

第 5 章

LoadRunner–Vugen 模拟用户行为

Virtual User Generator(虚拟用户生成器, 以下简称为 Vugen)是一种基于录制回放的工具, 当你按照业务流程执行了某个软件, 它会将你在操作中产生的协议录制下来, 自动转化成脚本, 执行完成对用户行为的模拟, 从而进一步对系统产生负载。而性能测试的第一步也是最重要的一步就是生成虚拟用户脚本(Vuser Script)。在 Vugen 中录制得到用户的行为就好比虚拟了一个用户的行为, 所以我们称该模拟的用户为 Vuser, 而这个脚本称为 Vuser Script。

5.1 Vugen 介绍

Virtual User Generator(虚拟用户生成器,必要时可简写为 Vugen),可通过录制用户在客户端应用程序中执行的典型业务流程来开发脚本,此脚本将模拟多用户并行工作,就像真的有成千上万的用户在访问系统一样,并追踪从服务器接收的数据。

录制用户行为过程中,Vugen 监控的是客户端与服务器之间的数据交互,将客户端与服务器的交互过程记录下来,如图 5-1 所示。

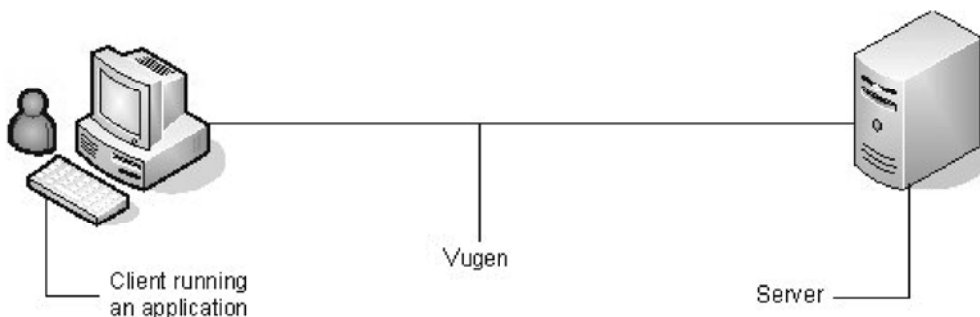


图 5-1 Vugen-1

回放时,Vuser Scripts 通过调用服务器 API 直接与服务器进行交流。因为不需要客户端界面,所以允许大量用户运行或使用更少的机器进行测试,如图 5-2 所示。



图 5-2 Vugen-2

Virtual User Generator 在此过程中,主要扮演一个 proxy server 的角色,在录制脚本时,记录下用户和服务器交互,然后自动生成脚本语言。在接下来的重放,或者大批量地加压时,模拟真实的用户向网站发送请求,并将服务器返回的结果,作为判断是否正确执行用

户操作的依据。

值得注意的是, Virtual User Generator 关心的是客户端与服务器之间收发的数据包, 而不会记录客户端界面的一些操作。

5.2 协议类型及选择

协议是客户端与服务器之间遵循某种数据包发送格式的约定, 各自遵循约定才能通信成功。

首先打开 HP Virtual User Generator 界面, 如图 5-3 所示。

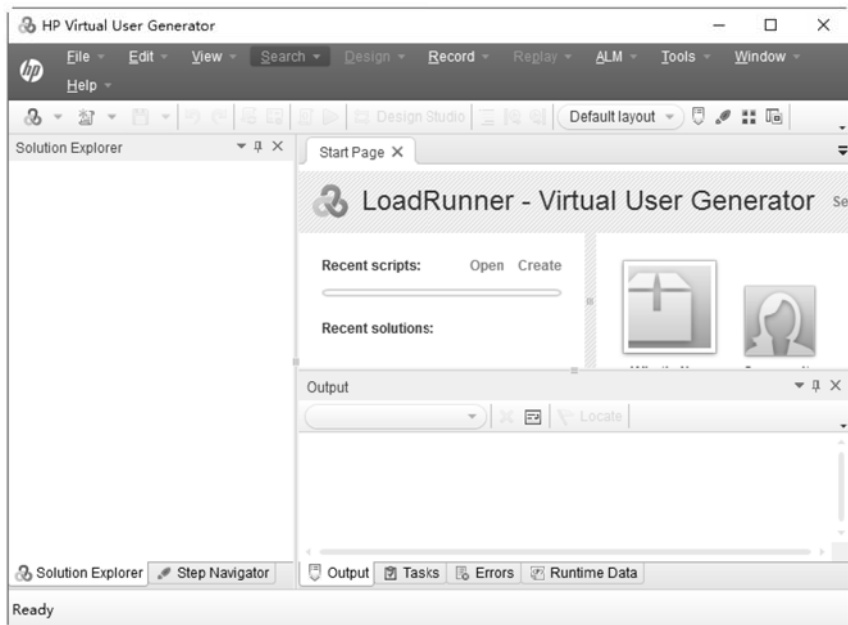


图 5-3 HP Virtual User Generator 界面

如图 5-4 所示, 选择 File→New Script and Solution 命令, 可以打开如图 5-5 所示的 Create a New Script 对话框, 可以看到 LoadRunner 支持的协议很多。

使用时如何选择呢? 首先要从 LoadRunner 的工作原理上深入理解协议的作用和意义。LoadRunner 启动后, 在任务栏上会有一个 LoadRunner Agent Progress 的进程, 这个进程的一项重要的工作就是监视各种协议的客户端和服务器的通信。只要是能够支持的协议, LoadRunner 在录制的过程中就可以通过脚本语言将通信过程录制下来。所以只有明确了被测软件的通信过程和所使用的协议, LoadRunner 才能正确地录制脚本。

