

1 测试环境准备

在进行性能测试前，需要先搭建好测试环境。包含压力机、应用服务器，仿真工具、文件服务器的搭建。

在压力机上安装 loadrunner11.00 版本，在安装和使用 loadrunner 时，需要使用系统管理员权限打开；当 loadrunner 的 license 过期后，需要重新破解即可。

信息系统进行查询交易，当应答报文中包含文件时，这些文件需要存放到文件服务器上，同时需要找开发把信息 router 的通讯地址修改为仿真工具的 IP 地址和文件服务器的 IP 地址。

在性能测试脚本录制前，需要对工行信息中心信息系统的各个子系统做如下修改：

系统	修改内容
信息平台（一代信息）	<ol style="list-style-type: none">1、去掉【登录】时证书验证功能，同时打开客户号和用户名的手动输入功能2、去掉【登录】时验证码验证功能3、去掉【登录】时登录密码验证功能4、增加密码输入超时时间，由原来的 5 分钟修改为 9999 分钟5、去掉交易时证书验证功能6、去掉交易时图形验证功能
信息平台（二代信息）	<ol style="list-style-type: none">1、去掉【登录】时验证码验证功能2、去掉【登录】时登录密码验证功能3、去掉 UKEY 验证功能4、增加密码输入超时时间，由原来的 5 分钟修改为 9999 分钟
移动用户（一代、二代）	<ol style="list-style-type: none">1、去掉【登录】时验证码验证功能2、去掉【登录】时登录密码验证功能3、去掉挑战令牌验证功能4、查询余额时，去掉账号验证功能5、增加密码输入超时时间，由原来的 5 分钟修改为 9999 分钟
微信用户	<ol style="list-style-type: none">1、需要去掉【登录】时【登录密码】验证的功能2、需要去掉交易中的动态口令校验功能3、需要去掉交易中的交易密码校验功能4、“登录 10 分钟后自动退出”功能修改为“登录后不自动退出”5、查询余额时，去掉账号验证功能

2 测试数据准备

目前信息系统的数据准备有三种方式，分别为：由开发人员在数据库中获取存量数据，开发提供 SQL 脚本由测试人员导入，测试人员提供数据由开发人员导入。

在使用由开发人员在数据库中获取存量数据的方法时，获取的数据可能存在问题（如缺少字段、同一用户操作不同账号、同一账号不同操作员使用等问题），导致一部分数据无法使用，不建议使用这种方法。

开发提供 SQL 脚本由测试人员导入和测试人员提供数据由开发人员导入这两种方式，都需要由测试人员准备测试数据，不同的是由开发人员批量生成 SQL 脚本导入，还是由测试人员批量生成 SQL 脚本导入（需要由开发人员提供 SQL 模板）。

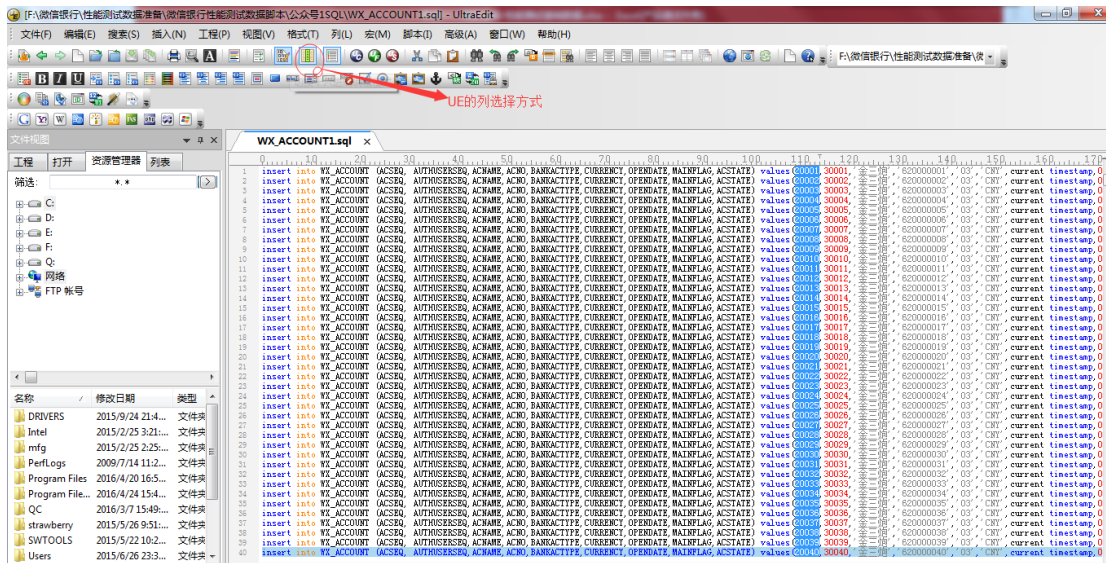
各个系统需要的数据如下：

系统	需要准备的数据
公共数据	1、已开通信息的用户 100 家 2、文件服务器、压力机、性能模拟器的存放 IP 地址
信息平台（一代信息）	1、个人用户登录名 3000 个（每家用户 30 个）、每个用户加挂借记卡账号 1 个、加挂信用卡账号 1 个 2、企业用户客户号、企业操作员用户名各 3000 个（每家用户 30 个）、每个操作员加挂账号 2 个 3、一代个人信息、一代企业信息的 web 访问地址
信息平台（二代信息）	1、个人用户登录名 3000 个（每家用户 30 个）、每个用户加挂借记卡账号 1 个、加挂信用卡账号 1 个 2、企业用户客户号、企业操作员用户名各 3000 个（每家用户 30 个）、每个操作员加挂账号 2 个 3、二代个人信息、二代企业信息的 web 访问地址
移动用户（一代、二代）	1、一代移动用户个人用户登录名 3000 个（每家用户 30 个）、每个用户加挂借记卡账号 1 个、加挂信用卡账号 1 个 2、二代移动用户个人用户登录名 3000 个（每家用户 30 个）、每个用户加挂借记卡账号 1 个、加挂信用卡账号 1 个 3、一代移动用户、二代移动用户的 web 访问地址
微信用户	1、2 个用户的微信公众号 2、2 个公众号从 gateway 服务器获取微信链接的链接地址 3、每个公众号提供 2000 个认证用户、每个用户各加挂 1 张借记卡和 1 张信用卡

信息系统数据准备的模板如下：

生成 sql 脚本的方法有两种方式：

a) 在 UE 中通过列替换的方式，具体操作如下选择要替换的数据进行复制粘贴。



b) 在 excel 中通过函数的方式：在 excel 表格中把各个字段按列表的方式列出，然后通过 CONCATENATE 函数替换对应字段的值,进行下拉。

如：=CONCATENATE("insert into PACCOUNT (ACSEQ, BANKSEQ, CIFSEQ, ACNO, BANKACTYPE, ACORDER,ACNAME, CURRENCY, CRFLAG, ACPERMIT, OPENDATE, CLOSEDATE, DEPTSEQ, ACSTALE, AALIAS) values ('A1','B1','C1','D1','E1','F1','G1','H1','I1','J1','K1','L1','M1','N1','O1','");")

生成结果如下图



3 脚本准备

3.1 脚本录制

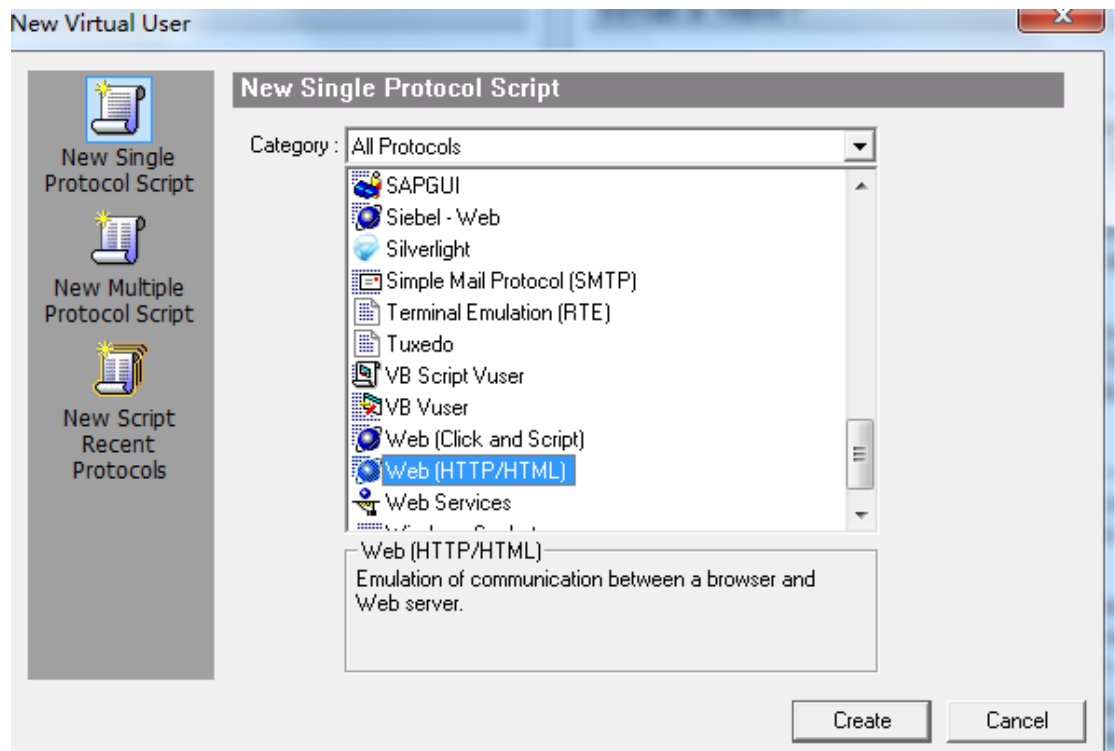
启动 Visual User Generator, 创建一个新的脚本, 开始录制脚本, VuGen 会自动捕获操作过程中 client 端与 server 端进行通信的所有数据, 根据录制时选择的协议, 生成测试脚本。

在脚本录制的过程中, 需要注意以下几个方面: 脚本协议的选择和 Recording Options 设置

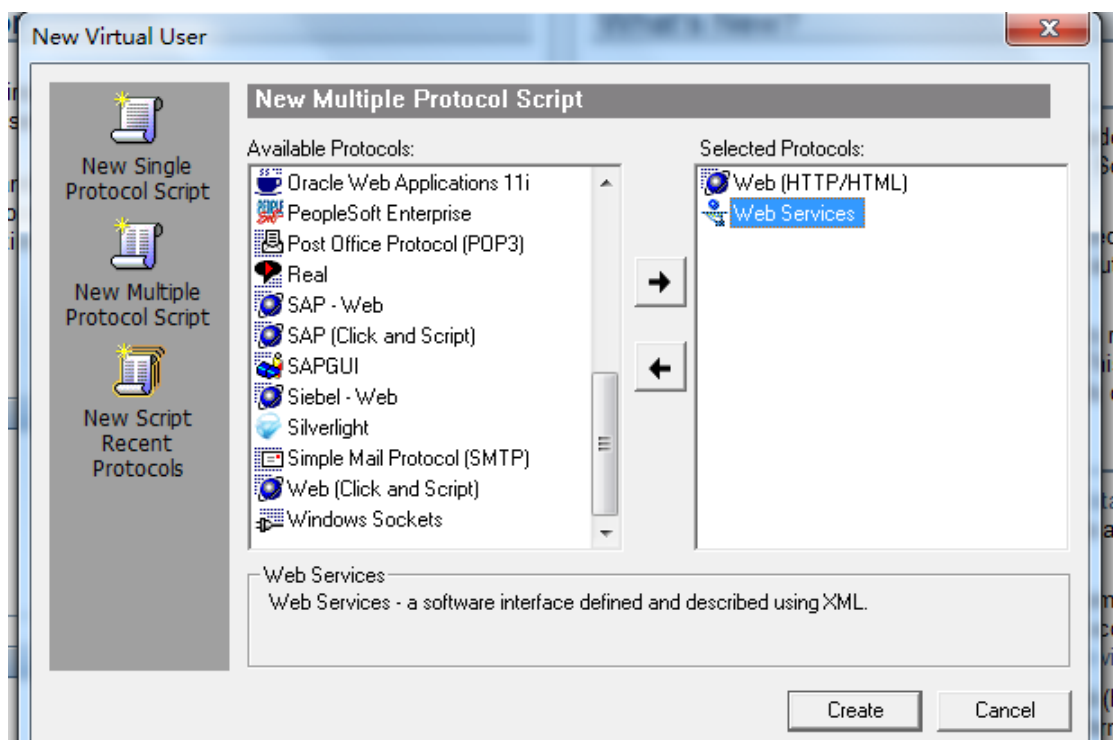
3.1.1 脚本协议的选择

在创建新脚本时, 会弹出一个对话框, loadrunner 提供了三种选择协议的方式, 分别为单协议脚本、多协议脚本和最近使用过的协议。

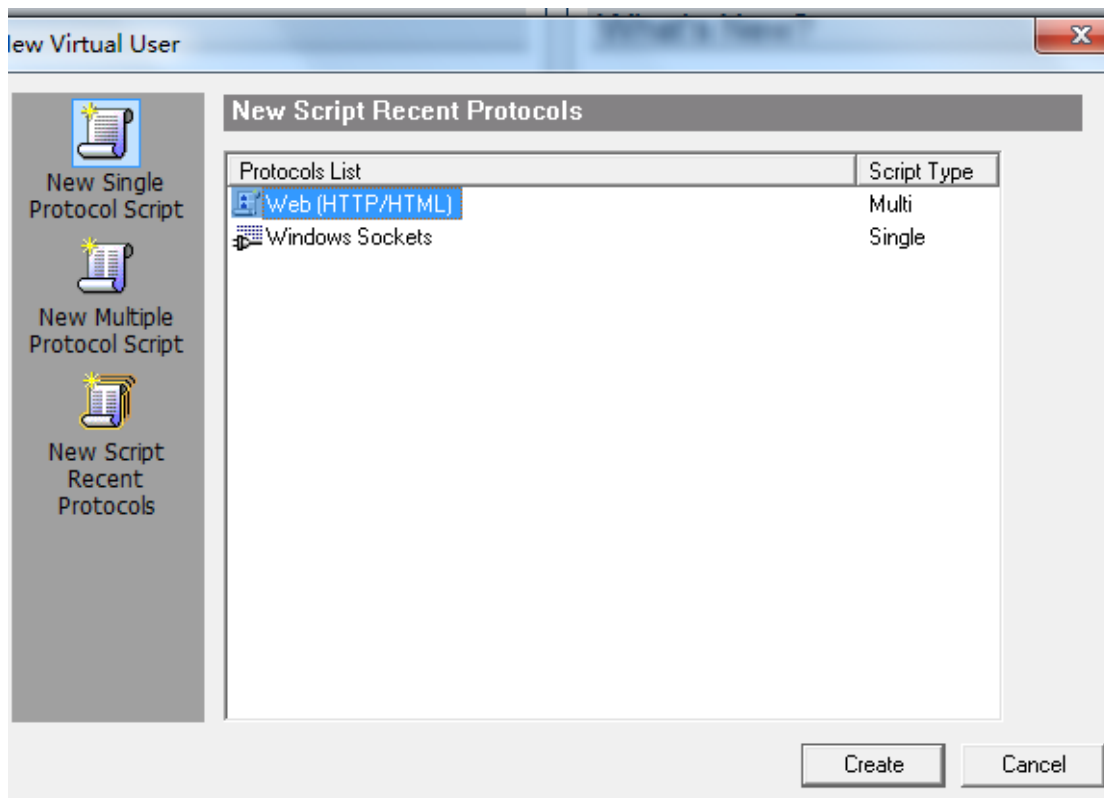
a) 创建单协议脚本, 在弹出的对话框中选择录制时需要的协议后, 点击 Create。如下图:



- b) 创建多协议脚本，在 Available Protocols 中选择一个或多个协议，点击右箭头，将其移入到 Selected Protocols 部分中，同样，在 Selected Protocols 中选择一个或多个协议，点击左箭头可以移除选中的协议。确认协议后，点击 Create 按钮。如下图：



- c) 最近使用过的协议：从最近录制脚本的协议列表中，选择一种协议进行录制，如下图



3.1.2 开始录制脚本

协议选择好后可以开始录制脚本，下面以 WEB 协议为例进行说明。
录制时系统弹出一个录制窗口，如图：



Application type 中选择应用程序类型:Internet Application

URL 中输入要测试的网址

Working directory 中设置工作目录

Record into Action 中选择要把录制的脚本放到哪一个部分

VuGen 中的脚本分为三部分: vuser_init、vuser_end 和 Action。其中 vuser_init 和 vuser_end 都只能存在一个，不能再分割，而 Action 还可以分成无数多个部分（通过点击 New 按钮，新建 ActionXXX）

“Record the application startup”默认情况下是选中的，说明应用程序一旦启动，VuGen 就会开始录制脚本；如果没有选中，应用程序启动后，VuGen 出现下面的对话框，并且暂时不会开始录制脚本，用户操作应用程序到需要录制的地方，按下“Record”按钮，VuGen

才开始录制。



开始录制后，在屏幕上会有一个工具条出现。下面我们简单介绍一下各个按钮的功能：



录制完成后，按下“结束录制”按钮，VuGen 自动生成用户脚本，退出录制过程。

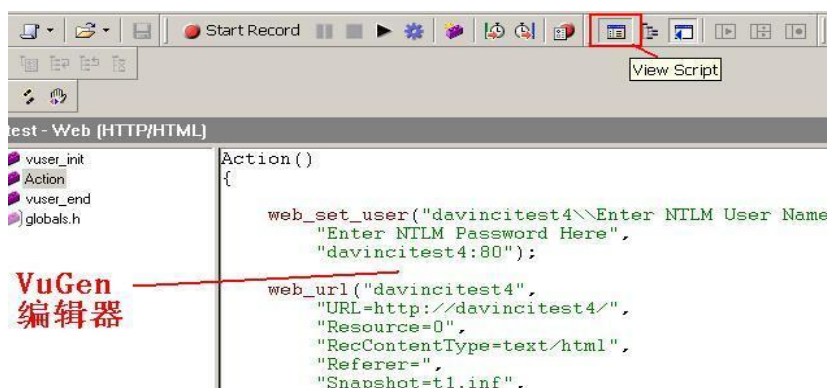
3.1.3 查看脚本

录制结束后，可以通过两种方式来查看脚本：脚本视图及树视图

a) 脚本视图

脚本视图是一种基于文本的视图，列出了作为 API 函数的 Vuser 操作。要在脚本视图中查看脚本，请单击工具栏上的“View Script”按钮或者选择“View” > “View Script”。

