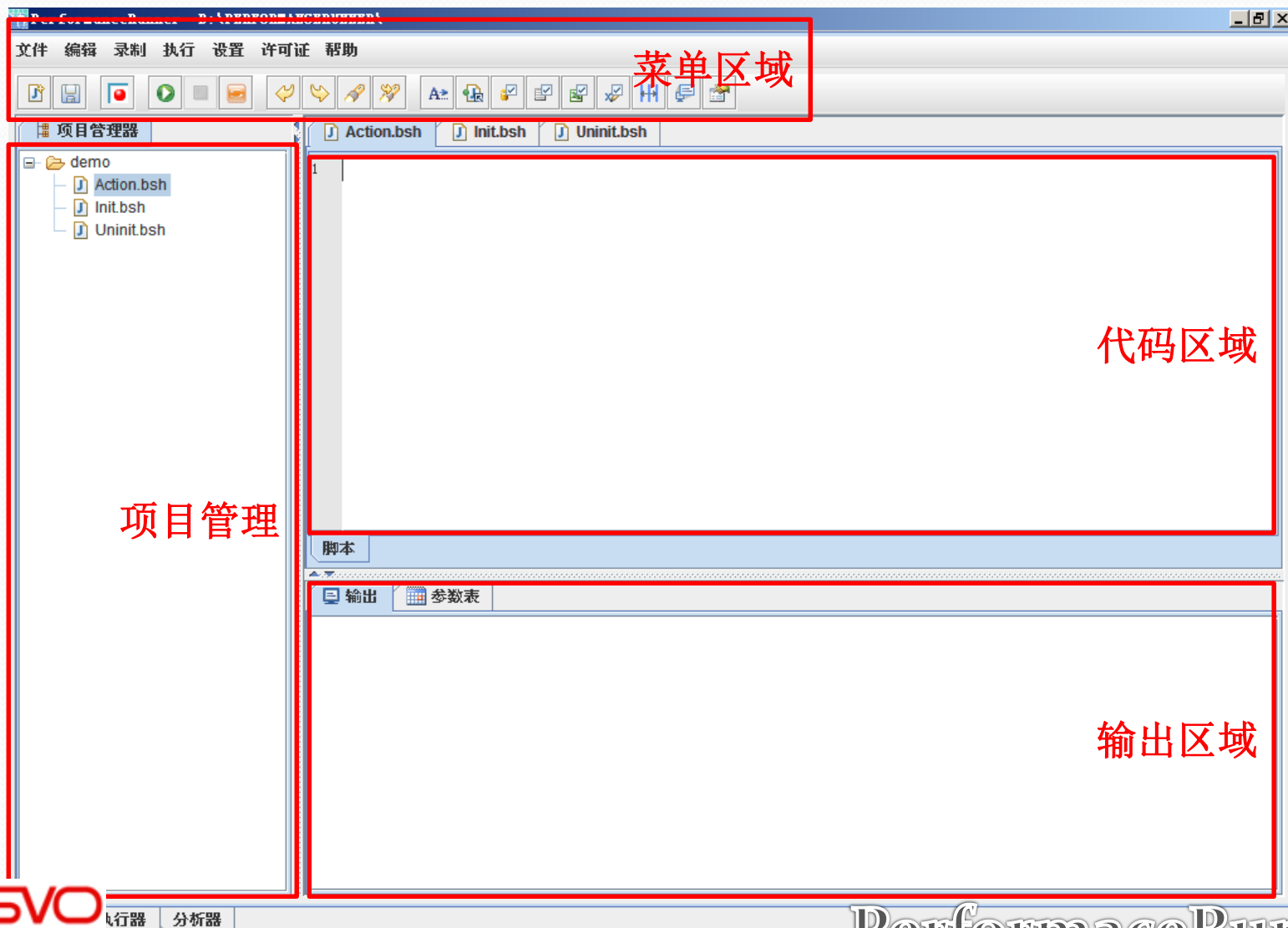


# PR功能简介及脚本加强

# 课程目标

- PR界面及菜单介绍
- 脚本录制及回放
- 关联
- 事务
- 集合点
- 参数化

# PR界面介绍



# PR菜单介绍

文件 编辑 录制 执行 设置 许可证 帮助

文件	编辑	录制	执行	设置	许可证	帮助
<ul style="list-style-type: none"><li>新建</li><li>导入</li><li>保存</li><li>另存为...</li><li>关闭</li><li>关闭全部</li><li>改变工作空间</li><li>最近打开</li></ul>	<ul style="list-style-type: none"><li>撤销</li><li>重做</li><li>查找/替换... Ctrl-F</li><li>查找下一个 F3</li><li>剪切 Ctrl-X</li><li>复制 Ctrl-C</li><li>粘贴 Ctrl-V</li><li>补全代码 Alt-/</li><li>调用脚本</li><li>校验数据库</li><li>校验文件文本</li><li>校验Excel文件</li><li>校验正则表达式</li><li>循环参数表</li><li>打印到输出</li><li>对象库</li></ul>	<ul style="list-style-type: none"><li>开始录制</li></ul>	<ul style="list-style-type: none"><li>开始执行</li><li>停止执行</li><li>关联</li></ul>	<ul style="list-style-type: none"><li>语言</li><li>字号</li><li>网络许可模式</li></ul>	<ul style="list-style-type: none"><li>申请许可</li><li>导入许可</li></ul>	<ul style="list-style-type: none"><li>帮助</li><li>关于</li></ul>