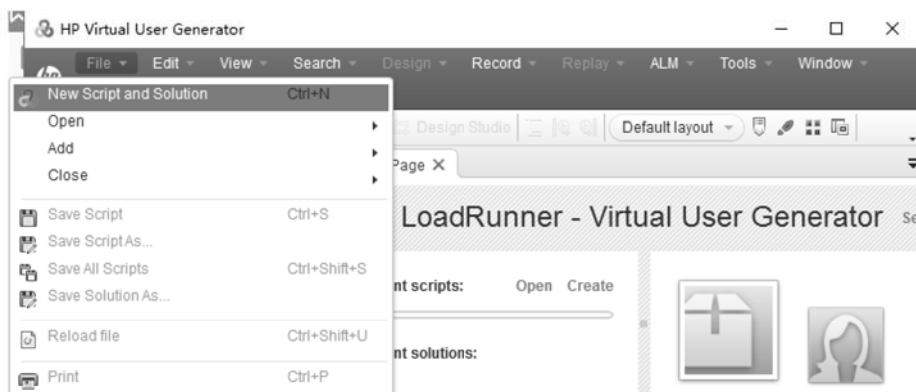


图 5-4 选择 File→New Script and Solution 命令

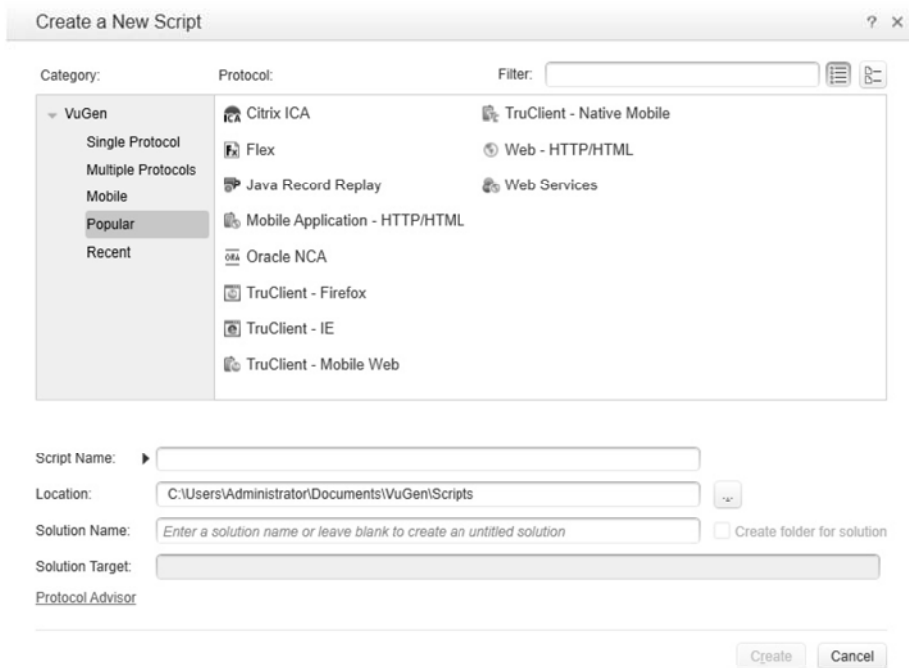


图 5-5 协议类型选择

对于常见的应用软件，我们可以根据被测应用是 B/S 结构还是 C/S 结构来选择协议：

- (1) 对于 B/S(浏览器/服务器)结构，可以选择 Web(HTTP/HTML)协议；
- (2) 对于 C/S(客户端/服务器)结构，可以根据后端数据库的类型来选择，如 SybaseCTLib 协议用于测试后台数据库为 Sybase 的应用，MS SQL SERVER 协议用于测试后台数据库为 SQL Server 的应用；
- (3) 对于没有数据库的 Windows 应用，可以选择 Windows sockets 这个底层的协议。

需要明确的是，B/S 结构的不一定都选择 WEB(HTTP/HTML) 协议，C/S 结构的不一定都选择 WinSocket 协议，有很多 C/S 结构的系统其实用的也是 HTTP 协议。

那么实际使用中如何判断客户端与服务器之间的协议类型呢？这里有两种方式供参考。

(1) 直接跟开发人员沟通。这是非常高效的一种方式，而且有时候如果开发人员能够提供报文格式是最好不过了，将省去很多的工作量。

(2) 自己动手，丰衣足食。你可以借助优秀的第三方协议分析工具来帮助你分析，如 Wireshark、Network Monitor、SniffPass 等。这些工具除了帮助分析协议外，还提供其他更详细的信息，作为一个性能测试人员，多学习一些网络协议方面的知识也是非常必要的。

5.3 录制前准备工作

在使用 Vugen 工具进行录制前，需要做一系列准备工作，下面分别进行介绍。

5.3.1 B/S 端录制脚本

这里以 B/S 结构的程序为例，讲解录制过程。如图 5-6 所示，选择 Web-HTTP/HTML 协议，下面的脚本名称等都设为默认。

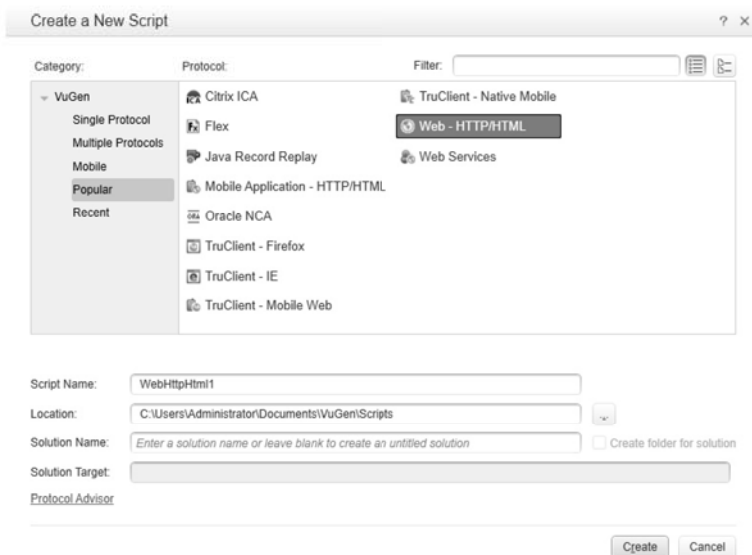


图 5-6 选择 Web-HTTP/HTML 协议

下面介绍脚本创建的四个阶段：

- 录制；
- 验证；
- 脚本增强；
- 准备压力测试。

单击 Next 按钮，弹出如图 5-7 所示的 Start Recording(开始录制)对话框。

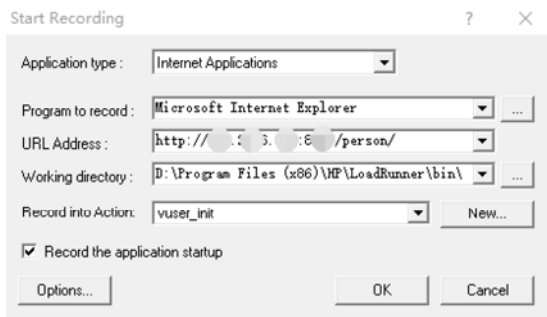


图 5-7 脚本录制对话框

Program to record 下拉列表框用来选择本地安装的浏览器。LoadRunner 12 支持的浏览器类型主要有以下几种：

- Microsoft Internet Explorer 6.0 SP1 or SP2；
- Microsoft Internet Explorer 7.0；
- Microsoft Internet Explorer 8.0；
- Firefox 4 版本及以下。

本书的讲解中我们选择 Firefox24.8.1。

URL Address 下拉列表框用来填写要访问的服务器地址，Working directory 下拉列表框用来选择是工作目录。Record into Action 下拉列表框用来选择录制的三个部分(对应三个文件)，如图 5-8 所示。



图 5-8 录制文件选择

一般情况下：

- vuser_init 录制用来做初始化的事情，比如要测试某个具体业务操作环节时，可以先把系统用户登录写在 vuser_init 的脚本中；

- action 录制的是操作的事件, 即需要测试的业务操作点;
- vuser_end 录制的是退出阶段。

LoadRunner 运行脚本的顺序是 vuser_init→Action(*N 次)→vuser_end; 其中 Action 可以运行 N 次(或一段时间), 这在 LoadRunner 中有设置(后面会讲到); 而 vuser_init 和 vuser_end 都只能运行一次。