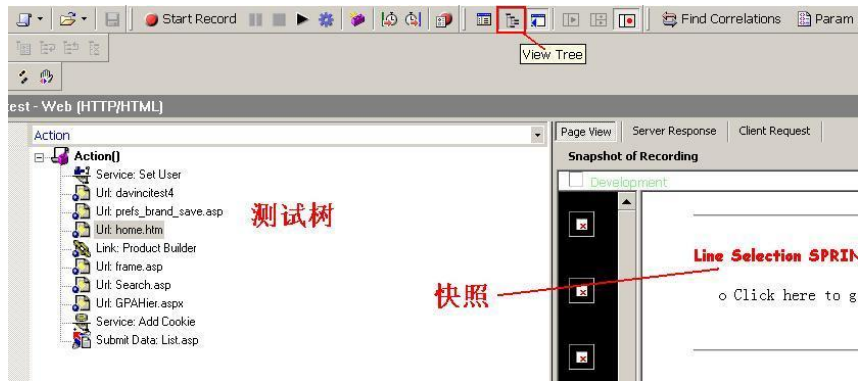
在脚本视图中，VuGen 将在编辑器中显示带有彩色编码的函数及其变量值的脚本。



b) 树视图

树视图是基于图标的视图，列出了作为步骤的 Vuser 操作。要在树视图中查看脚本，请单击工具栏上的“View Tree”按钮或者选择“View” > “View Tree”。

对于录制期间所执行的每一步骤，VuGen 都在测试树中生成一个图标和一个标题。在树视图中，将看到作为脚本步骤的用户操作。大多数步骤都附带相应的录制快照。



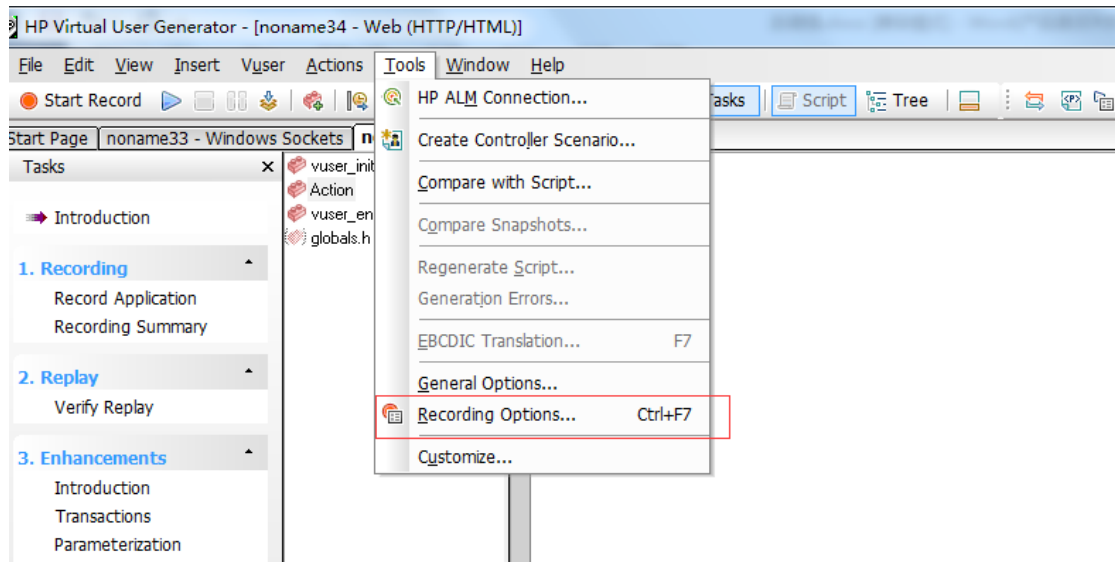
3.2 Recording Options 设置:

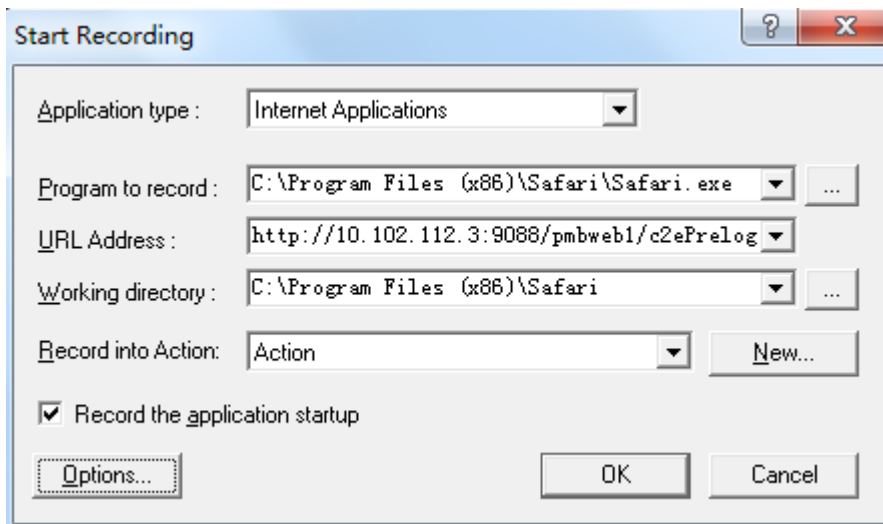
在录制脚本时，首先要对录制的一些参数进行设置，只有这些参数设置好，才能录制并生成需要的脚本

在 Recording Options 设置中，需要注意的设置项有三个选项卡，分别为 Recording 选项卡、Advanced 选项卡、Correlation 选项卡

3.2.1 Recording 选项卡

在 Tools 菜单中选择 Recording Options 或在 Stat Recording 界面点击 Options 按钮，可以打开 Recording Options 设置。如下图





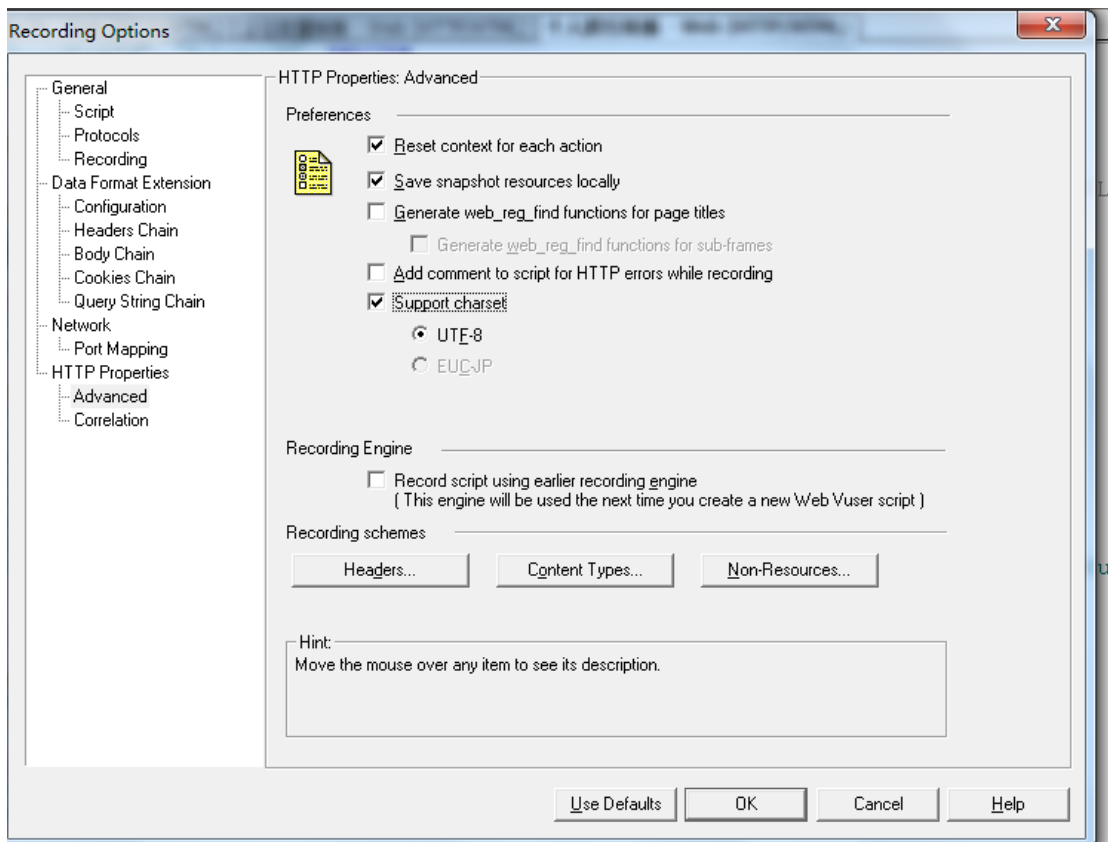
在 Recording Options 设置中，选择 Recording 选项卡，Recording Level 中有两种录制模式：HTML-based script 和 URL-based script。

HTML-based script 方式录制的脚本，对每个页面的录制成一条语句。

URL-based script 方式录制的脚本，将客户端发出的每个请求录制成一条语句。

3.2.2 Advanced 选项卡

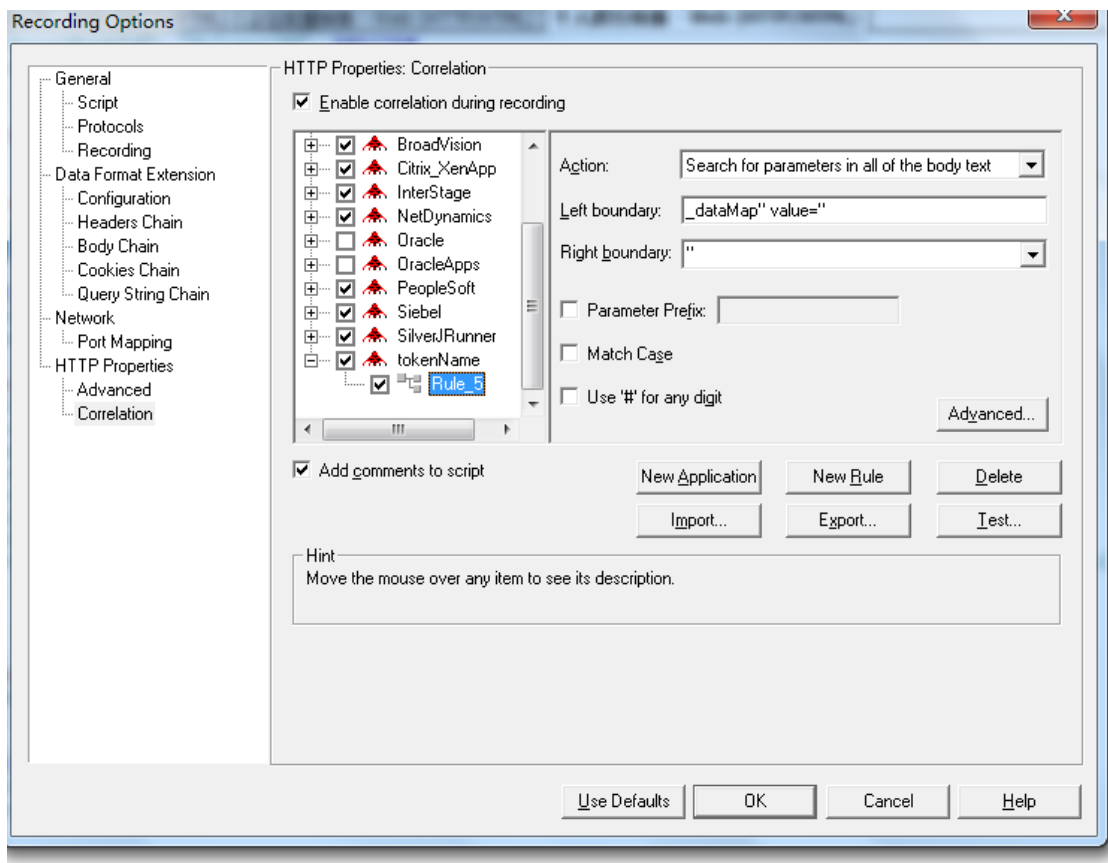
Advanced 选项卡，当录制的脚本中存在乱码时，设置 Advanced 选项卡中 Support charset 的格式为 UTF-8，可以解决。



3.2.3 Correlation 选项卡

Correlation 选项卡用来对脚本中的关联属性进行设置，选中需要的关联规则，在录制脚本时会自动进行关联。如果当前的这些关联规则，无法满足录制的要求，可以点击 NEW

APPLICATION 按钮来新建一个关联，在点击 new rule 按钮为该关联新建一个规则。



3.3 完善/编辑测试脚本

脚本录制完成并不代表这些脚本很接近用户的真实使用情况，为了使这些脚本更接近用户的真实使用情况，需要对脚本进行完善。

脚本的完善工作包含插入事务、插入集合点、插入检查点、参数化、设置关联等，下面对各个方面进行描述。

3.3.1 插入事务点

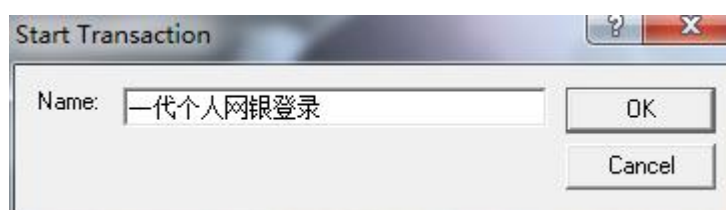
插入事务点有两种方法：一种是在脚本录制过程中插入开始和结束事务点；另一种是在编辑脚本时插入开始和结束事务点。一般情况下，选择在脚本录制过程中插入开始和结束事务点。这样做有两个优点，第一、保证不会遗漏需要插入的事务点，第二、避免在脚本录制完成后找不到确切的插入事务点的地方。

a) 方法一：在脚本录制过程中插入开始和结束事务点

当脚本录制到需要插入事务点时，在录制工具栏点击插入开始事务点按钮，如图



在弹出的开始事务对话框中输入开始事务名称，如图。事务点名称的格式应保持一致，使其具有代表性，遵守脚本命名规则



事务的开始点和事务的结束点具有一一对应性，即在插入了一个事务开始点后，一定要在接下来的过程中插入事务结束点，插入事务结束点和插入事务开始点的操作一致，当该业务流程执行完毕后添加事务结束点，在录制工具中点击插入结束事务点的按钮，如图

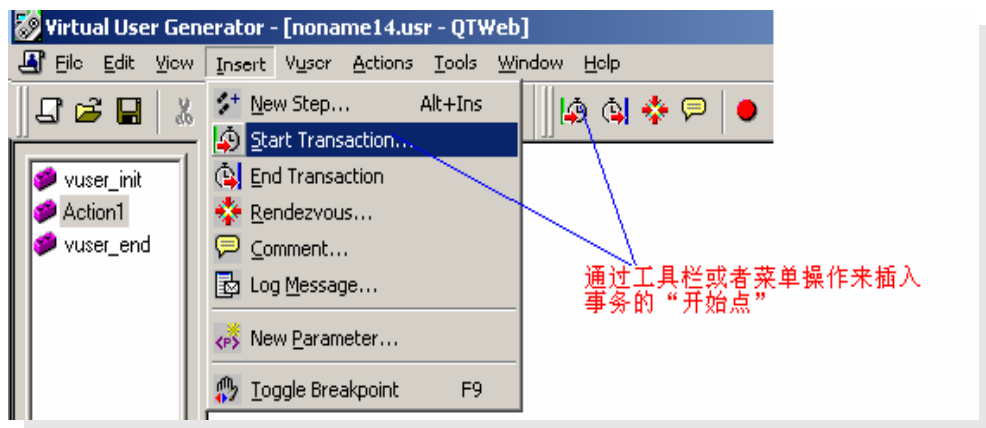


在弹出的对话框中输入选择对应的事务结束点名称。如图

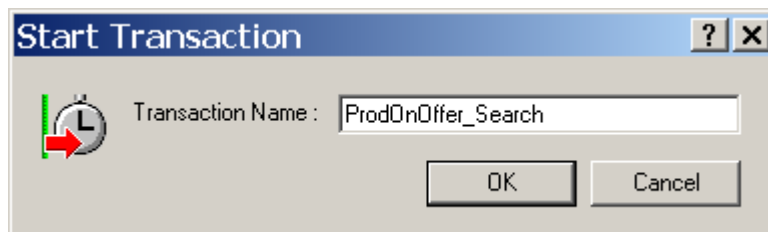


b) 方法二：在编辑脚本时插入开始和结束事务点

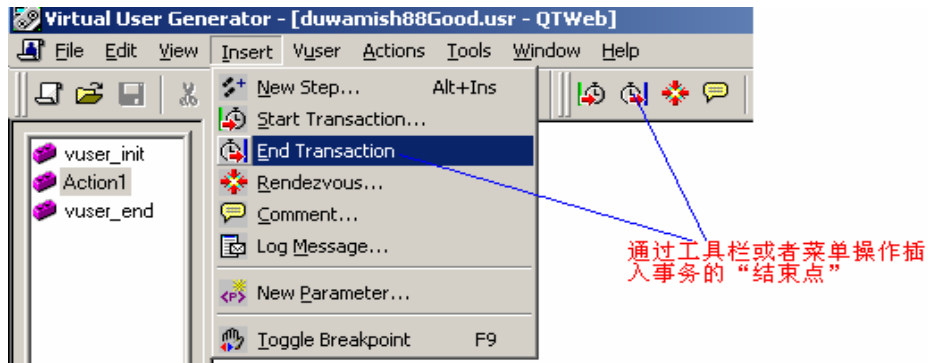
在需要定义事务的操作前面插入事务的“开始点”，通过菜单或者工具栏插入。如图



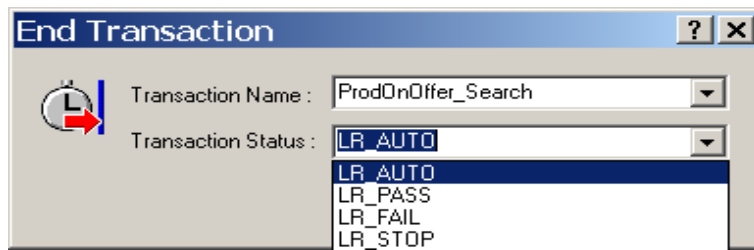
输入事务的名称。注意：事务的名称最好要有意义，能够清楚的说明该事务完成的动作



插入事务的开始点后，在需要定义事务的操作后面插入事务的“结束点”。同样可以通过菜单或者工具栏插入。



默认情况下，事务的名称列出最近的一个事务名称。一般情况下，事务名称不用修改。事务的状态默认情况下是 LR_AUTO。一般情况下，我们也不需要修改



3.3.2 插入集合点

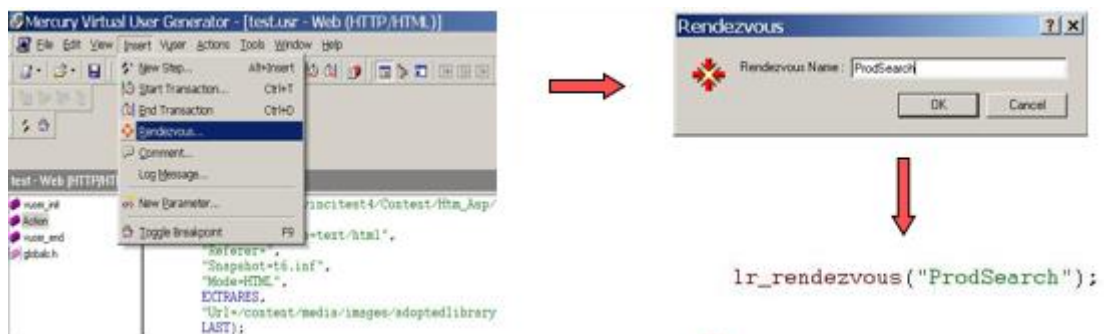
集合点是为了衡量在加重负载的情况下服务器的性能情况。

例如：在测试计划中，可能会要求系统能够承受 1000 人同时提交数据，在 LoadRunner 中可以通过在提交数据操作前面加入集合点，这样当虚拟用户运行到提交数据的集合点时，LoadRunner 就会检查同时有多少用户运行到集合点，如果不到 1000 人，LoadRunner 就会命令已经到集合点的用户在此等待，当在集合点等待的用户达到 1000 人时，LoadRunner 命令 1000 人同时去提交数据，从而达到测试计划中的需求。

