1. **操作变量：**通过使用Bean shell内置对象**vars**可以对变量进行存取操作

1. vars.get("name")：从jmeter中获得变量值。

2. vars.put("key"，"value")：数据存到jmeter变量中。

**二、操作属性：**通过使用Bean shell内置对象**props** 可以对属性进行存取操作

a) props.get("START.HMS");　　注：START.HMS为属性名，在文件jmeter.properties中定义

b) props.put("PROP1","1234");

三、**Bean Shell常用内置变量**

JMeter在它的BeanShell中内置了变量，用户可以通过这些变量与JMeter进行交互，其中主要的变量及其使用方法如下:

* **log**：写入信息到jmeber.log文件，使用方法：log.info(“This is log info!”);
* **ctx**：该变量引用了当前线程的上下文，使用方法可参考：[org.apache.jmeter.threads.JMeterContext](http://jmeter.apache.org/api/org/apache/jmeter/threads/JMeterContext.html" \t "_blank)。
* **vars** - (JMeterVariables)：操作jmeter变量，这个变量实际引用了JMeter线程中的局部变量容器（本质上是Map），它是测试用例与BeanShell交互的桥梁，常用方法：

　　　　a) vars.get(String key)：从jmeter中获得变量值

　　　　b) vars.put(String key，String value)：数据存到jmeter变量中

　　　　更多方法可参考：[org.apache.jmeter.threads.JMeterVariables](http://jmeter.apache.org/api/org/apache/jmeter/threads/JMeterVariables.html)

* **props** - (JMeterProperties - class java.util.Properties)：操作jmeter属性，该变量引用了JMeter的配置信息，可以获取Jmeter的属性，它的使用方法与 vars类似，但是只能put进去String类型的值，而不能是一个对象。对应于java.util.Properties。

　　　　a) props.get("START.HMS");　　注：START.HMS为属性名，在文件jmeter.properties中定义

　　　　b) props.put("PROP1","1234");

* **prev** - (SampleResult)：获取前面的sample返回的信息，常用方法：

　　　　a) getResponseDataAsString()：获取响应信息

　　　　b) getResponseCode() ：获取响应code

　　　　更多方法可参考：[org.apache.jmeter.samplers.SampleResult](http://jmeter.apache.org/api/org/apache/jmeter/samplers/SampleResult.html" \t "_blank)

* sampler - (Sampler)：gives access to the current sampler

**四、自定义函数：**

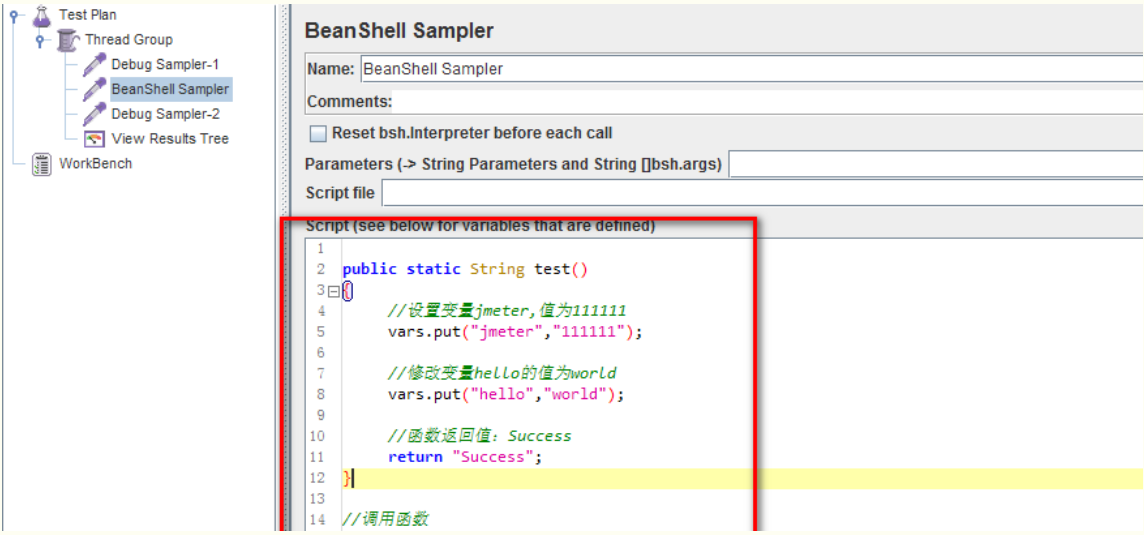
在BeanShell中，我们可以使用java语言自定义函数来处理特定的逻辑，结合BeanShell的[内置对象](http://www.cnblogs.com/puresoul/p/4915350.html" \t "_blank)进行变量的存取，方便我们进行测试提高脚本的灵活性。