Options 在开始使用时可以先不深究, 这里我们只简单介绍需要了解的几个设置。

打开 Recording Options 对话框, 如图 5-9 所示, Recording 页面有两个选项。

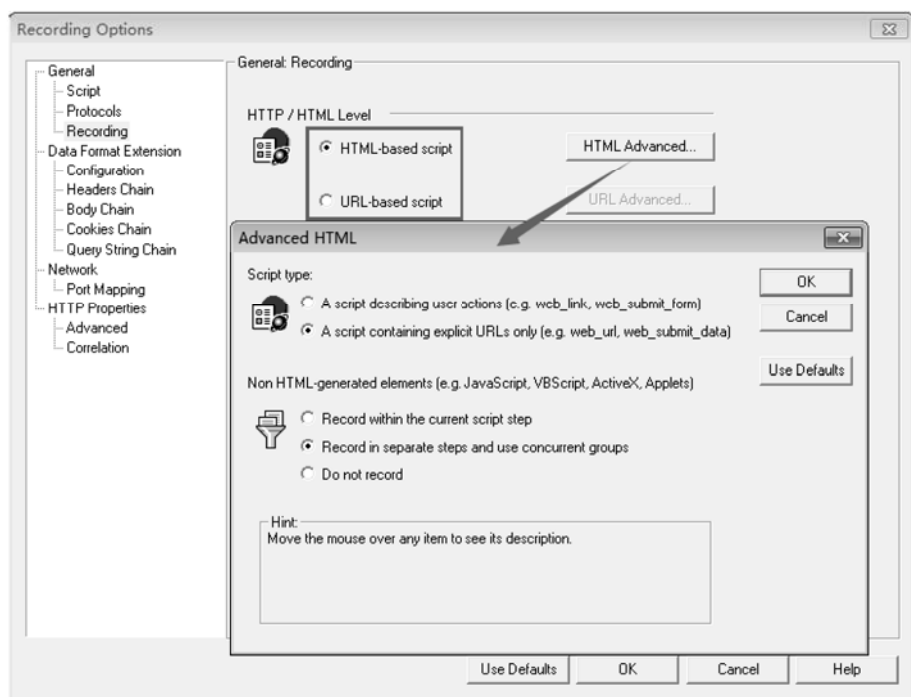


图 5-9 录制相关设置

(1) HTML-based script: 使用这种方式录制的代码, 一个操作(可能包含多个请求)只生成一个函数, 所有请求全部放在这个函数里面, 看起来比较简洁。

(2) URL-based script: 使用这种方式会生成很多函数, 它将每个请求都单独生成一个函数, 更接近请求-响应的本质。

录制时的选择遵循以下原则:

- ① 基于浏览器的应用程序推荐使用 HTML-based script。
- ② 不是基于浏览器的应用程序推荐使用 URL-based script。

如果基于浏览器的应用程序中包含了 JavaScript 并且该脚本向服务器产生了请求, 比如

DataGrid 的分页按钮等、Flash 等，也要使用 URL-based script 方式录制。

③ 基于浏览器的应用程序中使用了 HTTPS 安全协议，使用 URL-based script 方式录制。

初学者推荐使用 HTML-based script 方式，更容易阅读代码。

接下来我们开始录制一个常见的登录远程服务器并提交表单的操作：

- (1) 登录；
- (2) 依次浏览两个页面；
- (3) 在第二个页面进行录入提交操作；
- (4) 退出登录；
- (5) 关闭网页。

开始录制后，会自动打开 Firefox 浏览器，依次进行以上五个步骤的操作，过程中我们发给服务器的所有请求都会记录在脚本中，完成后生成脚本如下。

1. vuser_init

录制客户端登录服务器的操作，脚本如图 5-10 所示。

```
vuser_init()
{
    web_add_cookie("Hm_lvt_0fce9465689d3ab7ee3c77f3b2459172=1523171151,1523171192,1523174295,1523174385; DOMAIN=1.2.3.4");
    web_url("person",
        "URL=http://1.2.3.4:8080/person/",
        "TargetFrame=",
        "Resource=0",
        "RecContentType=text/html",
        "Referer=",
        "Snapshot=t140.inf",
        "Mode=HTML",
        LAST);
    lr_think_time(12);
    web_submit_data("login",
        "Action=http://1.2.3.4:8080/person/personal/login",
        "Method=POST",
        "TargetFrame=",
        "RecContentType=application/json",
        "Referer=http://1.2.3.4:8080/person/",
        "Snapshot=t141.inf",
        "Mode=HTML",
        ITEMDATA,
        "Name=username", "Value=admin", ENDITEM,
        "Name=password", "Value=123456", ENDITEM,
        "Name=randCheckCode", "Value=13", ENDITEM,
        LAST);
    return 0;
}
```

图 5-10 vuser_init 的脚本

可以看到客户端访问的服务器地址以及登录用户名、密码。

2. Action

录制用户切换至第二个页面，向服务器提交一组数据的操作，代码如图 5-11 所示。

```
web_url("newestReport",
  "URL=http://192.168.1.100:8080/person/health/newestReport",
  "TargetFrame=",
  "Resource=0",
  "RecContentType=text/html",
  "Referer=http://192.168.1.100:8080/person/personal/main",
  "Snapshot=t149.inf",
  "Mode=HTML",
  LAST);

web_submit_data("guideBaseHealthDataSave",
  "Action=http://192.168.1.100:8080/person/personal/main/guideBaseHealthDataSave",
  "Method=POST",
  "TargetFrame=",
  "RecContentType=application/json",
  "Referer=http://192.168.1.100:8080/person/personal/main/guideBaseHealthData?notCompletedHeight=0",
  "Snapshot=t168.inf",
  "Mode=HTML",
  ITEMDATA,
  "Name=healthInfoRecordId", "Value=", ENDITEM,
  "Name=userId", "Value=", ENDITEM,
  "Name=type", "Value=1", ENDITEM,
  "Name=examinationTime", "Value=2018-04-08 16:22:12", ENDITEM,
  "Name=highPress", "Value=124", ENDITEM,
  "Name=lowPress", "Value=92", ENDITEM,
  "Name=heartRate", "Value=72", ENDITEM,
  "Name=bloodOxygen", "Value=98", ENDITEM,
  "Name=temperature1", "Value=37.2", ENDITEM,
  "Name=weight", "Value=52", ENDITEM,
  LAST);
```