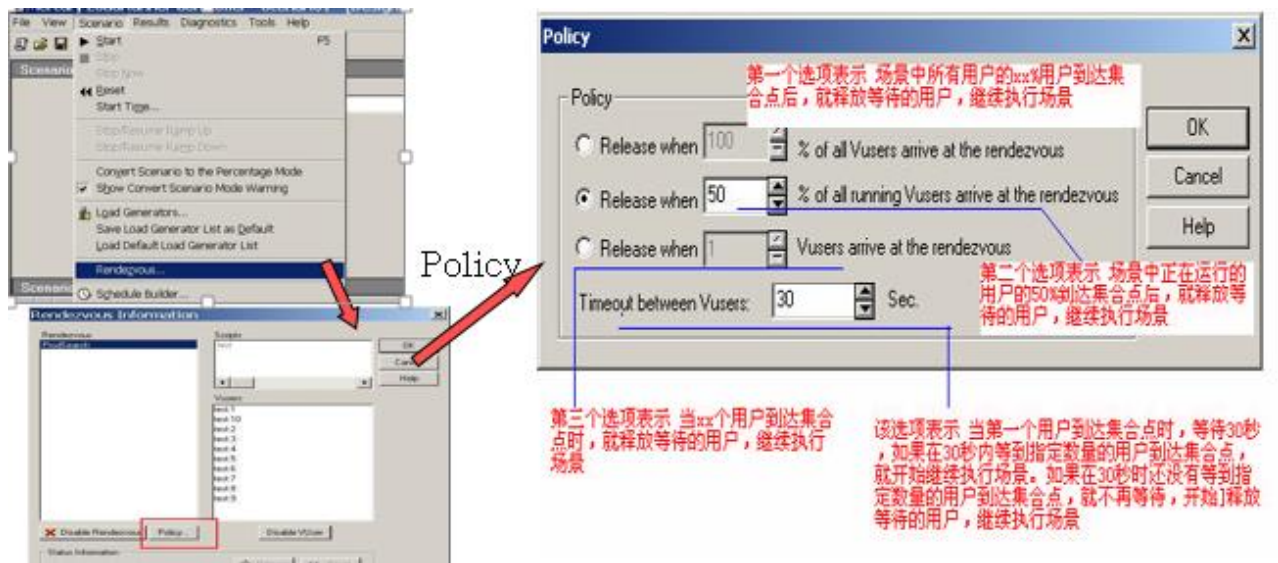
注意：集合点经常和事务结合起来使用。集合点只能插入到 Action 部分，vuser_init 和 vuser_end 中不能插入集合点。集合点需要插入到事务前，否则事务的响应时间会变长。

插入集合点的步骤如下

- 1、在需要插入集合点的前面，选择菜单 Insert->Rendezvous
- 2、在弹出的 Rendezvous 对话框中，输入要插入的集合点的名称，点击 OK 后，集合点插入完毕，集合点在脚本中的代码如下：lr_rendezvous(“名称”)



如果在脚本中设置了集合点，还需要在 Controller 中设置集合点策略，从菜单中调出集合点策略设置窗口，根据需求设置集合点策略。具体设置方式如下图

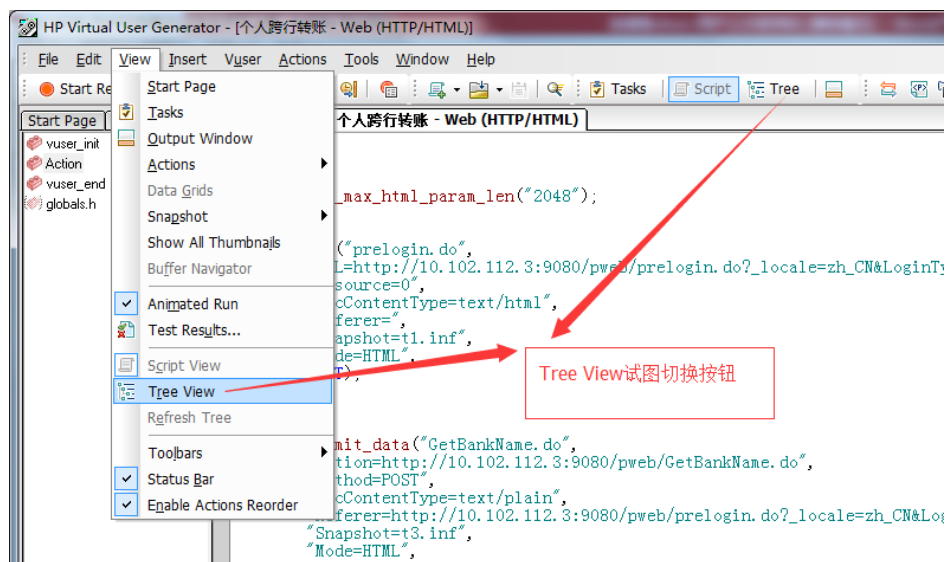


3.3.3 检查点

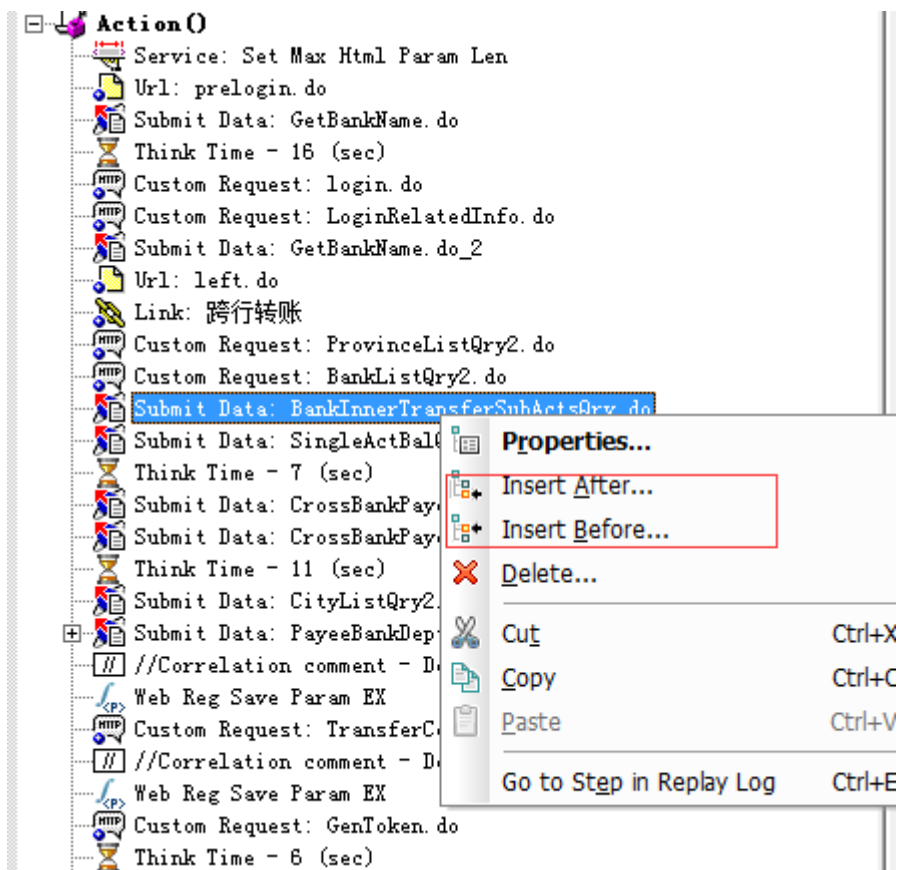
检查点是通过检查点函数将返回值的结果反映在 Controller 的状态面板上和 Analystist 统计结果中。检查点的目的是检查 Web 服务器返回的网页是否正确。例如：检查点函数 web_find 函数，如果他检查到的结果为空，它的返回值为 LR_FALL，这样整个结果置为 FALL；反之，检查到的结果为成功，则 web_find 返回的结果为 LR_PASS，整个结果置为 TRUE。

插入检查点的方法，是在 VuGen 中插入 Text/Image 检查点，插入检查点的步骤如下：

- 1) 将视图模式设置为 Tree View，VuGen 中的脚本包含 Tree View 和 Script View 两种视图模式，一般情况下都是使用 Script View，但在插入检查点时，一般将视图模式切换到 Tree View，这样方便找到要插入检查点的页面，可以保证插入检查点的位置正确。



- 2) 在 Tree View 中选择要插入检查点的位置
选中要插入检查点的位置后，点击鼠标右键，选择插入检查点的方式，可以选择插在前面，也可以选在插在后面。如下图



3) 选择检查点的类型和插入函数

插入检查点有文本检查点和图片检查点两种，选择的插入检查点的函数也有两个，分别为 `web_find` 和 `web_reg_find`。

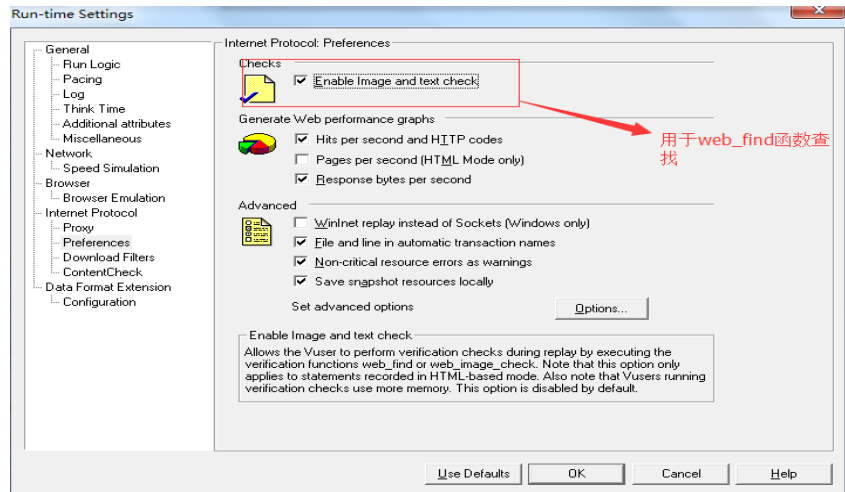
a) `Web_find` 函数

该函数作用是在页面中查找相应的内容。常用的参数含义如下：

```
Web_find("test", //检查点步骤名称
        "RightOf=", //定义查找字符串的右边界
        "LeftOf=", //定义查找字符串的左边界
        "What=", //定义检查字符串的内容
        LAST);
```

使用该函数需要注意以下几个方面

- 1、该函数只能对基于 **HTML** 模式录制的脚本进行查找
- 2、该函数必须在页面内容显示出来后才能进行查找，所以该函数必须写在查找内容所在页面后面
- 3、必须启用内容检查项，在 **Run-Time setting** 下的 **Preferences** 里面，把 **Enable Image and text check** 复选框选中，否则不执行该查找函数。如下图：



4、在 VB 和 JAVA 语法中不能使用该函数

该函数存在下面两个缺点：

- 1、执行效率较低
- 2、不返回查找结果情况。例如，要查看有多少虚拟用户登录成功的情况，这个函数无法做到，必须进一步操作才能实现

b) Web_reg_find 函数

该函数是在缓存中查找响应的内容，是一个注册函数，常用的参数及含义如下：

```
Web_reg_find("Text=", //定义要查找的内容
            "SaveCount=textCount", //定义查找计数变量名
            "Search=Body", //定义查找的范围
            LAST);
```

该函数必须写在要查找内容的请求之前，SaveCount 参数用来记录在缓存中内容被查找到的次数，进而来判断要查找的内容是否真的被查找找到。

Web_find 和 web_reg_find 虽然都是检查点函数，但两个函数还是有区别的，主要区别如下：

- 1、两个函数类型不同，web_find 只是一个普通函数，而 web_reg_find 是一个注册函数。
- 2、Web_find 函数在使用时必须开启内容检查选项，而 web_reg_find 函数没有此限制
- 3、Web_find 函数录制时只能基于 HTML 模式录制的脚本中，而 web_reg_find 函数没有此限制
- 4、Web_find 函数是在返回的页面中进行查找，web_reg_find 函数是在缓存中进行查找
- 5、Web_reg_find 函数在执行效率上要高于 web_find 函数

3.3.4 参数化

脚本参数化，就是针对脚本中的某些常量，使用参数来取代。参数中包含了很多数据源，数据源可以是一个文本文件也可以是数据库，当不同的 vuser 在执行相同脚本时，分别调用参数文件中的数据代替这些常量，从而达到模拟多用户真实使用的目的。

使用参数化，具有以下优点：

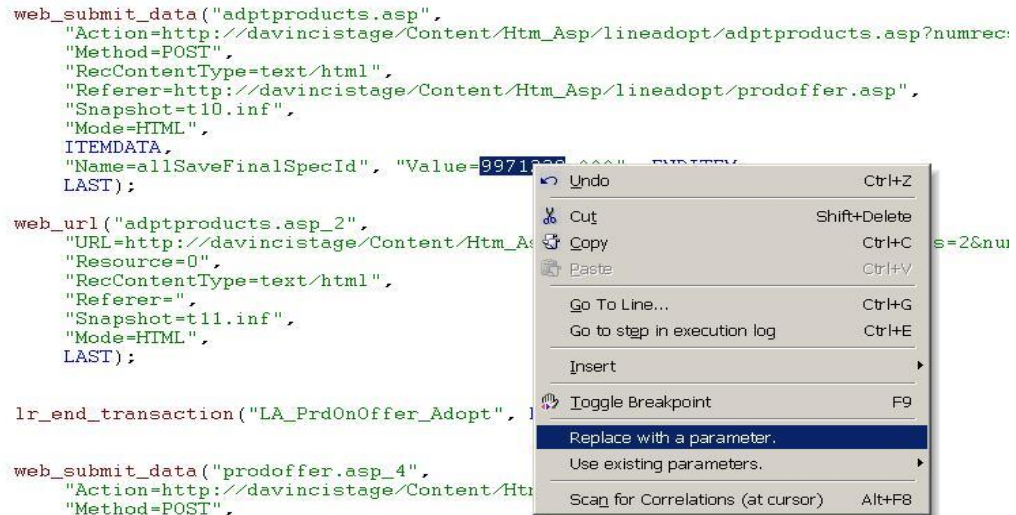
- 1、借助参数化可以减少脚本的数量，如果不进行参数化，为了达到目标，可能需要

拷贝并修改很多脚本。

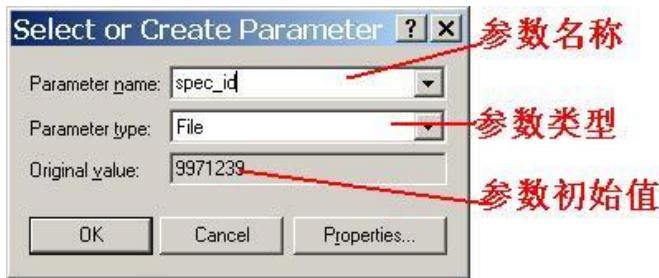
- 2、使业务更接近真实的客户业务，每个虚拟用户使用不同的参数值来模拟，这样可以更好的模拟真实用户

设置参数化的步骤如下：

- 1、在录制好的脚本中，选中要参数化的常量，点击鼠标右键，选中 **Replace with a parameter**



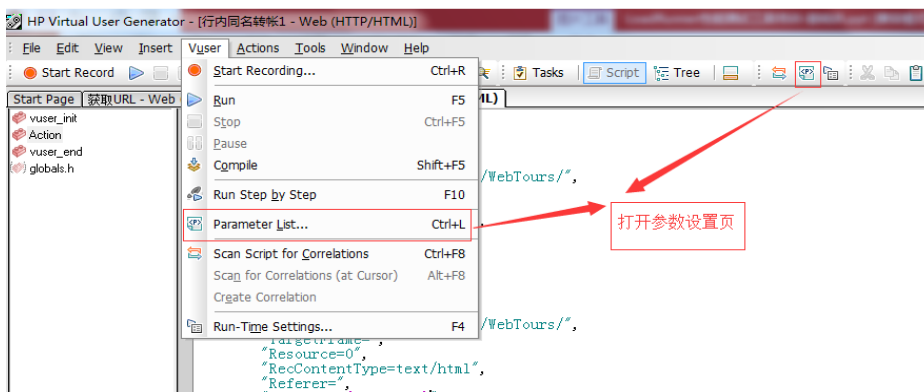
- 2、在弹出的 **select orcreate parameter** 对话框中设置参数的名称和参数类型，如下图



