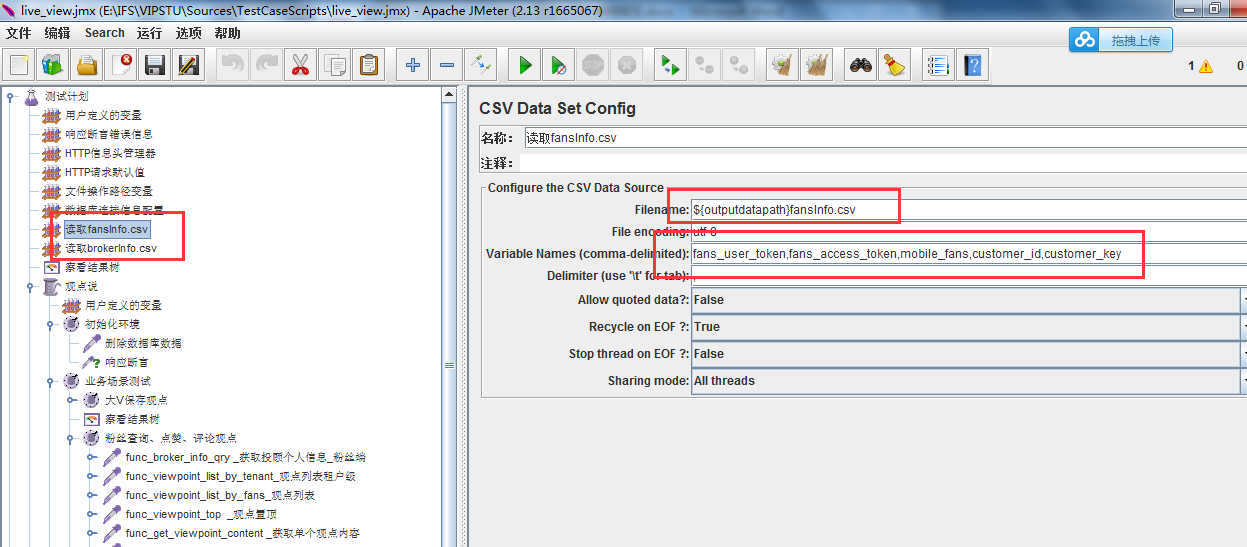
1. 根据业务特点划分业务块，每个业务块一个线程组；例如登录注册一个线程组，交易一个线程组；如图所示：