图 5-11 Action 的脚本

客户端向服务器提交了如图 5-11 所示的一组数据。另外如果有很多个操作，可以分布在多个 Action 中。

3. vuser_end

进入个人中心，退出登录，代码如图 5-12 所示。

```
web_url("info",
  "URL=http://192.168.1.100:8080/person/user/info",
  "TargetFrame=",
  "Resource=0",
  "RecContentType=text/html",
  "Referer=http://192.168.1.100:8080/person/health/newestReport",
  "Snapshot=t171.inf",
  "Mode=HTML",
  LAST);

web_url("logout",
  "URL=http://192.168.1.100:8080/person/personal/logout",
  "TargetFrame=",
  "Resource=0",
  "RecContentType=text/html",
  "Referer=http://192.168.1.100:8080/person/user/info",
  "Snapshot=t176.inf",
  "Mode=HTML",
  LAST);
```

图 5-12 vuser_end 的脚本

上述几个脚本中主要用到了以下几个函数：

(1) `web_url` 函数用来加载指定的网页(GET 请求)，其中 URL 是访问的服务器地址，Referer 是 URL 引用的页面(即被访问页面的来源 URL)；

(2) `web_submit_data` 函数用来生成表单的 GET 或 POST 请求，这些请求与 Form 自动生成的请求是一样的，ITEMDATA 是数据域和属性的分隔符，之后是提交表单的 name 和 value 字段。

(3) `lr_think_time` 函数用来设置 LoadRunner 的思考时间，为保持场景模拟的真实性，建议保留。

录制完成后，可以回放检查一下是否成功。

5.3.2 移动端录制脚本

因为是录制移动端脚本，所以必备的设备要有 Android 或 iPhone 手机和 具有无线上网卡的笔记本电脑或台式机(台式机可使用随身 Wi-Fi 作为无线上网卡)。

下面对笔记本电脑进行一些设置(手机端的设置参见 5.3.4 节)。

由于录制的时候需要笔记本电脑和手机处于同一网络环境下，而且由于大部分办公环境并没有公共网络可以连接，因此比较简单的方法就是开启笔记本电脑的热点，然后让手机连接。笔记本电脑开启热点可以通过工具，也可以通过 cmd 命令。

(1) 以管理员身份运行 cmd，然后输入命令：`netsh wlan set hostednetwork mode=allow ssid=WIFINAME key=0123456789`，按 Enter 键，系统会自动虚拟出一个 Wi-Fi 热点，如图 5-13 所示。



图 5-13 虚拟 Wi-Fi 热点

(2) 开启热点，输入命令：`netsh wlan start hostednetwork`，按 Enter 键，如图 5-14 所示。

```
C:\Users\Administrator>netsh wlan start hostednetwork
已启动承载网络。
```

图 5-14 开启热点

(3) 打开控制面板-网络和 Internet-网络连接, 会看到系统新建了一个 Wi-Fi 热点, Wi-Fi 名字就是刚才 cmd 中设置的 WIFINAME, 如图 5-15 所示。



图 5-15 验证热点

(4) 此时可以通过手机搜索到此热点并连接。获取本地连接 IP 的方法为: 网络连接中有无线网络 IP, 请忽略, 需要使用本地 IP, 如图 5-16 所示。

```
无线局域网适配器 无线网络连接:

连接特定的 DNS 后缀 . . . . . : 
本地连接 IPv6 地址 . . . . . : fe80::108f:fc5e:a828:1b25%25
自动配置 IPv4 地址 . . . . . : 169.254.27.37
子网掩码 . . . . . : 255.255.0.0
默认网关 . . . . . : 

以太网适配器 本地连接: http://blog.csdn.net/

连接特定的 DNS 后缀 . . . . . : sn.suning.ad
本地连接 IPv6 地址 . . . . . : fe80::6c6a:203e:aa91:fe21%11
IPv4 地址 . . . . . : 10.24.38.194
子网掩码 . . . . . : 255.255.255.0
默认网关 . . . . . : fe80::f5b1:c976:f4f1:d030%11
                      10.24.38.254
```