Parameter type 为参数类型，主要包含 File、Iteration Number、Unique Number、Group Name、User ID、Random Number、DateTime 等参数类型。

其中 **File** 参数类型，可以在参数编辑中编辑参数文件，也可以直接选择已编辑好的参数文件，还可以从现成的数据库中提取，这是最常用的一种参数化方式

- 3、通过 **vuser->parameter list** 或工具栏的 **open parameter list** 按钮，可以打开 parameter list 对话框。如下图



4、parameter list 设置

a) Browse 设置

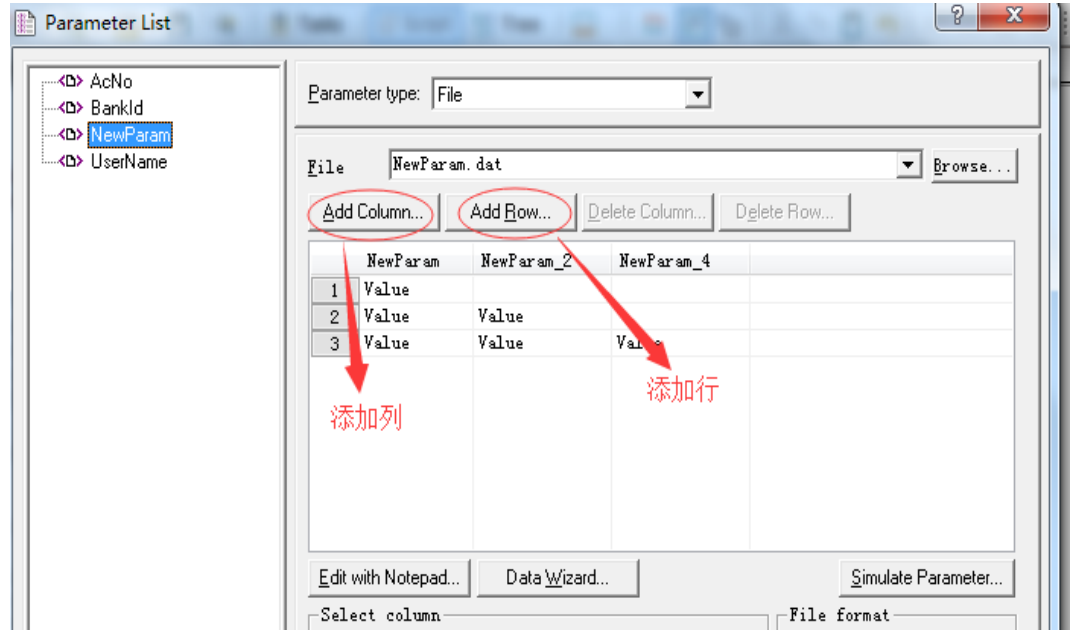
该设置用来选择参数文件的路径，进行参数化时，需要选择正确的路径，否则无法读取到参数文件中的参数



b) Edit with notepad 设置

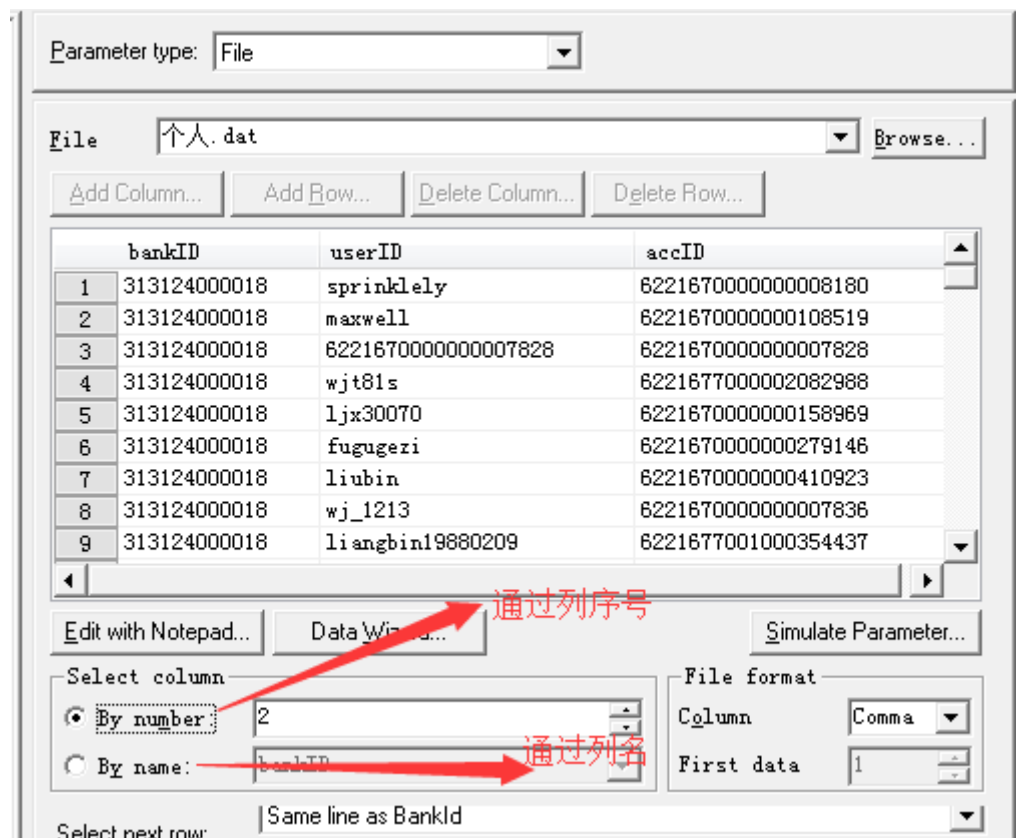
点击 edit with notepad 按钮，打开记事本，记事本内容的第一行为参数名称，第二行为参数的初始值，参数之间使用逗号、tab、space 等隔开。可以在记事本中对参数进行修改、添加和删除操作。如果再记事本没打开的情况下可以在参数属性对话框中点击 Add Column 进行添加列，点击 Add row 添加行。

注意:编辑文本时,需要在最后一行增加行空行,否则文本中最后一行数据无法读取



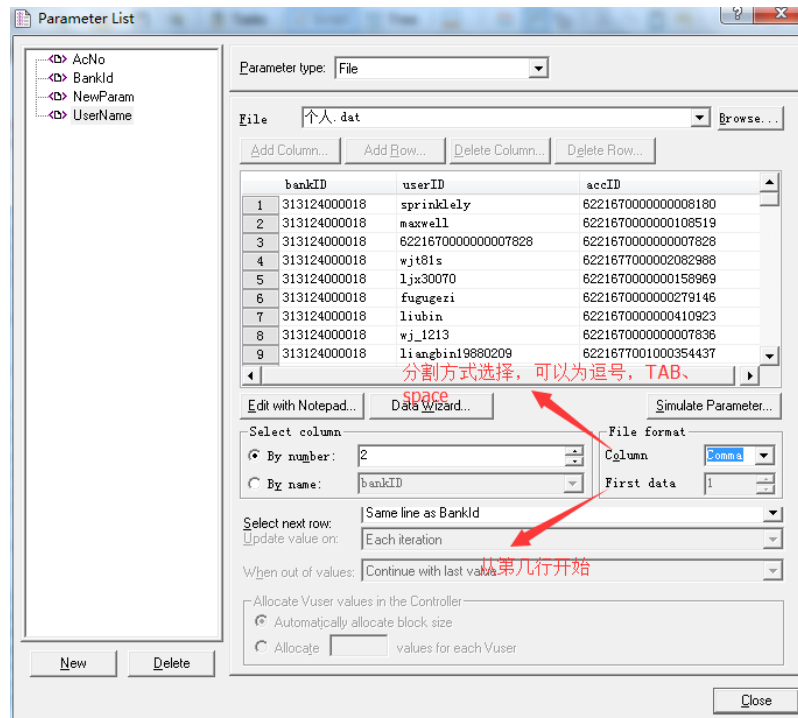
c) Select column 设置

指明参数选择的列。这里有两种方式可以进行选择，一种方式可以通过列名来选择，一种方式可以通过列的序号来进行选择



d) File format 设置

File format 设置用于设置列分割符以及从第几号开始获取数据



e) Select next row 设置

Select next row: 定义了 Vuser 如何从参数文件中选取下一行数据，有四种方式分别为 Sequential、Unique、Random、Same Line as ***。

Sequential (顺序): 虚拟用户 Vuser 按照行顺序读取文件中的数据，如果文件中的数据都执行了一遍，则返回第一行，继续执行

Unique (唯一): 强调虚拟用户的差异性。为每个 Vuser 分配一个唯一值；注意：使用该类型必须注意数据表有足够多的数。比如 Controller 中设定 20 个虚拟用户进行 5 次循环，那么编号为 1 的虚拟用户取前 5 个数，编号为 2 的虚拟用户取 6-10 的数，依次类推，这样数据表中至少要有 100 个数否则 Controller 运行过程中会返回一个错误。

Random (随机): 每个 Vuser 从表中随机的读取一个参数数据，假设有 50 条数据，那么随机数将在 1-50 之间随机取一个，然后用这个数做为行号，从表中读取相应的数据。

Same Line as *:** 与某个已定义好的参数取同一行值。注意：该方法要求至少其中的一个参数必须是 Sequential、Unique 或 Random

f) Update value on 设置

Update value on 设置脚本迭代过程中取值的策略，有 3 中方式的选择，分别为 Each Iteration、Each Occurrence、Once。

Each Iteration: 脚本每迭代一次都访问数据表中的下一个值，在同一次迭代的过程中，不管同以参数出现多少次，都只使用这一个参数。

Each Occurrence: 参数每次出现时更新，即使在同一次迭代的过程中，某个参数使用了多次，每次使用时都进行更新，而不会使用同一个参数。

Once: 同一参数在脚本执行的过程中均取同一个值。

3.3.5 设置关联

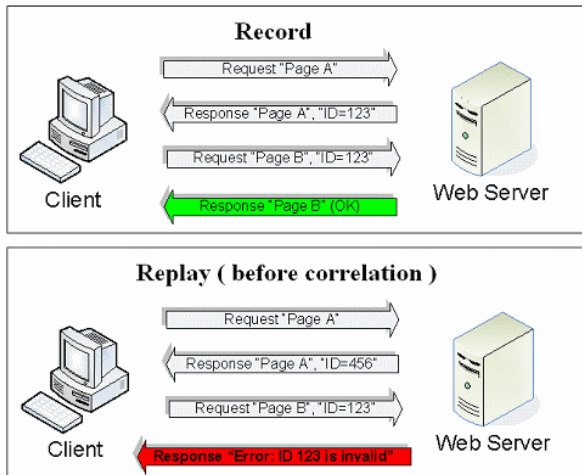
关联 (correlation) 就是把脚本中某些写死的 (hard-coded) 数据，转变成是撷取自服务器

所送的、动态的、每次都不一样的数据。

当录制脚本时，浏览器送出网页 A 的请求，服务器将网页 A 的内容传送给浏览器，并且夹带了一个 ID=123 的数据，当浏览器再送出网页 B 的请求时，这时就要用到 ID=123 的数据，服务器才会认为这是合法的请求，并且把网页 B 的内容送回给浏览器。

在执行脚本时会发生什么状况？

浏览器再送出网页 B 的请求时，用的还是当初录制的 ID=123 的数据，而不是用服务器新给的 ID=456，整个脚本的执行就会失败。



常用的关联技术有三种：

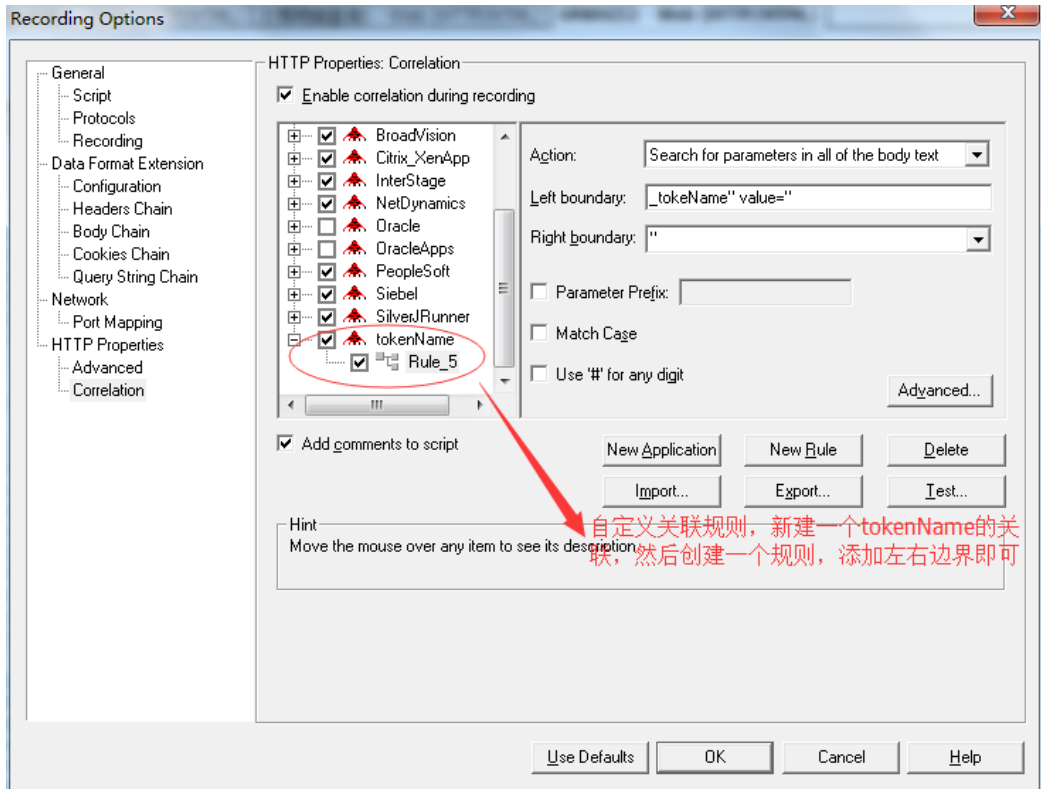
a) 录制中关联

VuGen 内建自动关联引擎，通过内建的关联规则和自定义规则，可以自动找到需要关联的值，并且自动关联函数建立关联。关联规则最重要的是制定两个边界，即被关联值的左边界和右边界。

点选 VuGen 的 Tools->Recording Options->Internet Protocol>Correlation，勾选 Enable correlation during recording，以启用自动关联。

关联的规则有两种：

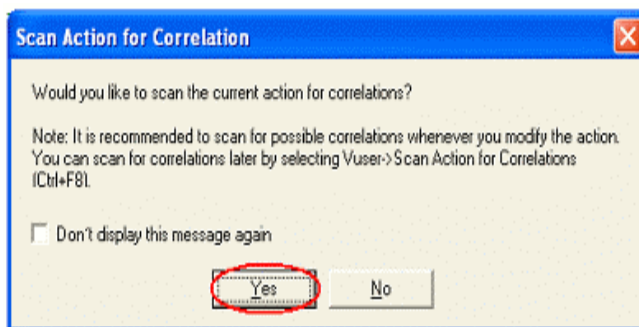
- 1、内建关联规则：loadrunner 内部自带的一些规则，如 AribaBuyer、BlueMartini、BroadVision、InterStage、mySAP、NetDynamics、Oracle、PeopleSoft、Siebel、SilverJRunner 等，请勾选相对应的应用系统。
- 2、用户自定义关联规则：如果在录制前发现默认的内建关联规则不满足录制需要，并且在录制前就已经知道了关联值的左右边界，可以通过自己动手新建一个关联规则点击 NEW APPLICATION 按钮来新建一个关联，在点击 new rule 按钮为该关联新建一个规则，新建规则时只需要添好左右边界信息即可。



b) 录制后关联

录制脚本并执行，执行完毕后，VuGen 会跳出下面的 **Scan Action for Correlation** 窗口，询问您是否要扫描脚本并建立关联，按下 **Yes** 按钮。扫描完后，可以在脚本下方的 **Correlation Results** 中看到扫描的结果。检查一下扫描的结果后，选择要做关联的数据，然后按下 **Correlate** 按钮，一笔一笔做，或是按下 **Correlate All** 让 VuGen 一次就对所有的数据建立关联。

注意：由于 **Correlation Studio** 会找出所有有变动的数据，但是并不是所有的数据都需要做关联，所以不建议您直接用 **Correlate All**。



c) 手动关联

有些需要做关联的动态数据，通过自动关联的两种方法都不能解决时，这时就只能通过做手动关联的方法进行关联了。手动关联和自动关联一样，需要先找到要关联的值，然后通过 loadrunner 提供的关联函数进行关联。

手动关联的执行过程如下：

- 1、使用相同的业务流程与数据，录制二份脚本

先录制一份脚本并存档。

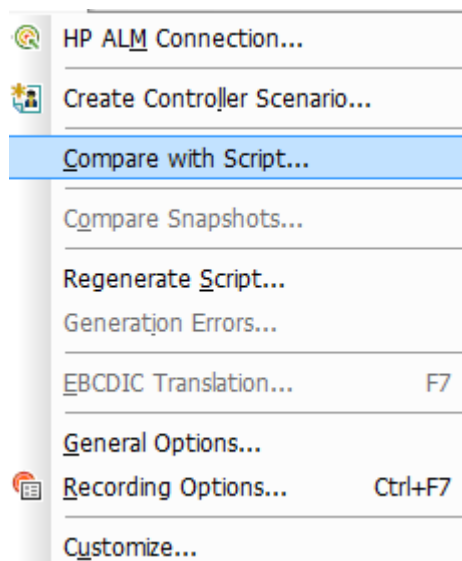
依照相同的操作步骤与数据录制第二份脚本并存档。注意，所有的步骤和输入的数据一定都要一样，这样才能找出由服务器端产生的动态数据。