# PR的测试流程

制定测试计划

生成测试脚本

创建场景

运行场景

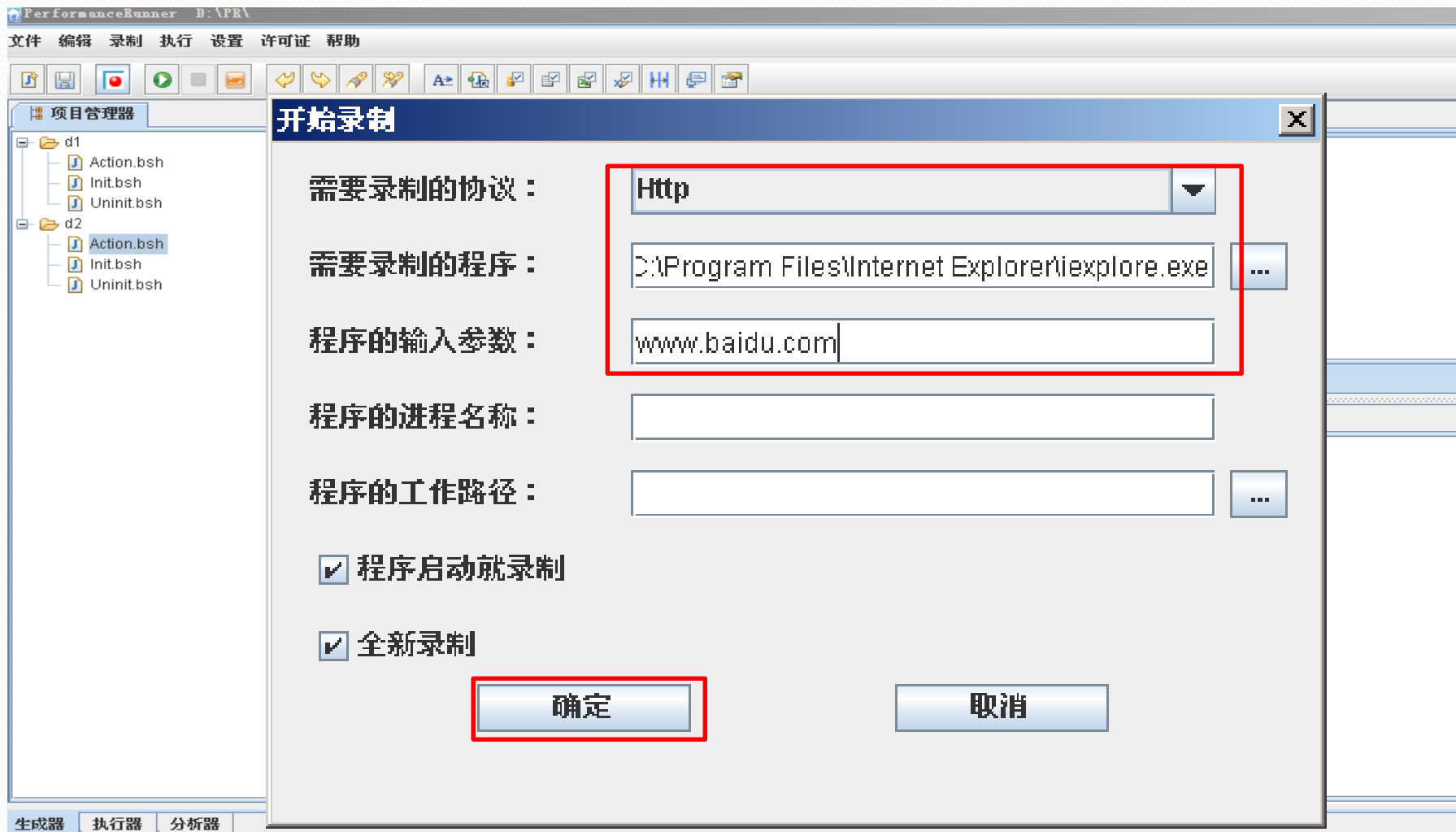
监控测试场景

分析测试结果

# 录制脚本

- 要创建一个自动测试，首先要创建自动测试脚本，成千上万的虚拟用户正是通过执行测试脚本来对系统施压的，录制脚本可以监视并记录客户端和服务端之间的通话，让虚拟用户模拟实际的业务流程，记录真正用户的操作行为，并将其转化为特定的测试脚本语言集合
- 如果想让多个虚拟用户使用几个不同的实际发生数据来执行同一脚本，就需要对脚本进行参数化，参数化后的脚本在运行的时候是使用多套不同的数据项服务器发起请求或执行某个操作，这样就更接近真实的情况