1. 登录后的账号及返回的user\_token保存到csv文件中；



1. 其他模块引用登录后的账号user\_token等信息，引用方式如下图所示：

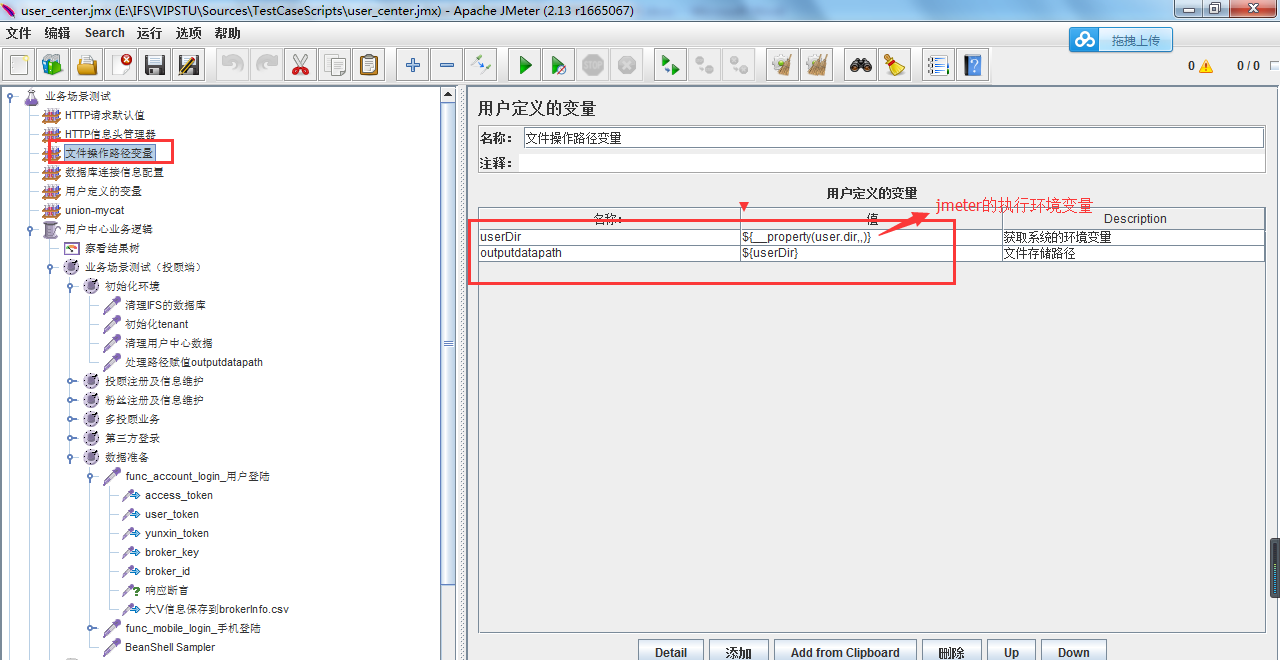


1. **文件的相对路径处理，及linux和windows不同路径符问题解决；**

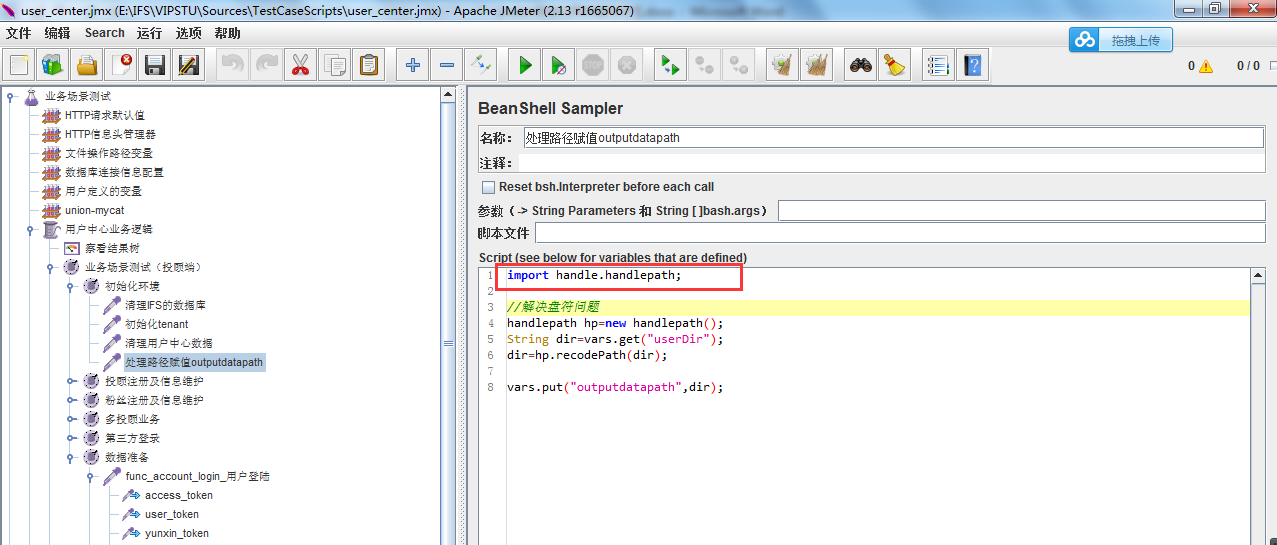
当测试用例中涉及到文件保存、引用时，都需要文件的路径。如果文件采用的是绝对路径，可移植性较差。并且windows和linux的路径盘符不同，windows采用是”\”,linux采用的是”/”;这些问题可以通过几个步骤：1.先通过获取jmeter的执行环境变量“${\_\_property(user.dir,,)}”来获取相对路径；2.调用包handle.handlepath处理windows和linux下不同盘符问题；3.设置获取到的文件路径称为jmeter的属性；以确保其他jmx能获取。4.其他项目里可直接引用；