有时候会遇到真的无法使用相同的输入数据，那也要记住使用的输入数据，到时才能判断是输入的数据，还是变动的数据。

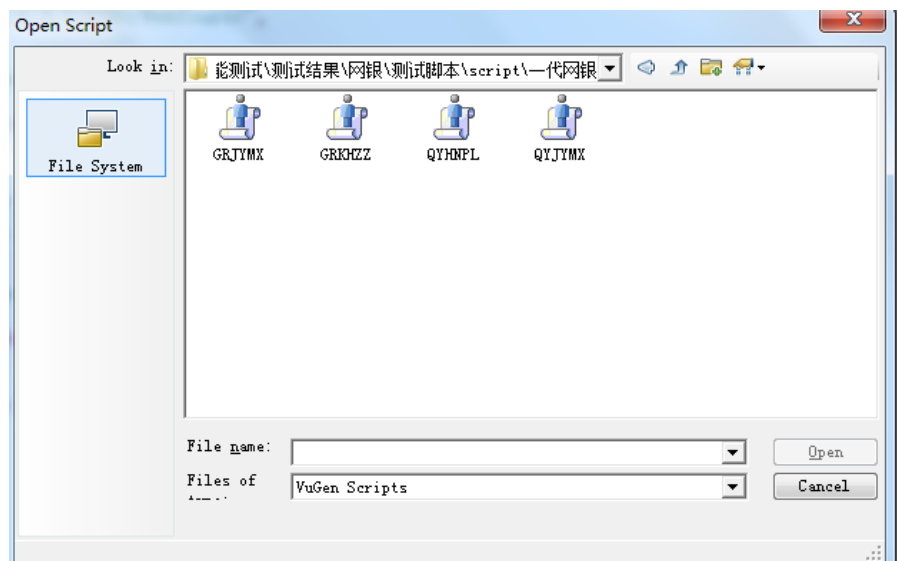
2、使用 WinDiff 工具比较两份脚本，找出需要关联的数据

WinDiff 是 loadrunner 自带的比较工具，用于比较两个脚本的内容，找出两者之间的不同的地方，对两份脚本中不同的地方进行判断，进而找出需要做关联的数据。因为需要关联的数据都是从服务器返回的数据，如果再比较中发现，对于同一个参数，服务器返回的值不同，那么这个数据是动态的，需要进行关联

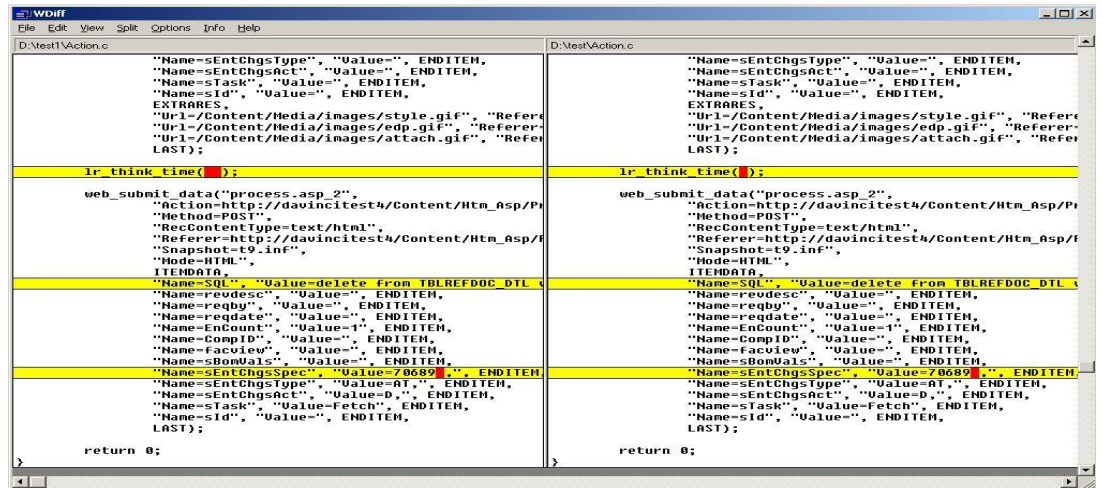
在 tools 菜单下选择 Compare with Script 选项，如图，



在弹出的对话框中选择要与当前脚本比较的脚本



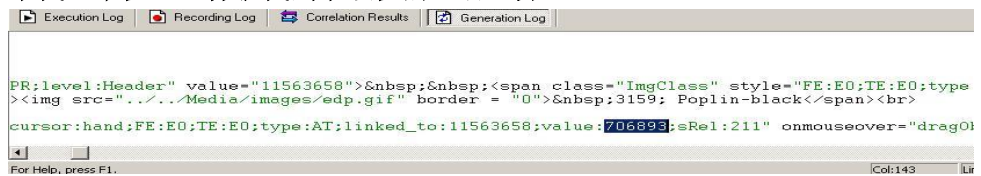
接着 WinDiff 启动，显示两个脚本，并显示有差异的地方，WinDiff 会以黄色表示有差异的脚本。如图



注意：请忽略 lr_thik_time 的差异部份，因为 lr_thik_time 是用来模拟每个步骤之间使用者思考延迟的时间

3、找到左边界和右边界字符串

在插入关联函数之前，一定要先找到左边界和右边界字符串，选择将 WinDiff 比较中不同的字符拷贝出来，在 Recording Log 或 Generation Log 中找这个值，这样就找到了该值的左右边界。



4、使用 web_reg_save_param 函数手动建立关联

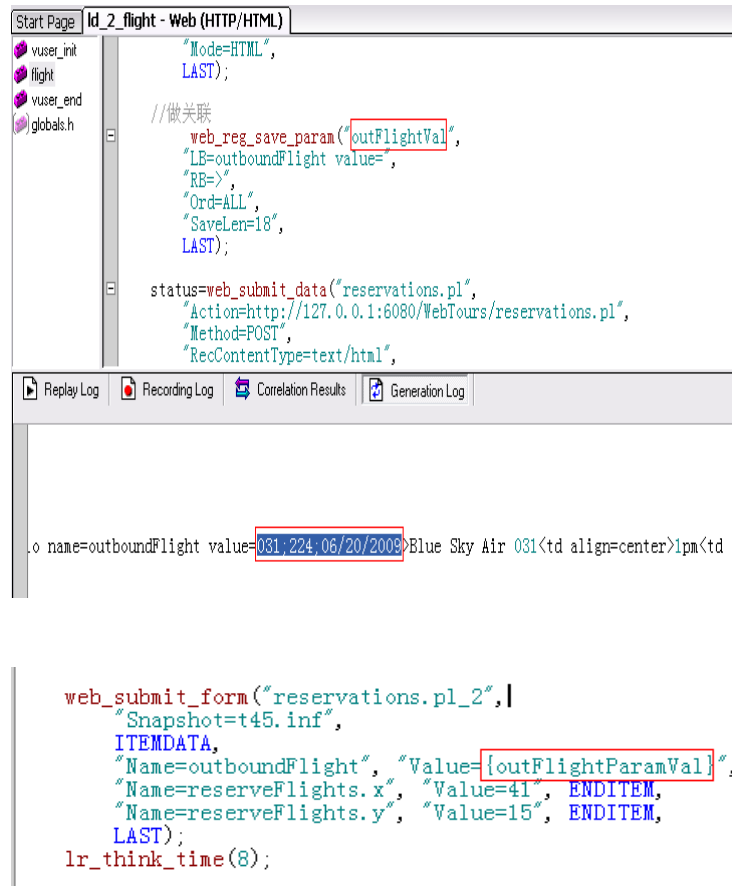
找到关联的左右边界后，需要插入 web_reg_save_param 函数来手动建立关联，首先需要找到关联函数的插入位置，在 Recording Log 或 Generation Log 中找到 WinDiff 中比较不同的字符串，这段代码前面就是插入关联函数的位置。

web_reg_param 函数说明：

web_reg_save_param 函数中设置“ORD=ALL”参数，可以使函数返回所有满足条件(所给定的左边界和右边界)的记录

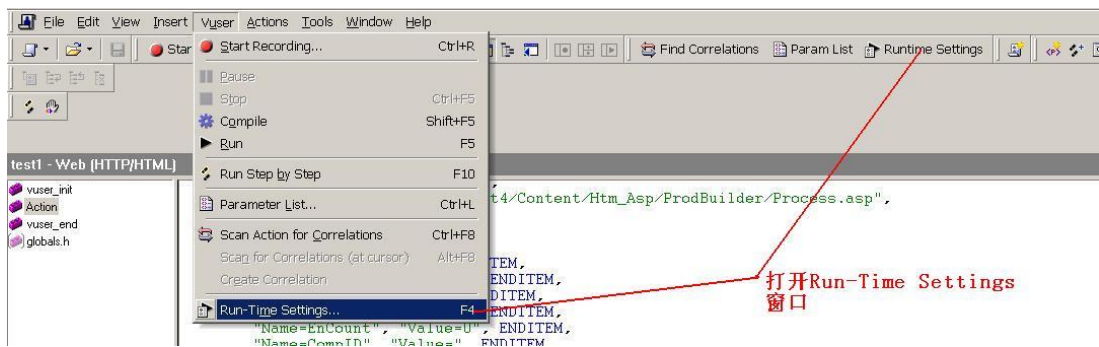
web_reg_save_param 函数中设置的“SaveLen=18”，因为服务器返回值的默认选中的第一行是” 030;251;10/18/2007 checked >”，设置 SaveLen 为 18 可以使函数的返回值中，只包含 030;251;10/18/2007，而不包含 checked 参数 outFlightVal_count 是自动创建的，在脚本中无需定义这些参数。outFlightVal_1 代表第一条记录，outFlightVal_2 代表第二条记录...依次类推。

5、将脚本中有用到关联的数据，以参数取代。如下图：



3.4 配置 Run-Time Settings 项

当完善了测试脚本后，需要对 VuGen 的 Run-Time Setting 进行配置首先打开 Run-Time Setting 窗口，可以通过菜单或者工具栏进行

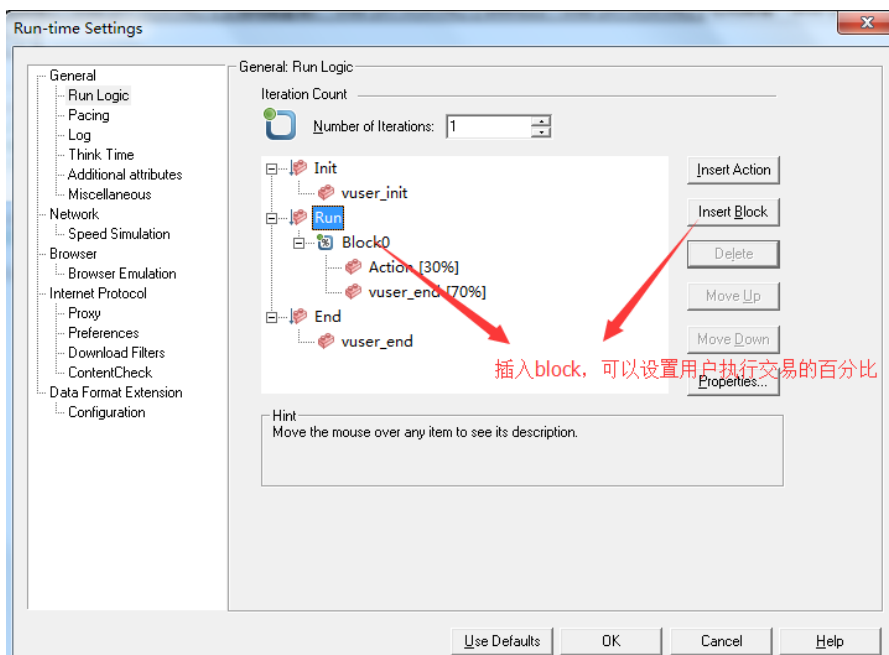


3.4.1 RunLogic 页面设置

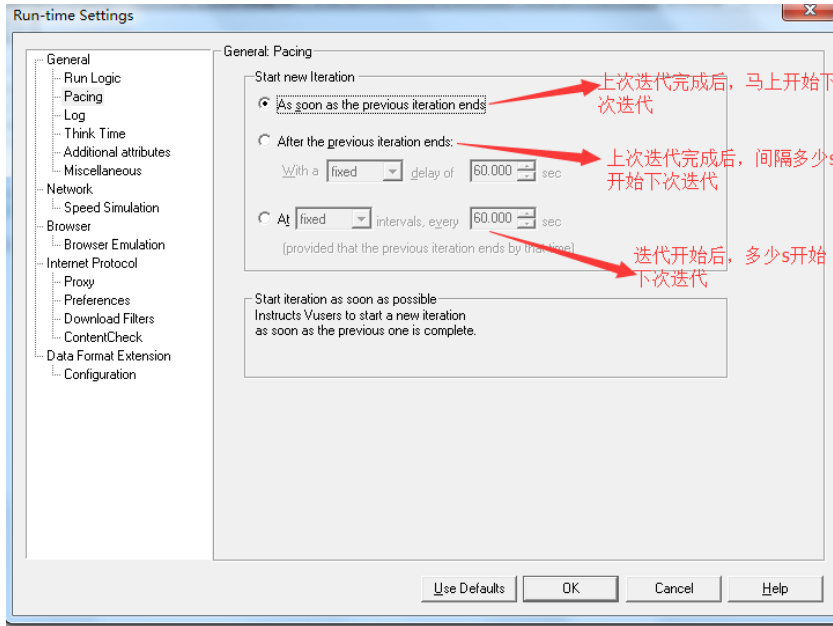
在 RunLogic 页面可以设置迭代次数，设置的迭代次数，只迭代 action 部分，init 和 end 部分的脚本不进行迭代



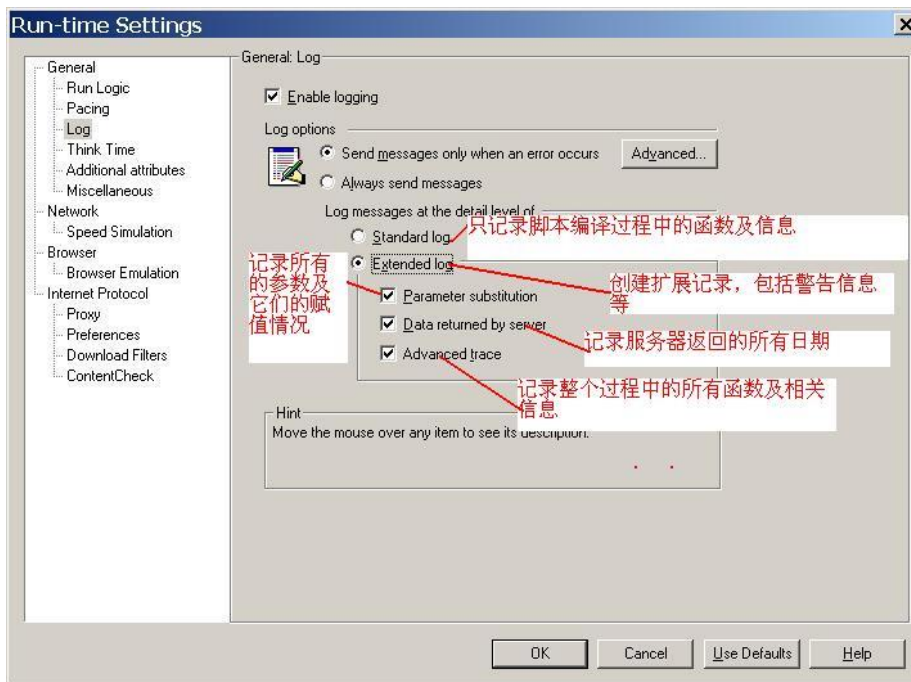
在 RunLogic 页面可以设置 block，block 可以设置不同的脚本，执行不同的次数，也可以设置不同的脚本执行百分比，如 1000 登录后，其中 30%的用户在执行查询操作，70%的用户直接退出。