# 录制脚本

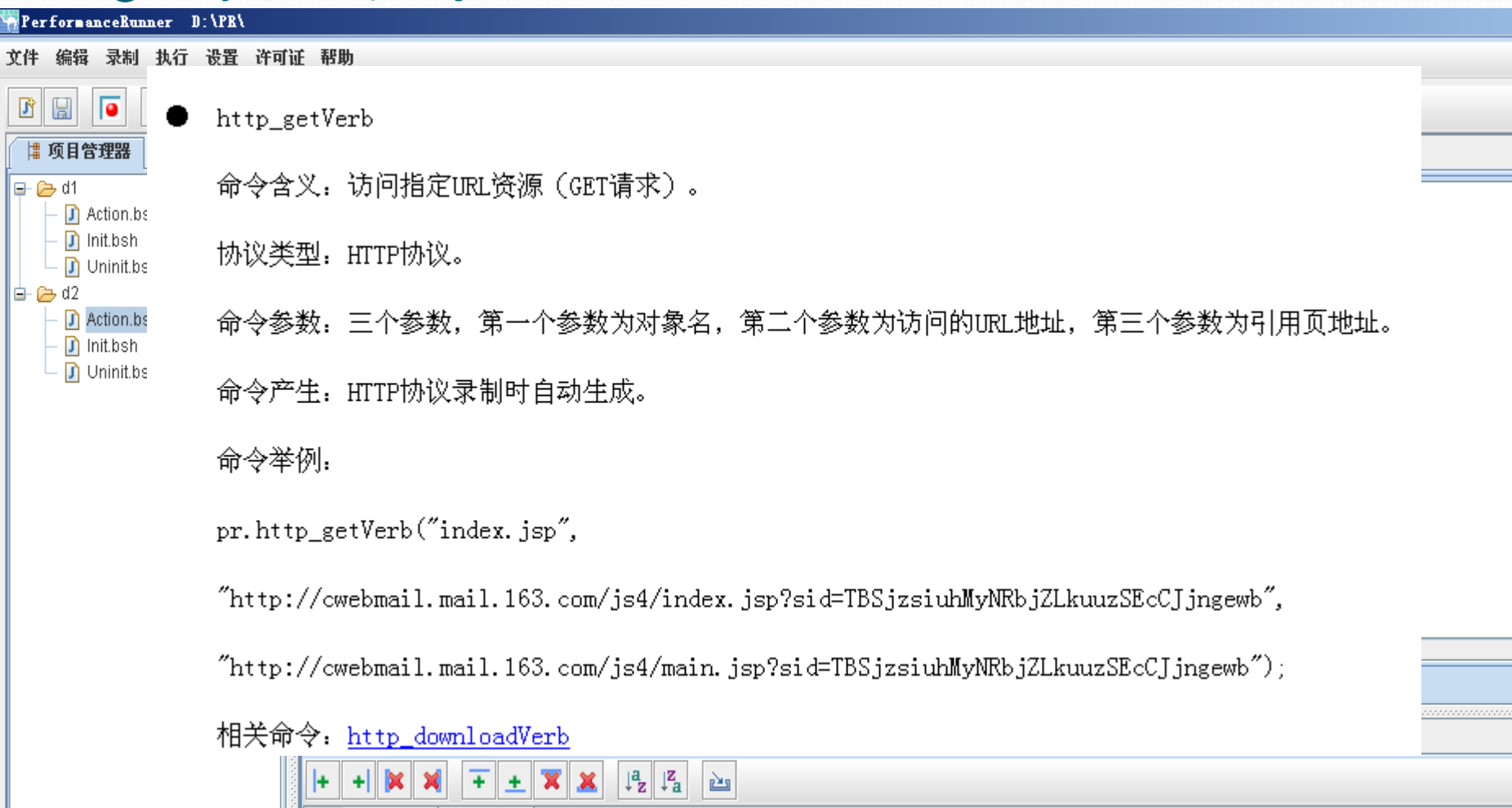


# 录制脚本

- 会出现三个脚本文件，
- **Action.bsh**: 存放实际的操作脚本，可多次执行，可创建多个Action脚本
- **Init.bsh**:一般用于存放应用程序初始化脚本
- **Uninit.bsh**: 一般用于存放应用程序注销或关闭脚本



# 录制脚本



The screenshot shows the PerformanceRunner application window. The title bar reads 'PerformanceRunner D:\PR\'. The menu bar includes '文件', '编辑', '录制', '执行', '设置', '许可证', and '帮助'. On the left is a '项目管理器' (Project Manager) pane showing a tree structure with folders 'd1' and 'd2', each containing 'Action.bs', 'Init.bsh', and 'Uninit.bs'. The main area displays a list of commands, with 'http\_getVerb' selected and highlighted. To the right of the command name, its meaning, protocol type, parameters, generation, and an example are provided. At the bottom is a toolbar with various icons for adding, deleting, and editing commands.