示例：

1、在Test Plan中添加一个变量：hello = kitty

2、Debug sampler-1和Debug sampler-2什么都不处理，用来查询对比beahshell处理前后的结果

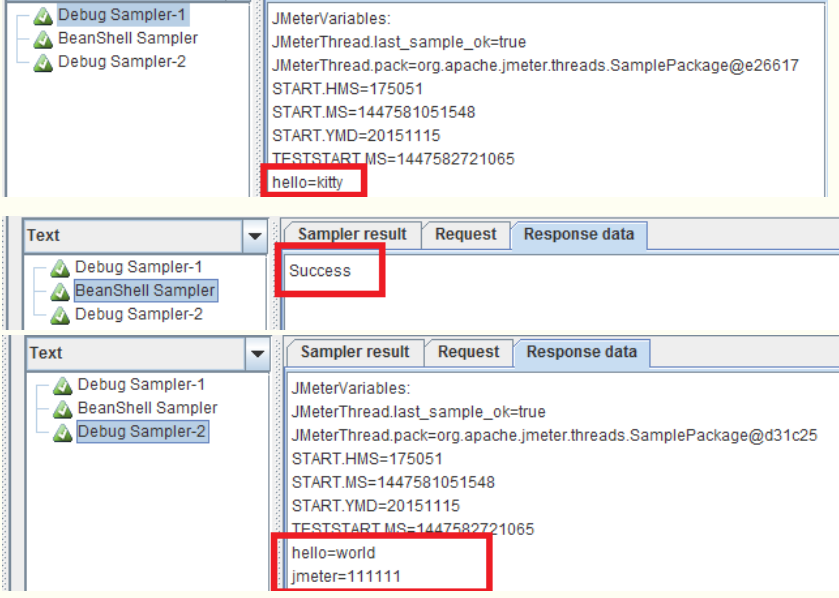
3、BeanShell Sampler中的脚本如下：



　test();

4、运行结果：

* Debug sampler-1中显示：hello=kitty
* BeanShell sampler中 返回结果为：success
* Debug sampler-1中显示：hello=world,jmeter=111111



**五、引用外部java文件：**

有没有觉得上面(三)中自定义函数这样的方式太麻烦并且也不美观？而且如果我们已经有现成的java源文件或者class文件时，我们有没有什么办法直接在jemter中引用？这就是这部分要介绍的内容，直接上示例：

1、假如我有一个java 源文件，名为：Myclass.java，代码如下：

package test;

public class Myclass

{

public int add(int a, int b)

{

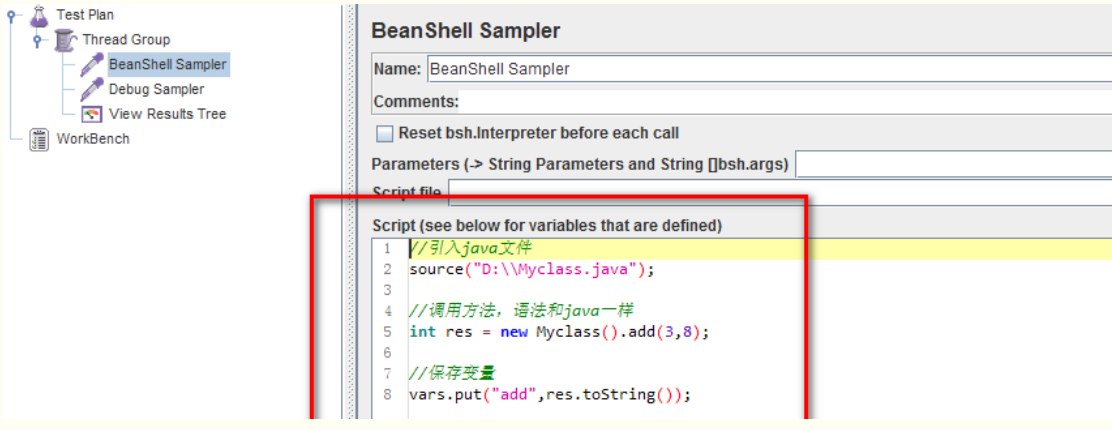
return a + b;