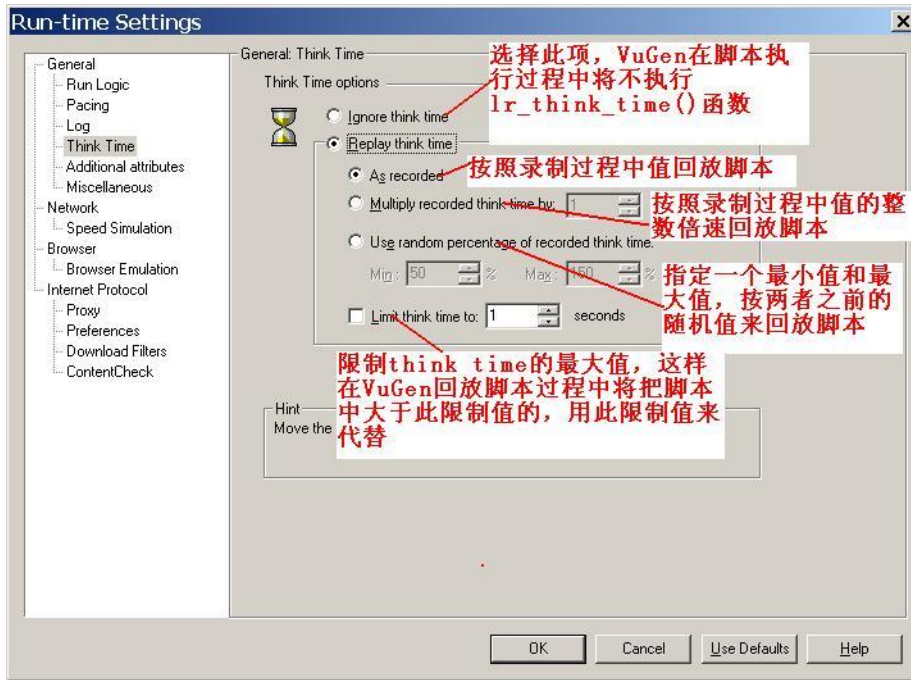
3.4.2 Pacing 页面设置



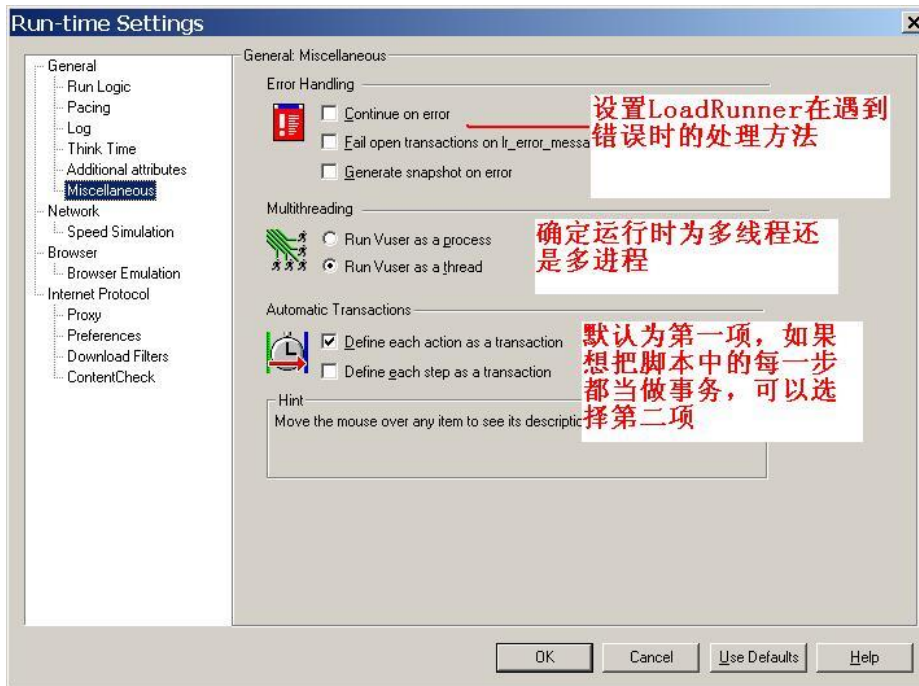
3.4.3 Log 页面设置



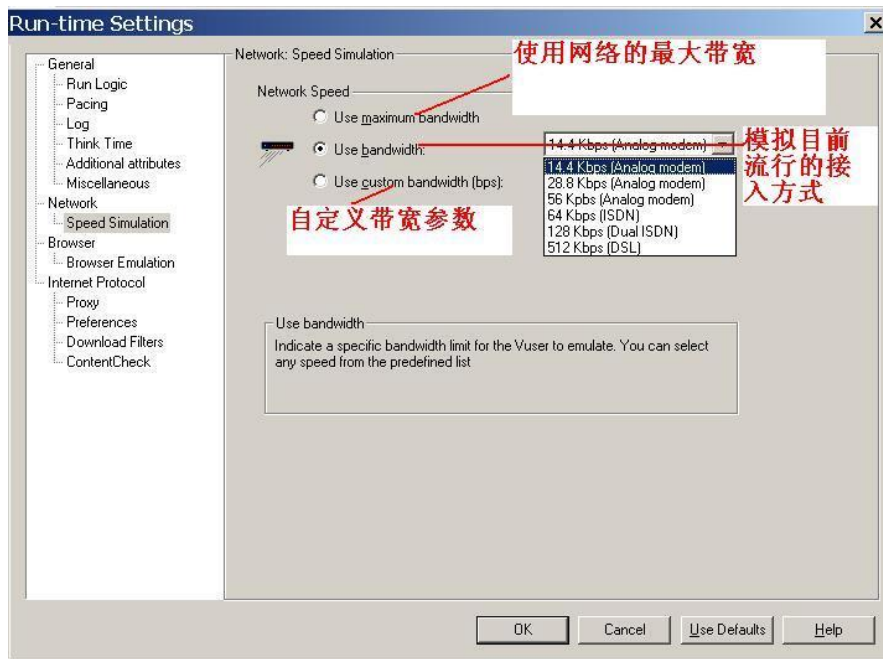
3.4.4 Think Time 页面设置



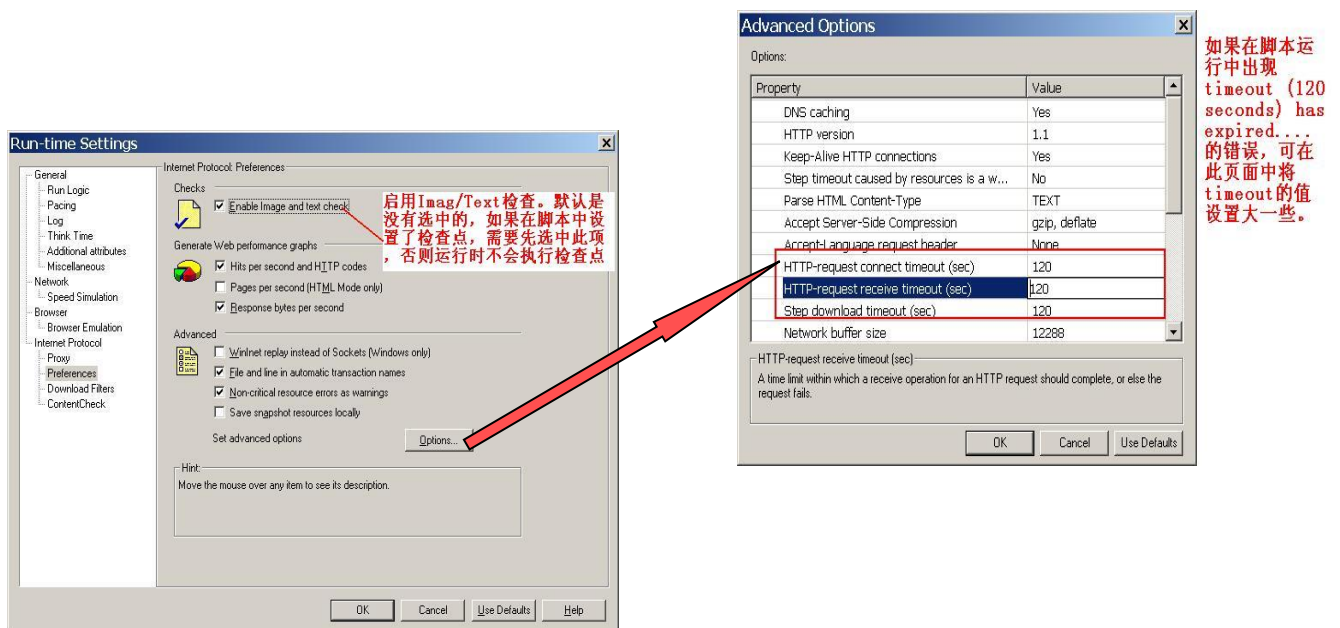
3.4.5 Miscellaneous 页面设置



3.4.6 Speed Simulation 页面设置



3.4.7 Preferences 页面设置

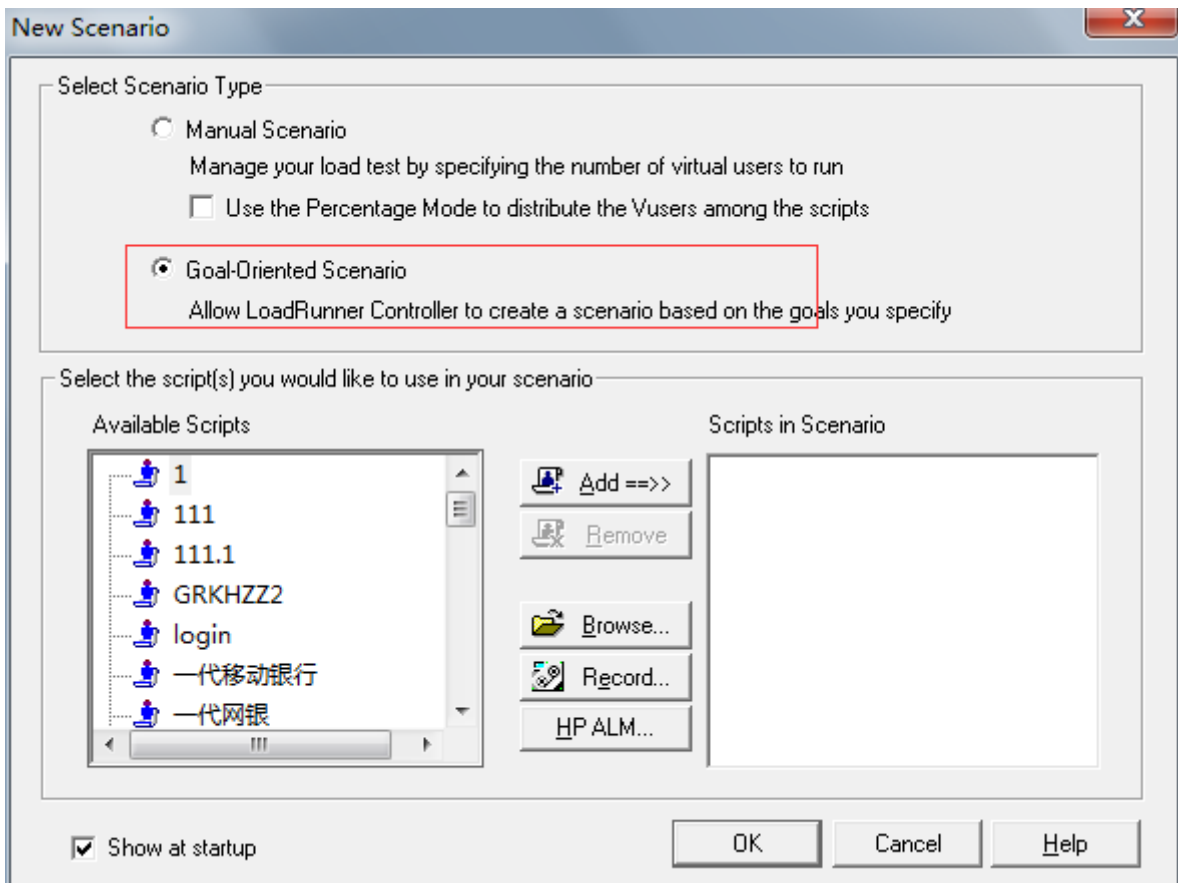


4 场景设计

4.1 目标场景

在面向目标场景中，首先要定义测试要达到的目标，然后 loadrunner 会根据这一目标创建场景。

在新建场景对话框中，选择面向目标 Goal-Oriented Scenario，选择该项后，LoadRunner 基于这个目标，自动为你创建一个场景。在场景中，我们只要定义好我们的目标即可。如下图

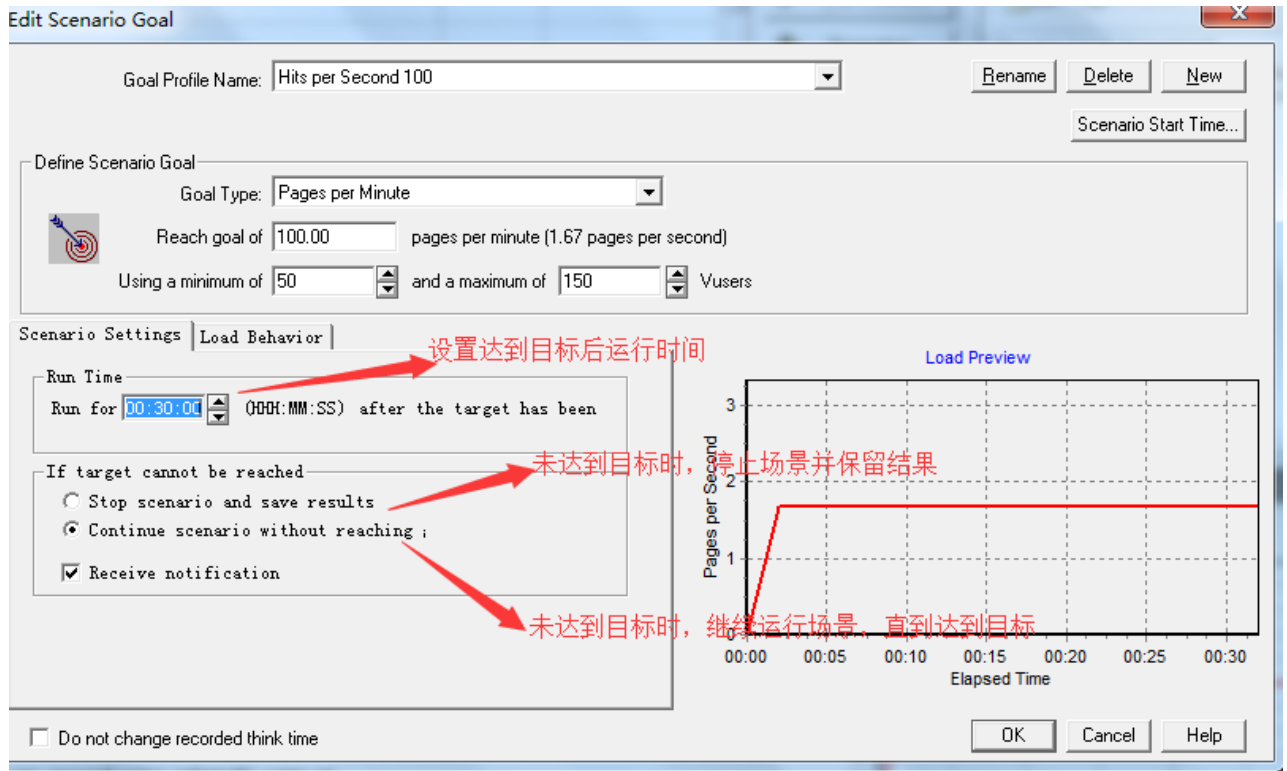


在场景设置页面，点击 **Edit Scenario Goal** 按钮，进入编辑目标场景对话框，包含五种目标类型，分别为 Virtual Users、Hits Per Second、Transactions Per Second、Transactions Response Time、Pages Per Minute。下面以 Hits Per Second 目标类型为例，说明各项设置。

1、Scenario setting 设置

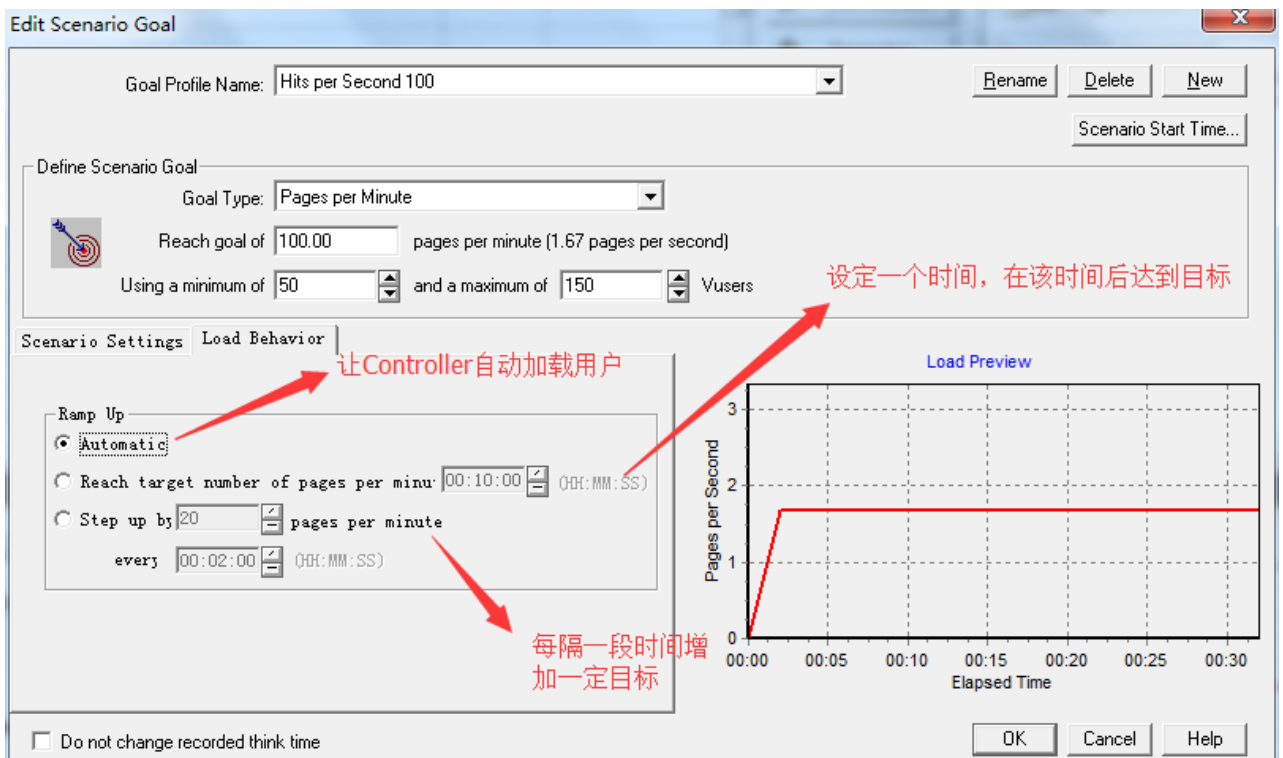
Run Time 中设置一个时间值，表示当执行达到目标后，该场景还会持续运行一段时间才结束运行

If targrt cannot be reached 设置表示如果目标无法达到，controller 将如何处理场景。有两种选择，可以选择停止运行场景并保留结果或继续运行场景直到达到目标