图 5-16 获取连接

5.3.3 LoadRunner 的设置

使用 Vugen 工具之前, 需要先设置 LoadRunner, 步骤如下。

(1) 打开如图 5-17 所示的 Create a New Script 对话框, 选择 Web-HTTP/HTML 协议。

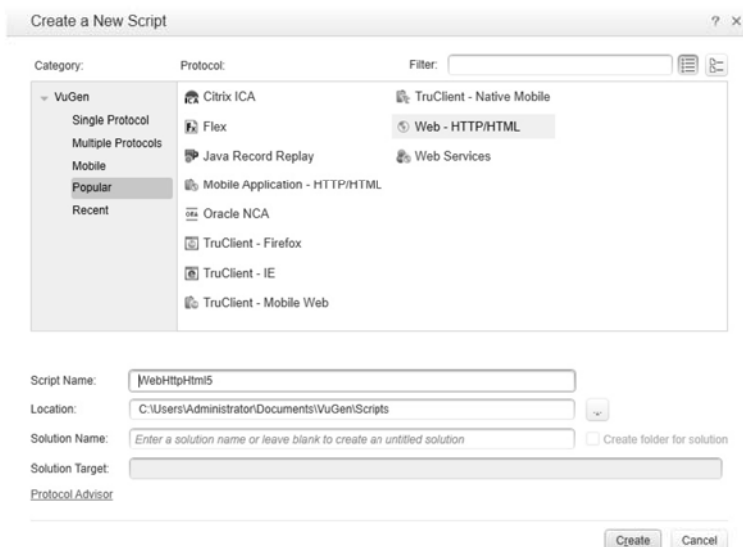


图 5-17 选择 Web-HTTP/HTML 协议

(2) 单击 Create 按钮，即可打开如图 5-18 所示的新建脚本窗口。

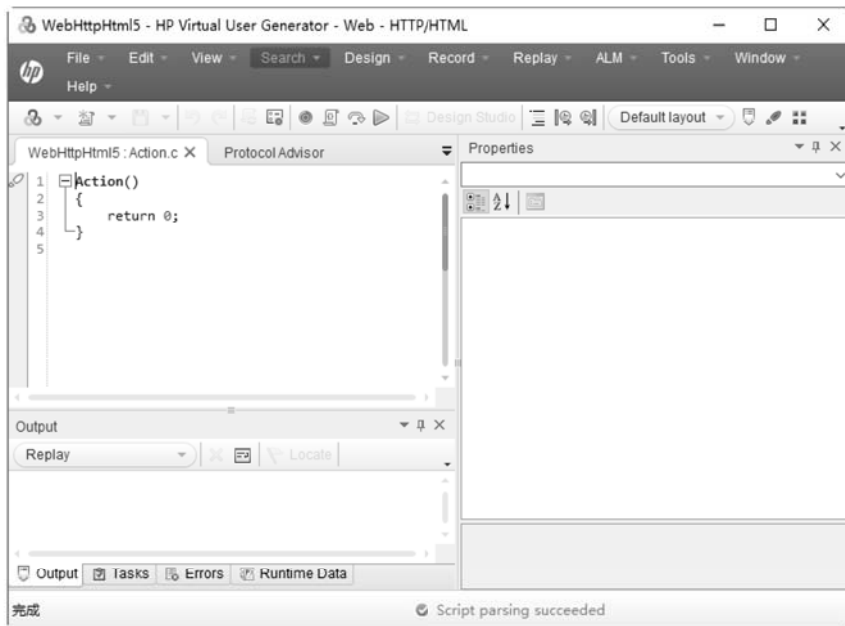


图 5-18 新建脚本窗口

(3) 如图 5-19 所示，选择 Record→Recording Options 命令，打开如图 5-20 所示的对话框。

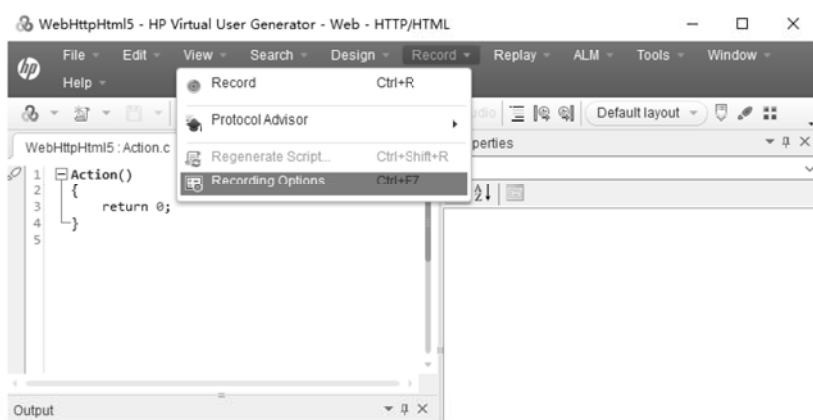


图 5-19 选择录制选项

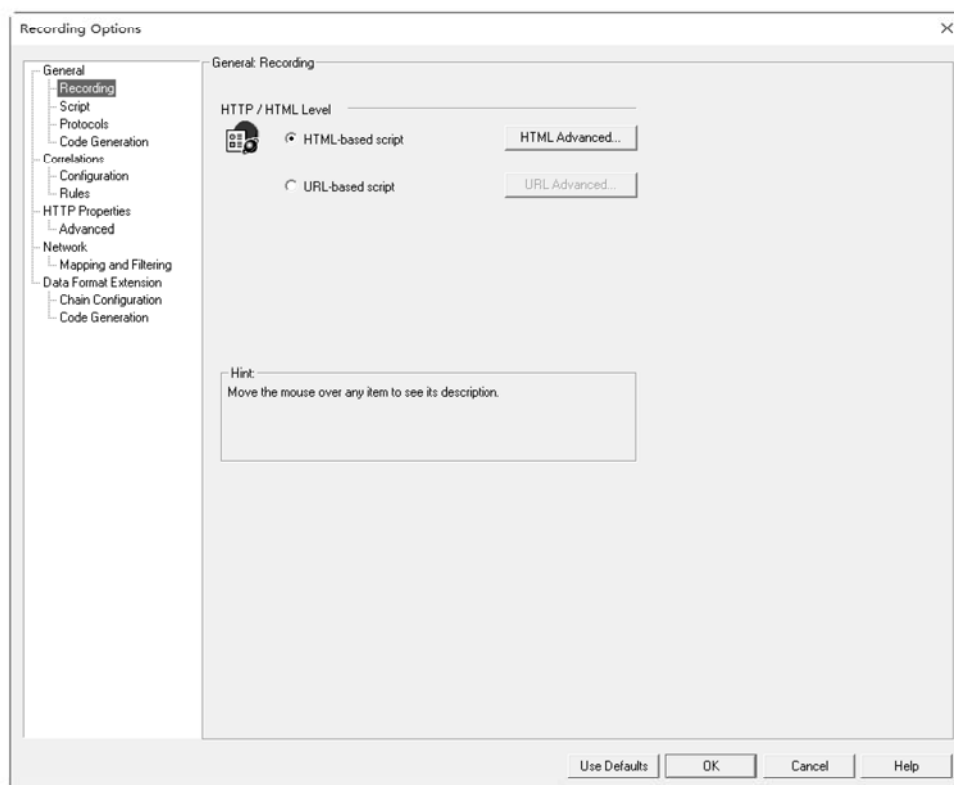


图 5-20 录制选项的设置界面

- (4) 设置录制选项，在 Capture level 下拉列表框中选择 Socket level data 选项，如图 5-21 所示。
- (5) 单击图 5-21 中的 New Entry 按钮，打开如图 5-22 所示的 Server Entry-Port Mapping