● http\_getVerb

命令含义：访问指定URL资源（GET请求）。

协议类型：HTTP协议。

命令参数：三个参数，第一个参数为对象名，第二个参数为访问的URL地址，第三个参数为引用页地址。

命令产生：HTTP协议录制时自动生成。

命令举例：

```
pr.http_getVerb("index.jsp",  
"http://cwebmail.mail.163.com/js4/index.jsp?sid=TBSjzsiuhMyNRbjZLkuuzSEcCJjngewb",  
"http://cwebmail.mail.163.com/js4/main.jsp?sid=TBSjzsiuhMyNRbjZLkuuzSEcCJjngewb");
```

相关命令：[http\\_downloadVerb](#)

# 事务

- 录制的脚本没有插入事务时，是一个整体，很难分析系统瓶颈是哪些动作导致的，所以我们引入事务
- 我们可以把一个较大的脚本中不同的动作，分成不同的事务，然后进入性能分析，这样在性能分析里边就会把每个事务分别进行分析
- 这样就可以更详细，具体地知道是用户的哪些动作对系统性能影响比较大
- 事务：就是在脚本中定义的某段操作，也可以说是一段脚本语句。

# 定义事务

- 定义事务时，首先在脚本中找到事务的开始和结束位置，然后分别插入起始标记
- 这样，当脚本运行的时候，PR会自动在事务的起始点开始计时，脚本运行到事务的结束点时计时结束，系统会自动记录这段操作的运行时间等性能数据
- 在脚本运行完毕以后，系统会在结果信息中单独反应每个事务的运行结果