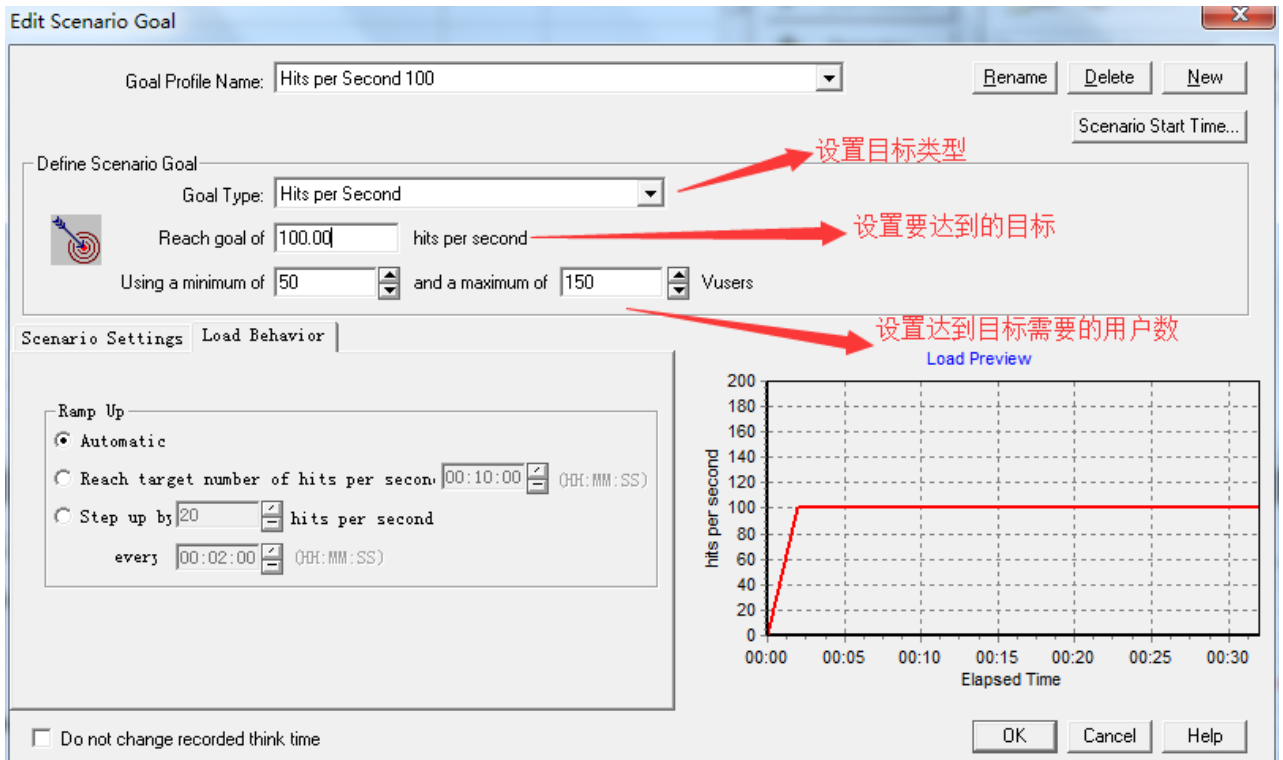


2、Load behavior 设置

设置加载行为，如下图所示



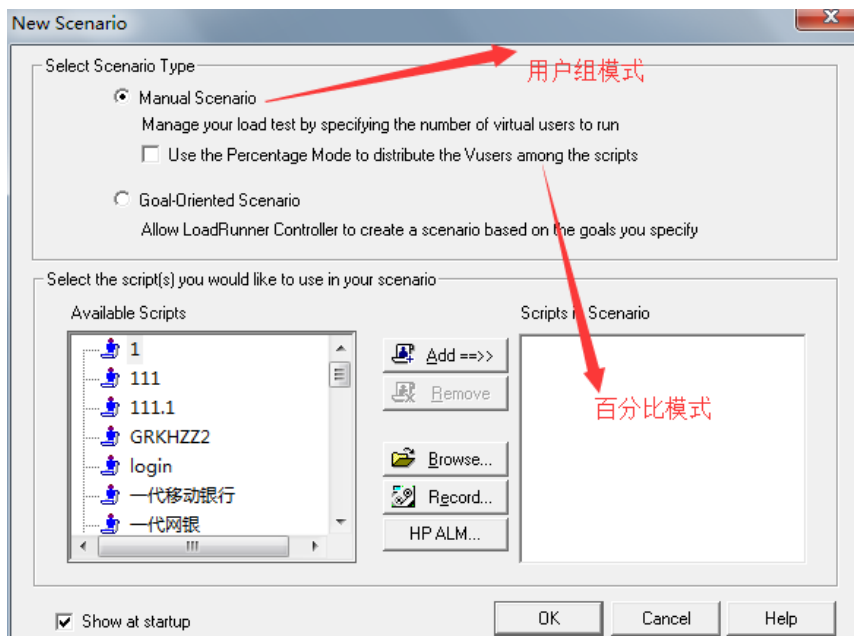
3、Define Scenario Goal 设置



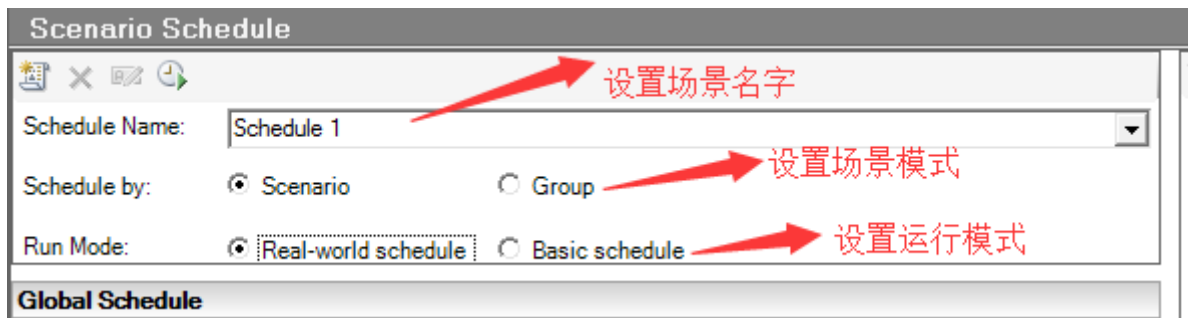
4.2 手动场景

因为手动场景更加灵活，可以更好的模拟用户的真实行为，所以一般情况下，手动测试场景使用较多

手动场景又包含两种模式：用户组模式和百分比模式，这两种模式的不同之处在于计算虚拟用户的方式不同

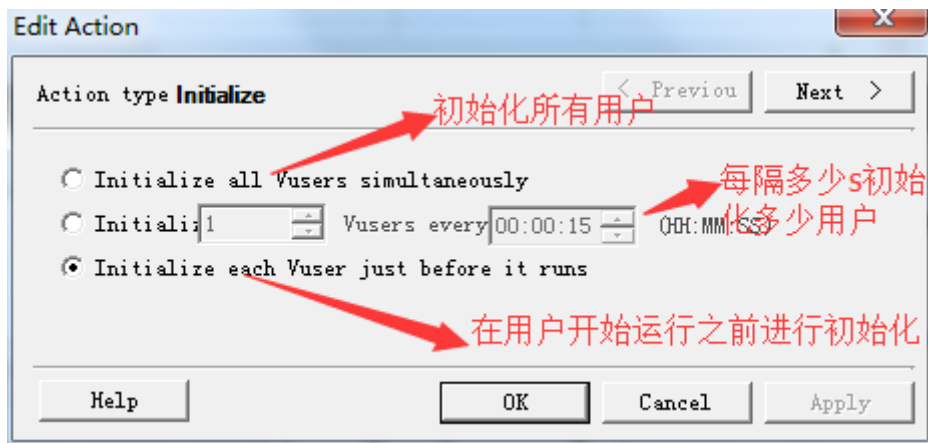


1、Scenario schedule 设置

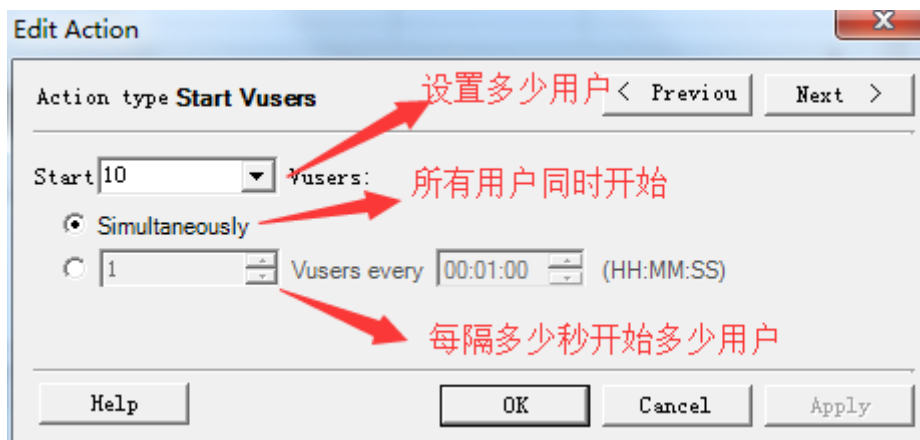


2、Global schedule 设置

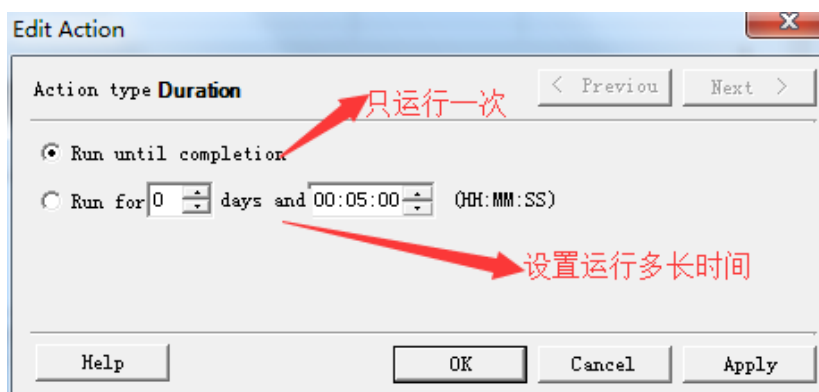
Initialize 设置:



Stat Vusers 设置:

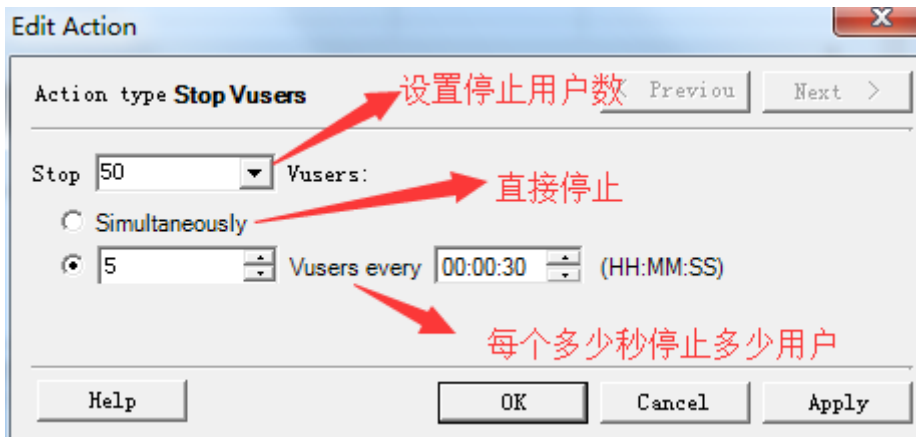


Duration 设置:

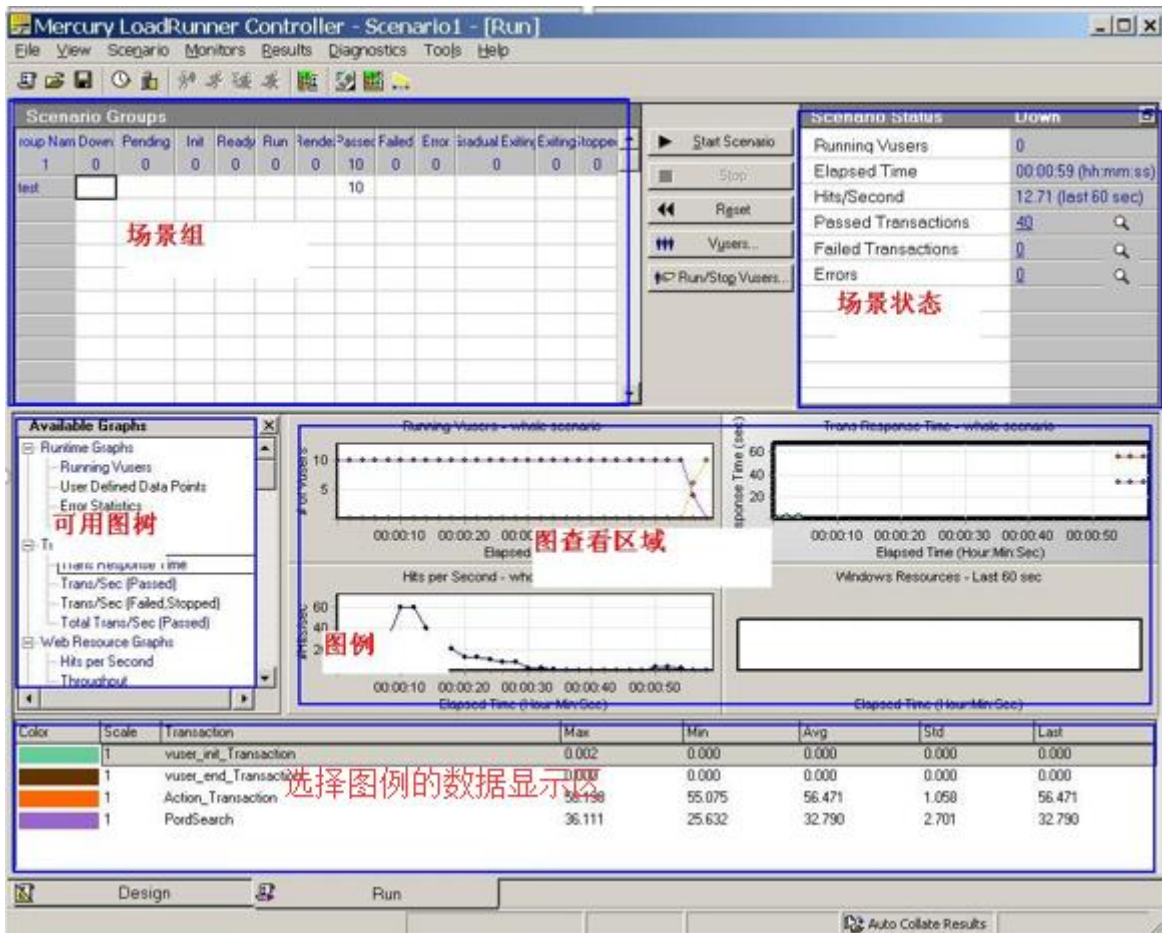


注意：当设置为 Run until completion 时，run time setting 设置的迭代次数生效。

Stop users 设置：



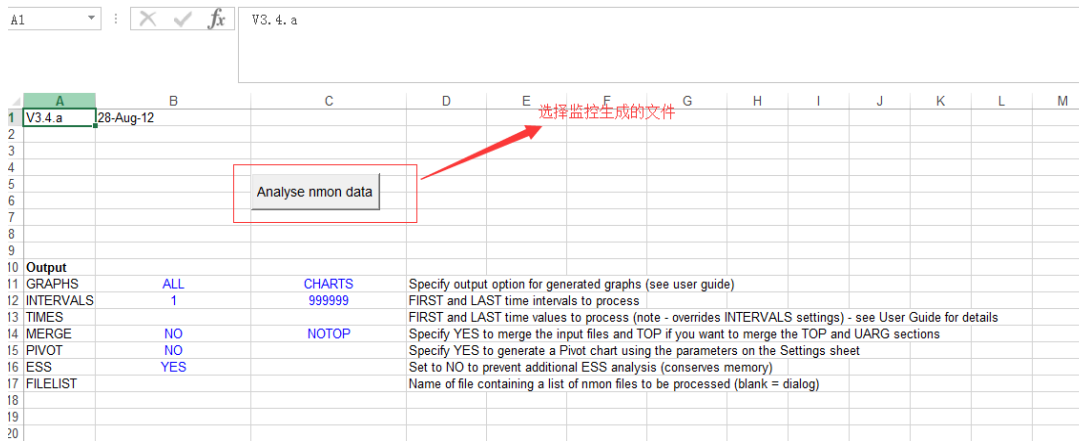
场景运行窗口：



4.3 NMON 监控

使用 NMON 监控的执行命令：`nmon -f -s 1 -c 600` 其中-s 为每个多少秒监控一次，-c 为监控的次数，例如：要监控 5 分钟，每 1 秒监控一次，需要修改指令为 `nmon -f -s 1 -c 300`

Nomo 监控生成的文件需要使用 `nmon analyser v34a.xls` 工具打开，通过点击 Analyse nmon data 按钮，选择监控生成的文件，打开监控结果