}

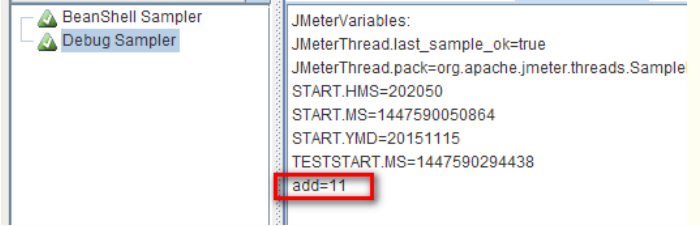
}

2、Bean Shell使用代码如下：

　　在bean shel中通过source("代码路径")方法引入java，然后调用方法和java一样，new一个class,再调用里面的add 方法。



3、运行结果：



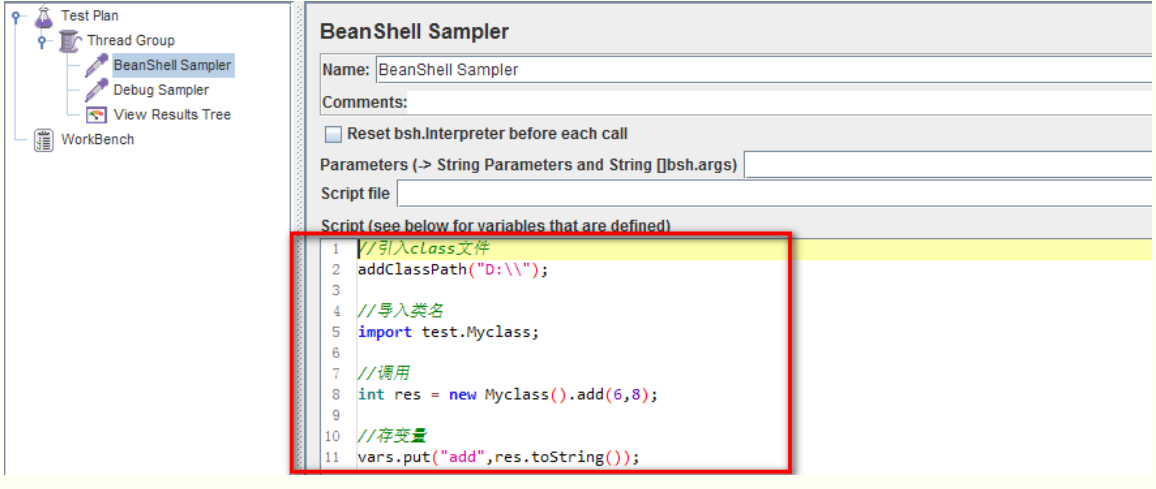
**六、引用外部class文件：**

　　现在知道如何引用外部文件，有时候如果我们只有class文件怎么办呢？其实在jmeter中也可以直接引用class文件，示例如下：

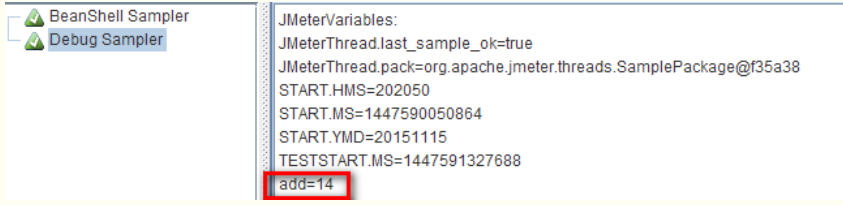
1、直接把上例中的java文件编译成class文件，如何编译请自行百度。

2、Bean Shell使用代码如下：

　　用addClassPath("D:\\")方法引入 class文件，在用import导入包及类，然后就可以像java一样调用了



3、运行结果：



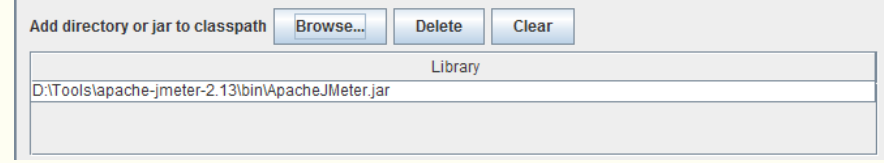
**七、引用外部Jar包：**

上面四、五介绍了如何引用外部java和class文件，如果文件比较多时我们可以把它们打成一个jar包然后在jemter中调用，具体如何使用可以看我上一篇有介绍：[Jmeter之Bean shell使用(一)](http://www.cnblogs.com/puresoul/p/4915350.html" \t "_blank)。

　　在这里想补充一点的是jmeter中引入jar的方法：

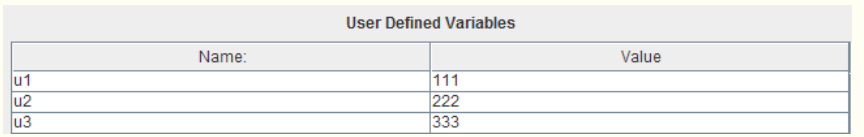
　　1、上一篇中已使用过的：把jar包放到jmeter目录\apache-jmeter-2.13\lib\ext下