对话框，可以配置 New Entry 选项。

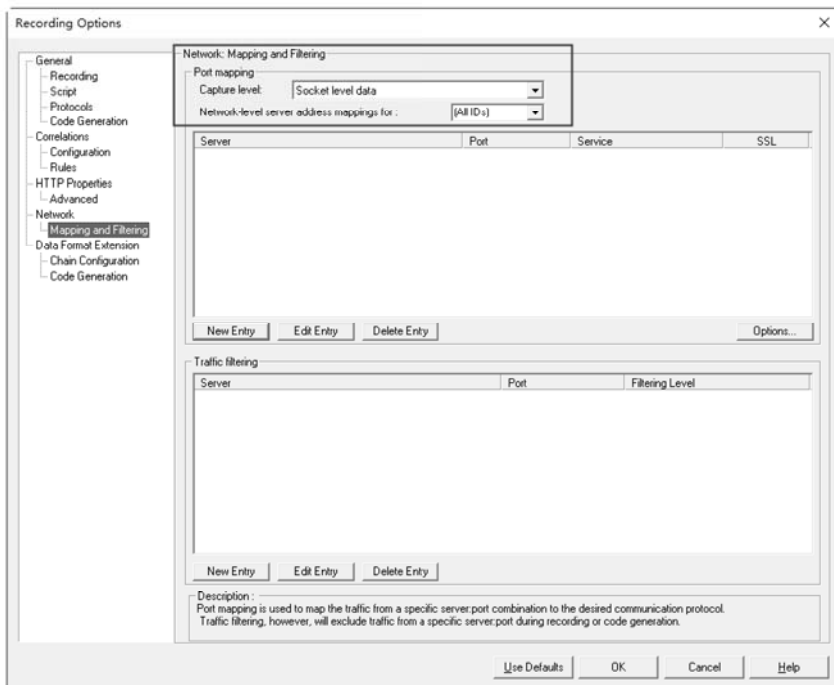


图 5-21 设置 Capture Level 为 Socket level data

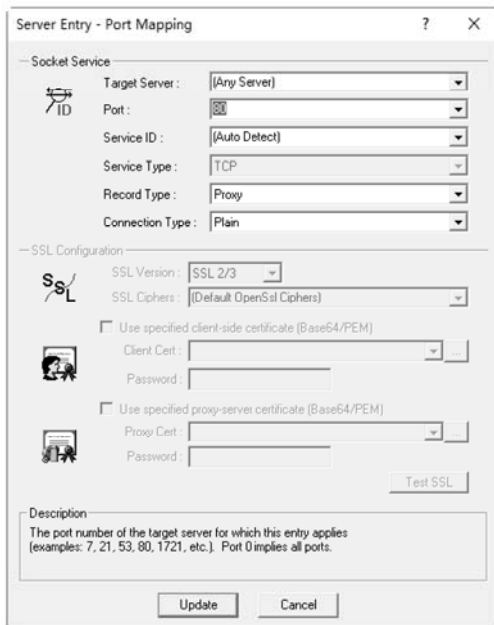


图 5-22 设置 New Entry 选项

说明：Target Server 为需要录制的地址，Port 设置为 80。

(6) 配置完成后保存即可。

5.3.4 手机端的设置

手机端设置的步骤如下。

(1) 连接上公共 Wi-Fi，如 OFFICE-Adviser。

(2) 设置 Wi-Fi 代理。其中 IP 为笔记本电脑的 IP，代理端口为 LoadRunner 设置的代理转发端口号，如图 5-23 所示。



图 5-23 设置代理

5.4 录制脚本

单击如图 5-7 所示的 Start Recording 对话框的 OK 按钮，即可开始录制。如图 5-24 所示是用公司内部 APP 环境地址成功录制的截图。



图 5-24 录制成功界面

失败的情况原因有很多，其中一个常见的原因是代理没有设置，如图 5-25 所示。

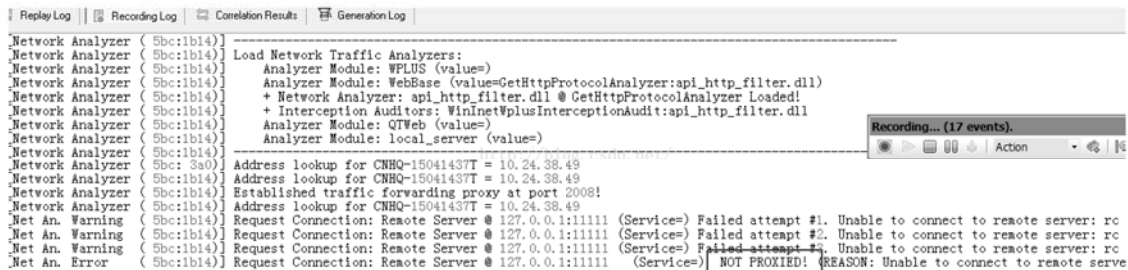


图 5-25 录制失败

5.4.1 修改/完善脚本

录制生成的脚本通常需要优化后，才能在后续场景实现目的。脚本中常用的优化类型有以下五种。

1. 插入事务点

```
lr_start_transaction("login");    ——开始
lr_end_transaction("login", LR_AUTO); ——结束
```


2. 添加集合点

```
lr_rendezvous("login");
```

3. 设置参数化

```
"Name=username", "Value={userName}", ENDITEM, -----粗体部分为参数化的内容  
lr_log_message("UserName:%s",lr_eval_string("{userName}")); -----参数化日  
志打印(查看参数化是否正确)
```

4. 设置文本检查点

```
web_reg_find("Search=Body", "Text=1029742081", LAST);
```

5. 设置字符有效长度(默认是 256)

```
web_set_max_html_param_len("102400");
```

5.4.2 变量参数化

在录制程序运行的过程中，Vugen(脚本生成器)自动生成了脚本以及录制过程中实际用到的数据，即脚本和数据是混在一起的，这不符合数据驱动测试的思想。

另外，进行网页性能测试时，经常需要对网页的输入操作进行压力测试，为模拟真正的使用场景，我们需要多用户并发操作，这时就需要脚本中的输入数据是变化的，而不是固定值。

基于上述两个原因的考虑，我们需要对脚本中的录入数据(比如登录名和密码)进行参数化，LoadRunner 对变量的参数化分为以下几个步骤完成：

- (1) 选择要进行参数化的字段，以变量名代替；
- (2) 编辑选定的变量，列举取值范围；
- (3) 设置参数遍历方式。

变量参数化可以直接在界面上添加编辑，具体步骤如下。

(1) 单击 P 图标或者按 Ctrl +L 键进入参数化设置界面，左侧列表中，会出现已经选定进行参数化的变量名称，如图 5-26 中的 name 和 psw。

(2) 单击要进行编辑的变量参数，例如 name，单击 Edit with Notepad 按钮，弹出记事本窗口，输入好用户名，保存，参数就设置完成了，如图 5-27 所示。