打开后，可以在生成的 Excel 表格中，查看服务器的资源使用情况。

常用的几项指标可以在以下 sheet 页中进行查看

Sheet	资源
CPU_ALL	查看 CPU 的使用情况
MEM	查看内存的使用情况
Disk_SUM	查看磁盘读写及 IO 使用情况

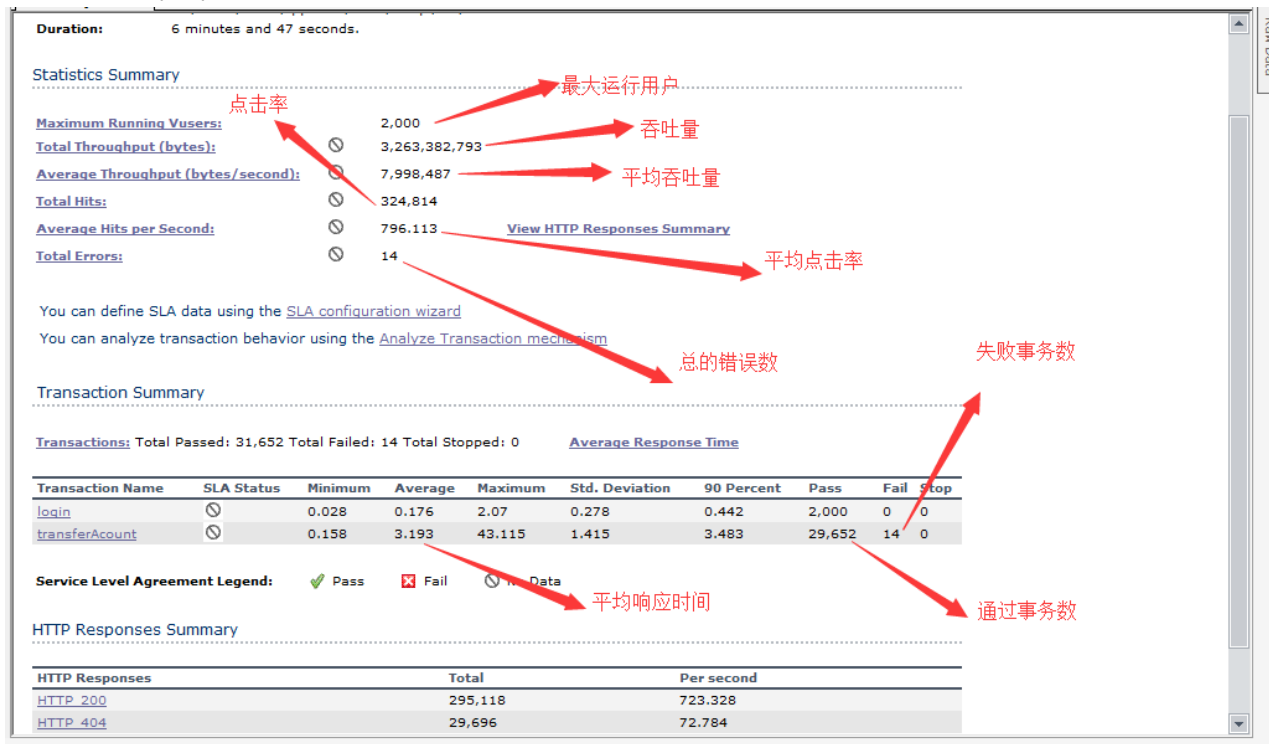
5 结果分析

在 Analysis 会话过程中生成的图和报告提供了有关系统性能的重要信息。使用这些图和报告，可以轻松地标识和确定应用程序中的瓶颈以及提高系统性能所需的改进。

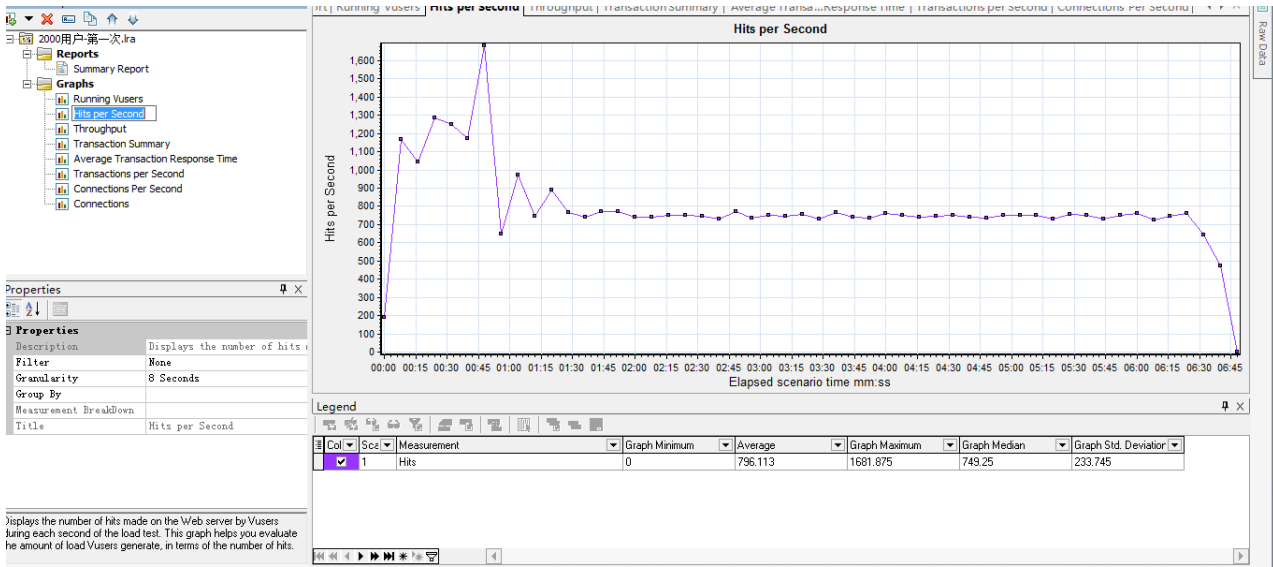
从 AnalySis 中可以得到用户终端的事务响应时间，事务的平均响应时间，网络和服务器的响应时间等。

下面对 AnalySis 的各个界面做下介绍

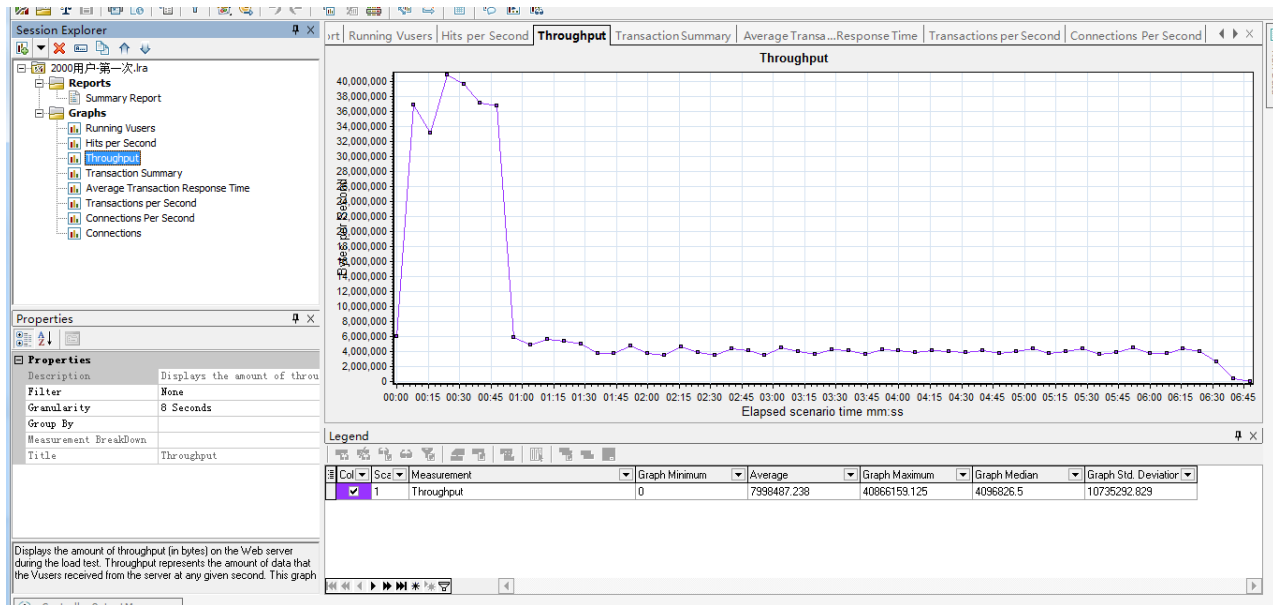
Summary report 界面介绍：可以在该界面获取运行用户数、平均相应时间和通过事务数、失败事务数



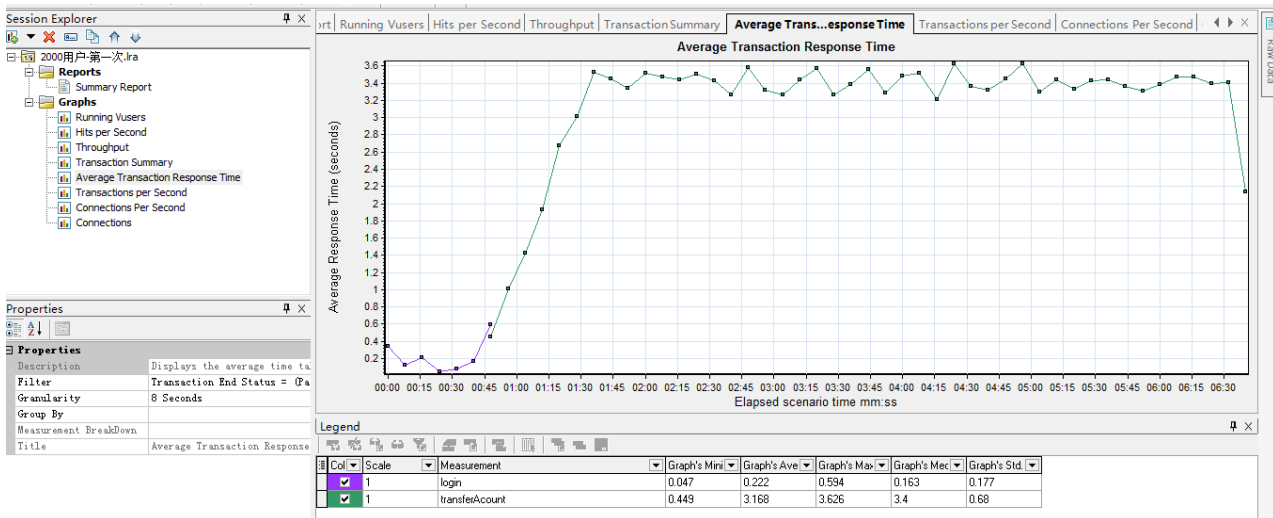
Hits per Second 图表：查看点击率



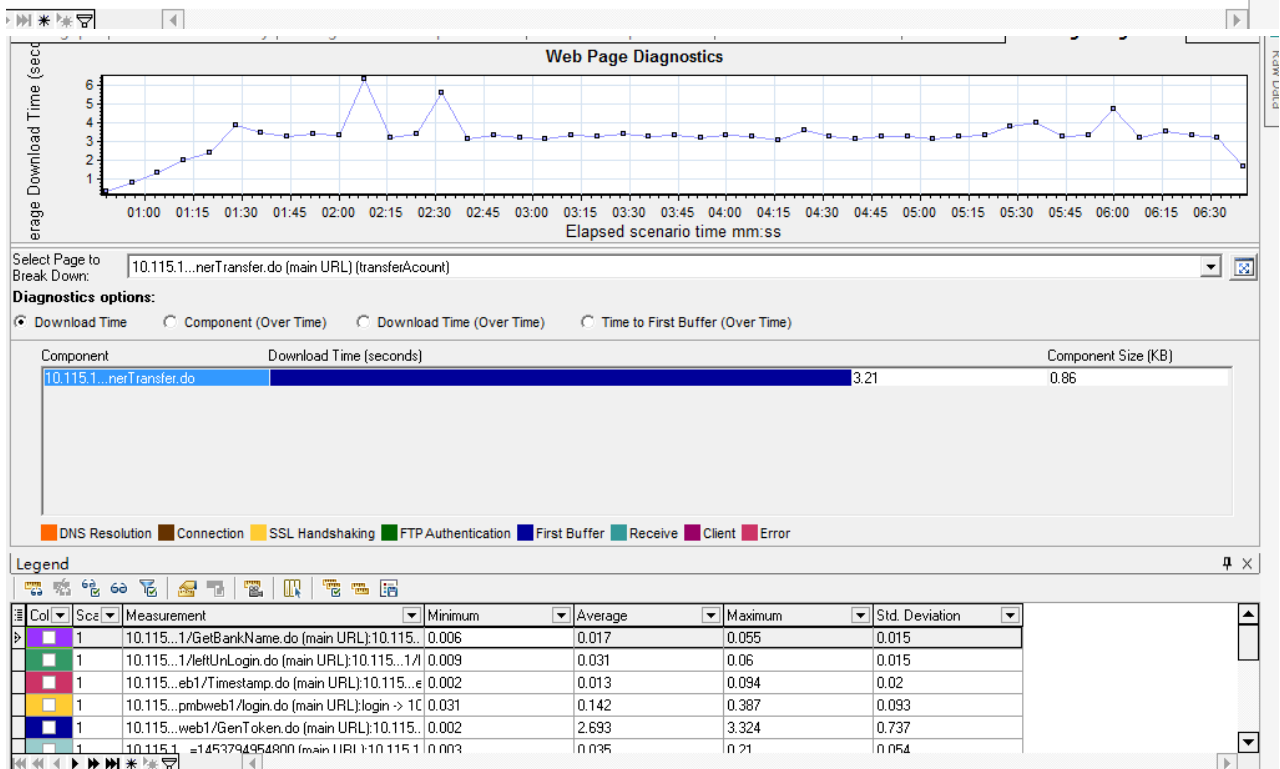
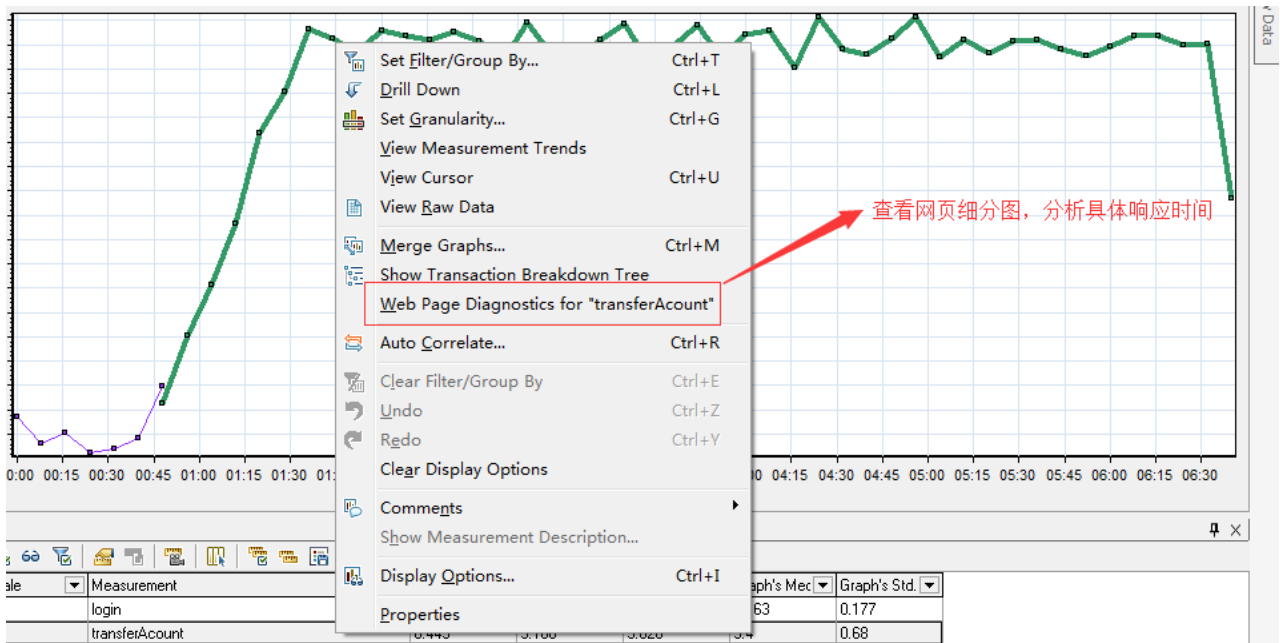
Throughput 图表：查看吞吐量



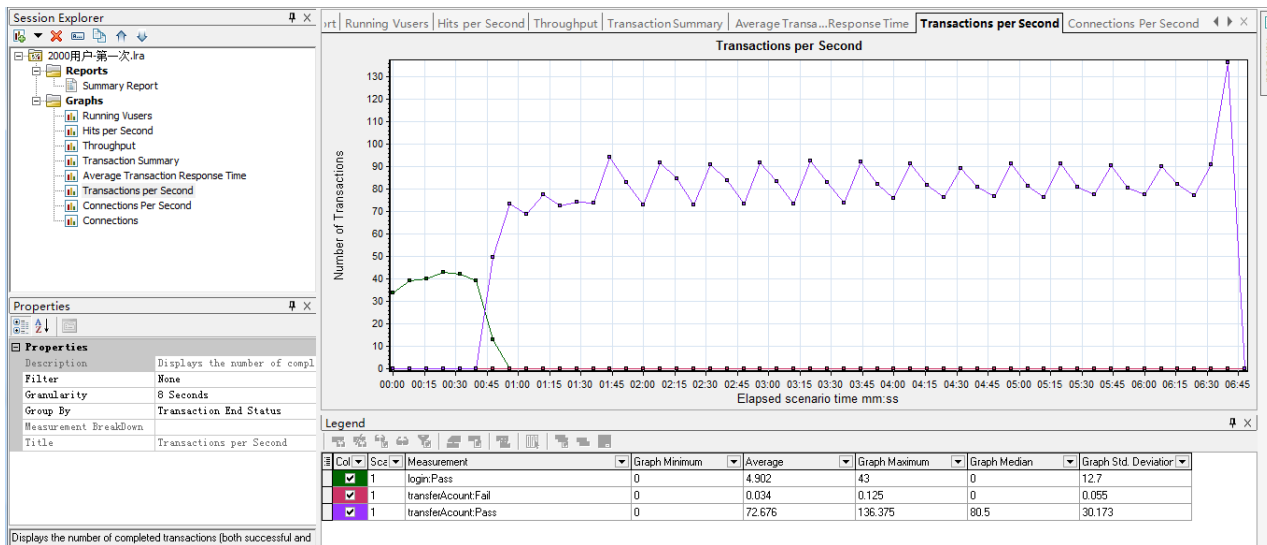
Average Transaction Response Time 图表：查看平均响应时间



在平均响应时间图表中，当响应时间较长或者波动较大时，可以选中响应时间图线，点击右键，选择 **web page diagnostics**，可以查看页面细分图，分析具体响应时间的消耗情况。



Transaction per second 图表：查看每秒事务数(Tps)



6 常见问题

- 录制脚本时，IE 弹出后马上关闭
解决方法：关闭 IE 高级选项中的第三方支持
- 录制脚本时，脚本显示的内容为乱码
解决方法：修改录制选项支持 UTF-8
- 关联失败
解决办法：打开日志，检查关联结果；调整关联函数的位置，将左右边界的位置设置的更为明确。当_dataMap 的参数值太长时，可以通过添加 `web_set_max_html_param_len("2048");`调整关联值的长度。
- 录制时脚本为空
解决方法：关闭掉杀毒软件和 360 保护软件等，检查浏览器是否为支持的浏览器；关闭浏览器后重新录制。
- 在场景中如何查看每个用户的运行情况
解决方法：打开用户管理，在你想要查看的用户上单击鼠标右键，打开 `show Vuser` 选项即可
- 为什么在场景中运行时间很长，但实际操作时间没那么长
解决办法：查看事务函数中是否有 `lr_think_time()`函数，在结果中过滤掉即可。
- 在场景中每个用户的执行结果均为 PASS，但在系统中没有任何操作完成。
解决办法：用户的状态并不代表用户的执行状态，用户最终状态为 PASS，只能说明用户完成了请求的发送过程，但不能表示操作的逻辑成功。要检查操作的逻辑是否成功，需要添加事务检查来判断事务的完成状态。
- 在 win7 系统中，loadrunner 经常无法完成操作
解决办法：由于 win7 系统默认并不是以系统管理员的权限运行系统，就会由于权限的问题，导致一部分操作无法执行，使用右键使用管理员权限打开软件即可。
- 录制的文件上传脚本，回放错误
解决办法：一般上传文件时，是通过 `web_submit_data` 函数实现，在提交表单时，由于文件位置并不是放在脚本目录下，提交时报错。可以直接修改文件名为绝对路径或将文件存放在脚本目录下即可解决。