　　2、在Test Plan的右侧面板最下方直接添加需要引用的jar包，如下图：



**八、其它用法：**

1、在Test Plan中定义如下三个变量：

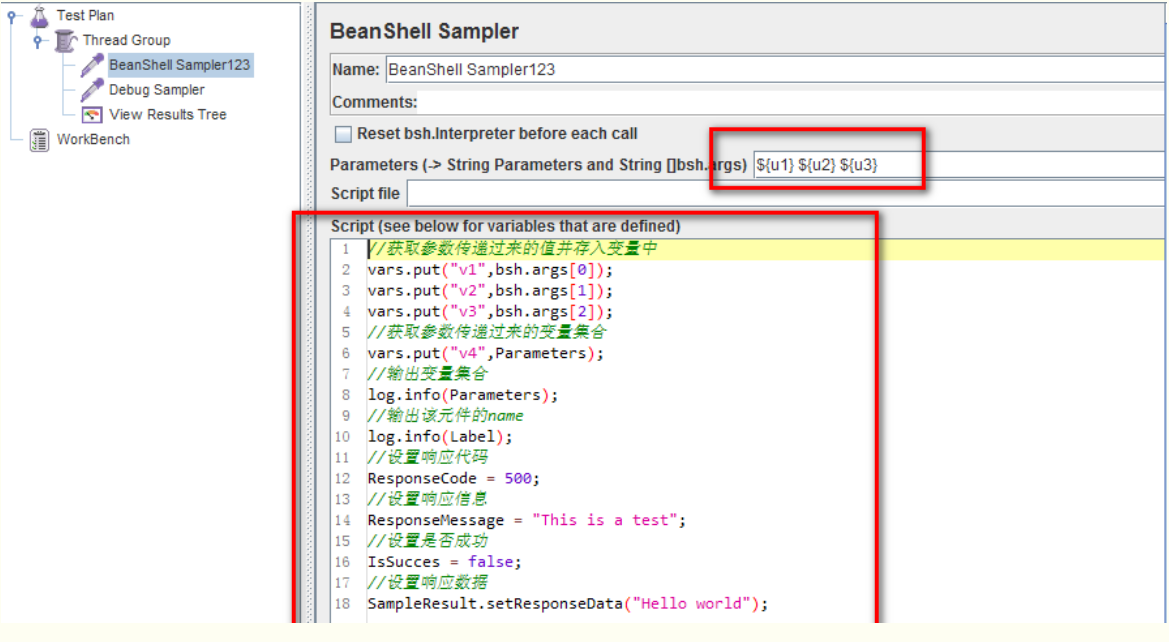


2、Bean Shell可脚本如下：

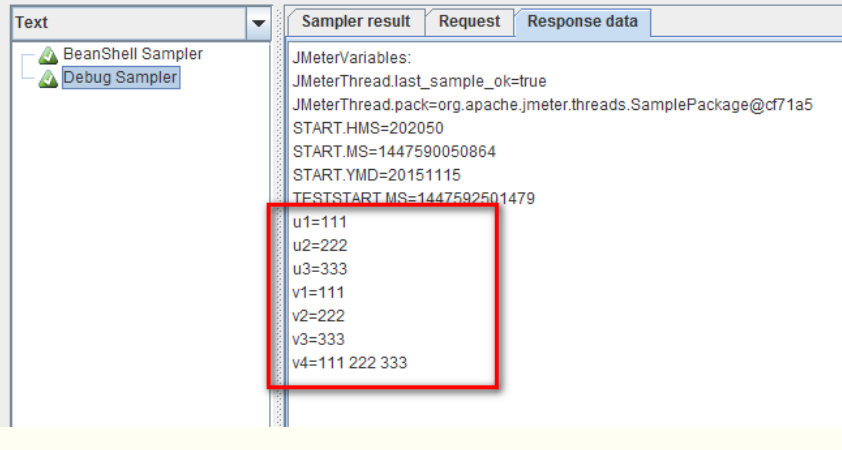
　　a、bean shell可以接受传入参数，如下图：${u1} ${u2} ${u3}，中间要有空格符

　　b、参数可以通过bsh.args[]按顺序提取

　　c、bean shell提供了一个内置变量Parameters，来保存参数的集合



3、运行结果：



下图中1输入的这两句设置：

ResponseCode = 500;  
ResponseMessage = "This is a test";

下图中2输入的这两句设置：

log.info(Parameters);

log.info(Label);