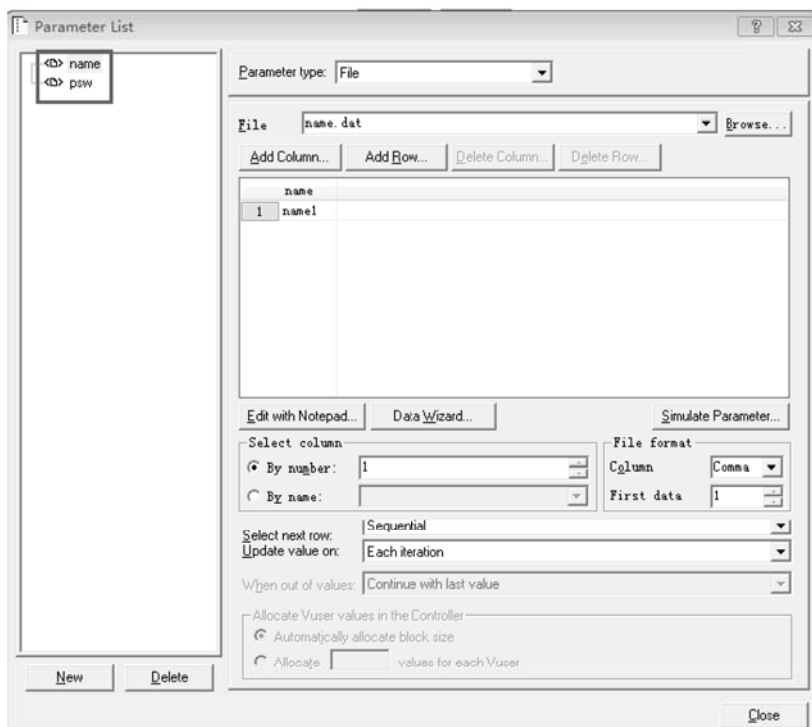


图 5-26 Parameter List 对话框

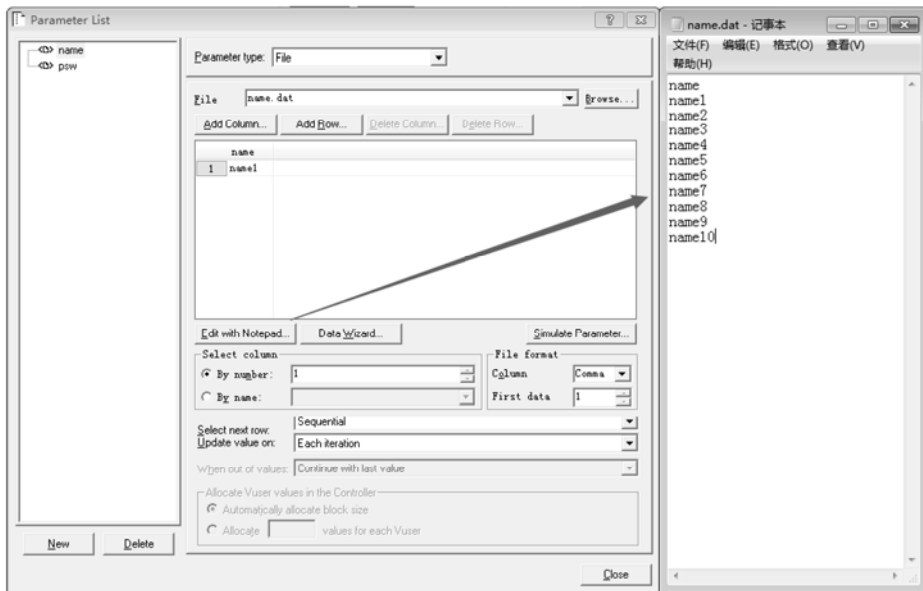


图 5-27 在 Parameter List 对话框中单击 Edit with Notepad 按钮

(3) 保存后，界面显示如图 5-28 所示。

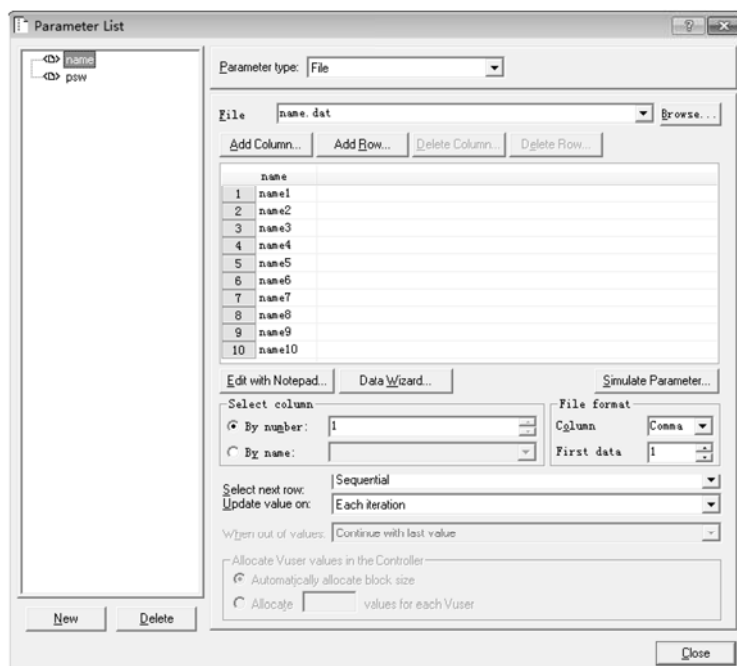


图 5-28 参数化设置界面(用记事本编辑后)

(4) 或者在界面上直接编辑, 添加行、列或者删除行、列, 如图 5-29 所示。

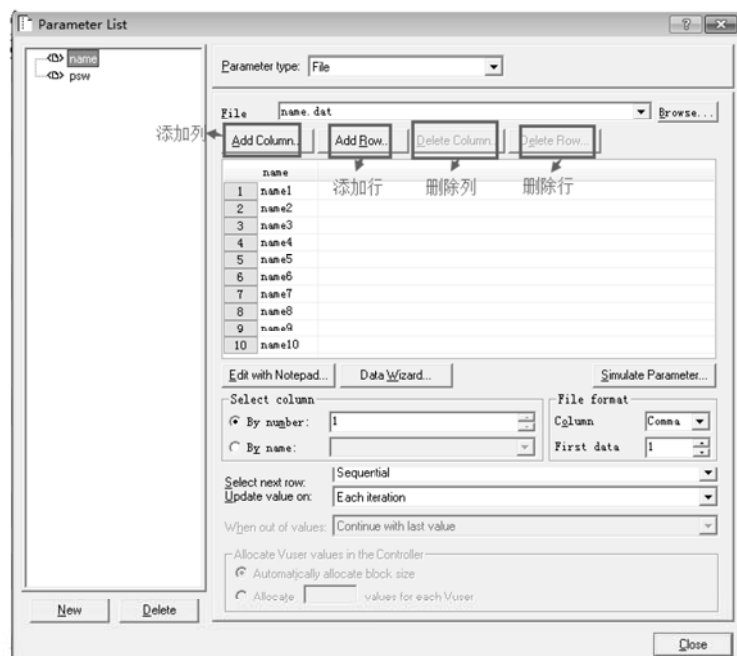


图 5-29 参数化设置界面(添加或删除行/列)

(5) 参数都编辑之后, 注意每个参数对应的名称, 以便后续使用。注释如图 5-30 所示。

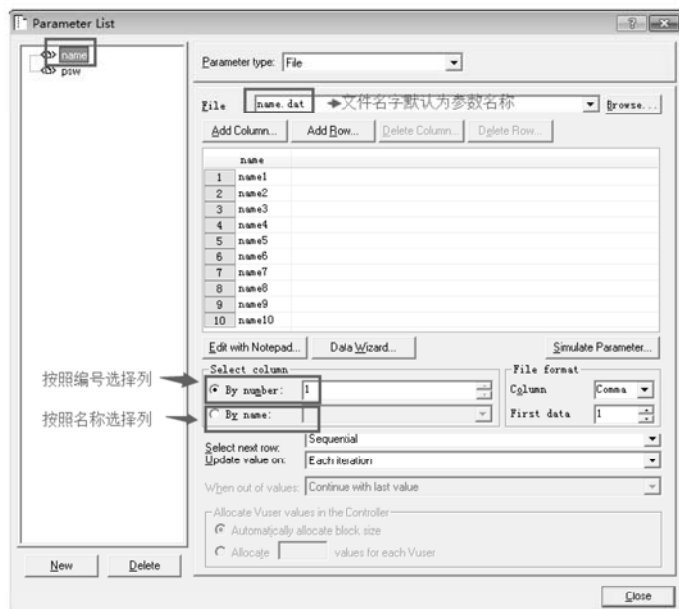


图 5-30 参数化设置界面(如何选择行、列)

(6) 也可以把所有的参数编辑在一个文件里面, 一行对应一个参数。当参数放在一个文件里面时, 设置处就要一一对应, 如图 5-31、图 5-32 和图 5-33 所示。