**具体操作如下：**

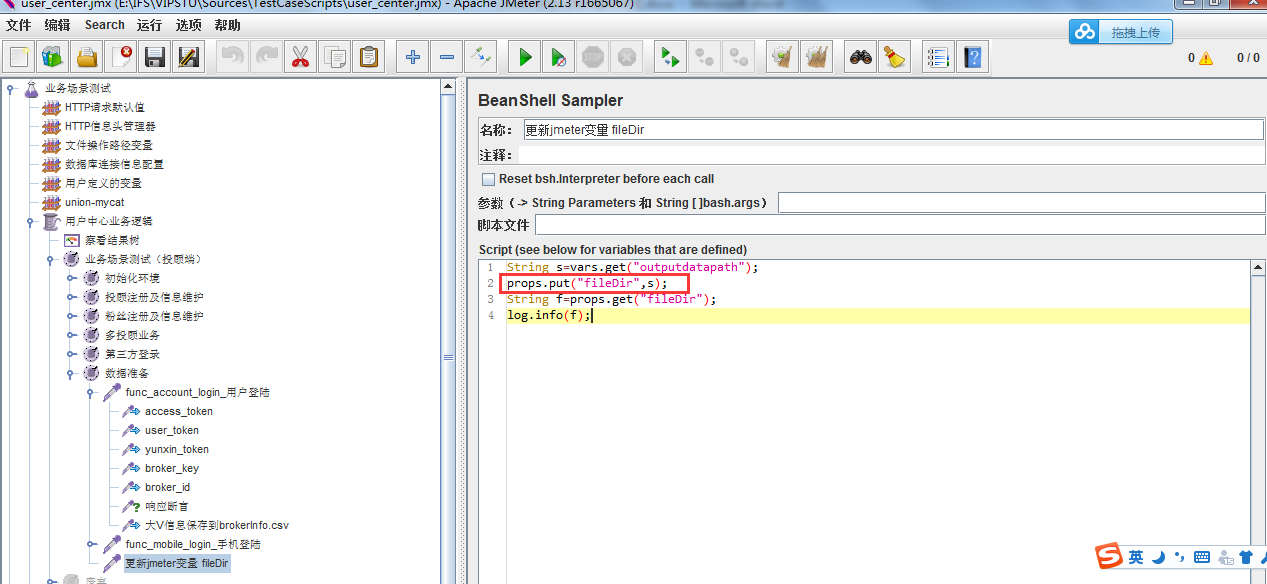
获取文件相对路径方式：



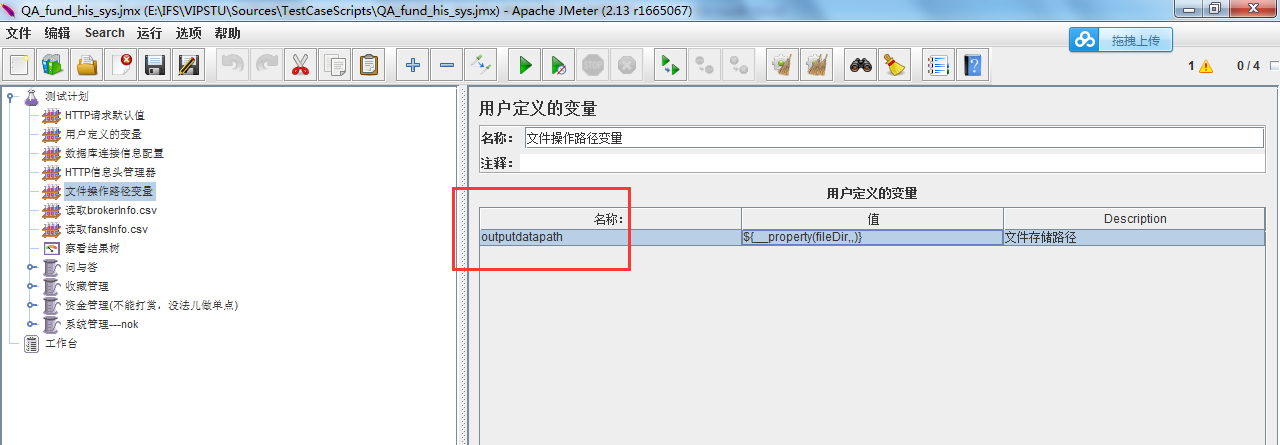
处理linux和windows下的盘符代码：



更新jmeter的文件路径属性：



其他项目的引用模式如下：



1. 完整的一个测试用例包含的几大块如下所示：