# 插入事务代码

- `pr.startTransaction(“t1”);`//开始
- `pr.http_getVerb(“s”,`
- `“http://www.baidu.com/s?wd=”+pr.parameterData.getFrom(“w1”)+“&rsv_bp=0&ch=&tn=baidu&bar=&rsv_spt=3&ie=utf-8&rsv_sug3=5&rsv_sug1=4&rsv_sug4=674&rsv_sug=0&inputT=2437”,`
- `“http://www.baidu.com/”);`
- `pr.endTransaction(“t1”);`//结束

# 集合点

- 假设创建场景时设置了10个虚拟用户同时加载，也就是所谓的10个用户同时并发，那么这个并发过程是不是能够贯穿整个运行过程呢？答案是否定的，客观的讲，这个并发过程仅仅体现在开始执行的一刹那，随着服务器对请求的响应时间的不一致或系统环境条件的限制，10个用户的步调就会马上表现出不一致，有的快些，有的慢一些，在运行的过程中能够集合到一个点上的可能性微乎其微。
- 系统负荷最大的情况是所有用户都集中到系统瓶颈的某个点上进行操作。为了解决这个情况，PR提供了集合点的功能，帮助测试人员实现真正意义上的并发

# 插入集合点

- `pr.rendezvous("search");`
- `pr.startTransaction("t1");`
- `pr.http_getVerb("s",`
- `"http://www.baidu.com/s?wd="+pr.parameterData.getFrom("w1")+"&rsv_bp=0&ch=&tn=baidu&bar=&rsv_spt=3&ie=utf-8&rsv_sug3=5&rsv_sug1=4&rsv_sug4=674&rsv_sug=0&inputT=2437",`
- `"http://www.baidu.com/");`
- `pr.endTransaction("t1");`

# 脚本参数化

```
67     "http://www.baidu.com/");
68 */
69 for (ParameterData pd : pr.getParameterDataList("Action.xls")/*.subList(0, 3)*/)
70 {
71     //pr.parameterData = pd;//pr.parameterData可用于脚本之间传递参数
72 pr.http_getVerb("su_6",
73     "http://suggestion.baidu.com/su?wd="+pr.parameterData.getFrom("w1")+ "&p=3&cb=window.bdsug.sug&sid=&".
74     "http://www.baidu.com/");
75 pr.http_getVerb("su_6",
76     "http://suggestion.baidu.com/su?wd=test&p=3&cb=window.bdsug.sug&sid=&t=1386128709296",
77     "http://www.baidu.com/");
78
79 pr.http_getVerb("s",
80     "http://www.baidu.com/s?wd=test&rsv_bp=0&ch=&tn=baidu&bar=&rsv_spt=3&ie=utf-8&rsv_sug3=5&rsv_sug1=4&".
81     "http://www.baidu.com/");
82
83 pr.http_getVerb("s", "http://www.baidu.com/s?wd=test&rsv_bp=0&ch=&tn=baidu&bar=&rsv_spt=3&ie=utf-8&rsv_sug3=5&rsv_sug1=4&".
```

脚本

输出

参数表

+	+	×	×	+	+	×	×	↓ <sub>a</sub> ↑ <sub>z</sub>	↓ <sub>z</sub> ↑ <sub>a</sub>	↩
0	名称	#0	#1	#2						
0	w1	test	a	b						

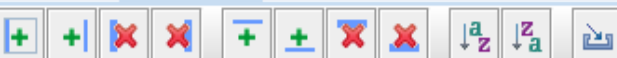


# 脚本 参数化

```
20 for(ParameterData pd : pr.getParameterDataList("Action.xls")/**.subList(0, 4)*/)
21 {
22     pr.parameterData = pd; //pr.parameterData可用于脚本之间传递参数
23     pr.http_postVerb("submit.asp",
24         "http://localhost/submit.asp",
25         "http://localhost/login.asp",
26         "username="+pr.parameterData.getFrom("username")+"&password="+pr.parameterData.getFrom("password")+"&Submit=登 录");
27 }
```

脚本

输出 参数表



	名称	#0	#1	#2
0	username	test	test1	admin
1	password	test	test1	admin



# 小结

- PR界面及菜单介绍
- 脚本录制及回放
- 关联
- 事务
- 集合点
- 参数化