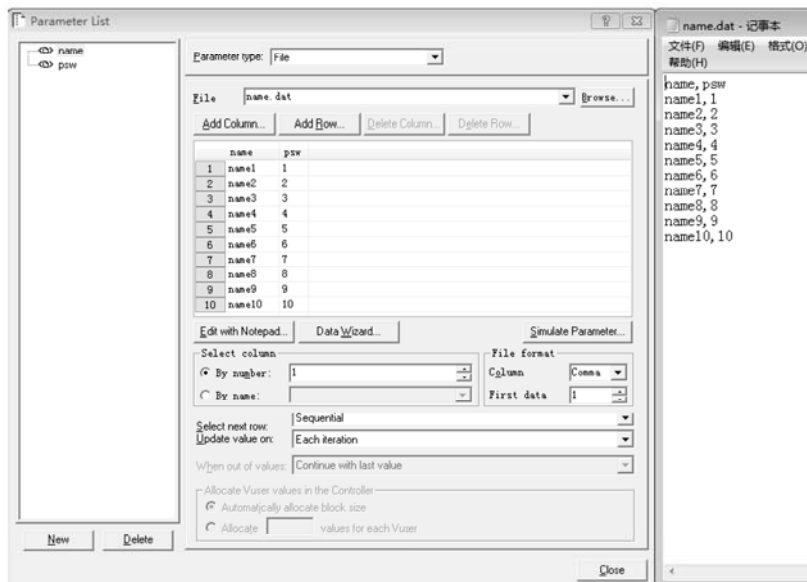


图 5-31 参数化设置界面(多个参数在一个文件中)

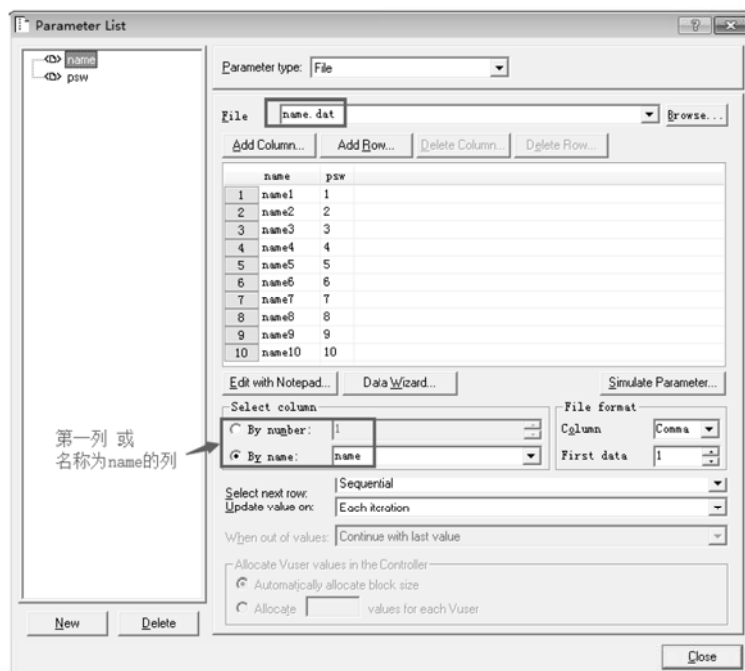


图 5-32 参数化设置界面(选择第 1 列)

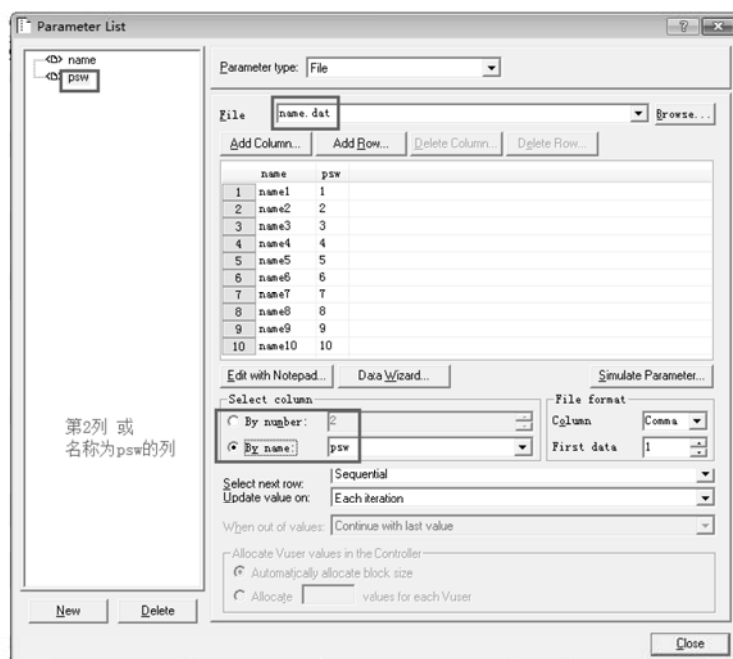


图 5-33 参数化设置界面(选择第 2 列)

5.4.3 导入文件

LoadRunner 支持用多种文件格式导入数据, 如 .dat、xls、Access 等。在此以 .dat 格式举例说明, 步骤如下。

(1) 在记事本中编辑变量的取值范围。在文件处选择已经编辑好的数据文件位置, 单击打开后, 文件的数据就自动导入进来了, 如图 5-34 和图 5-35 所示。

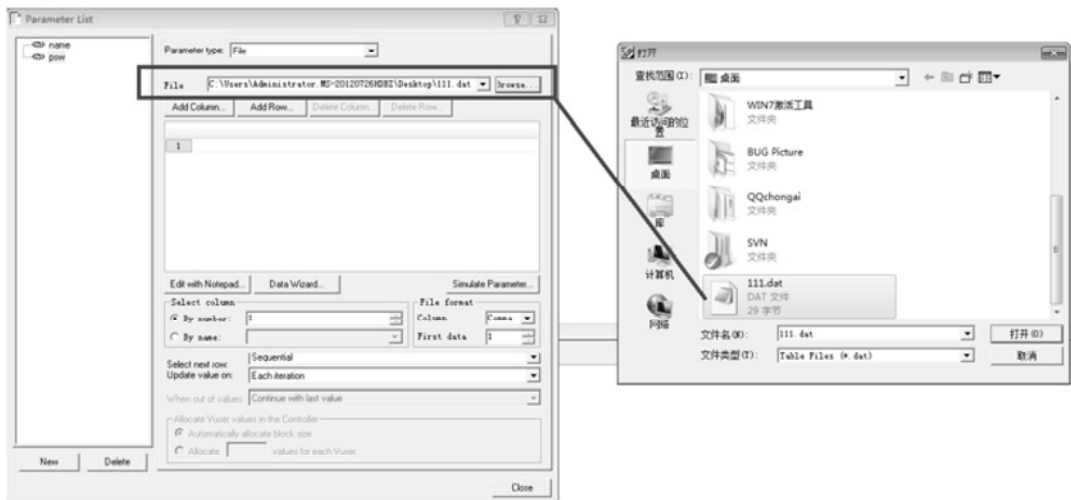


图 5-34 在参数化设置界面中导入数据(1)

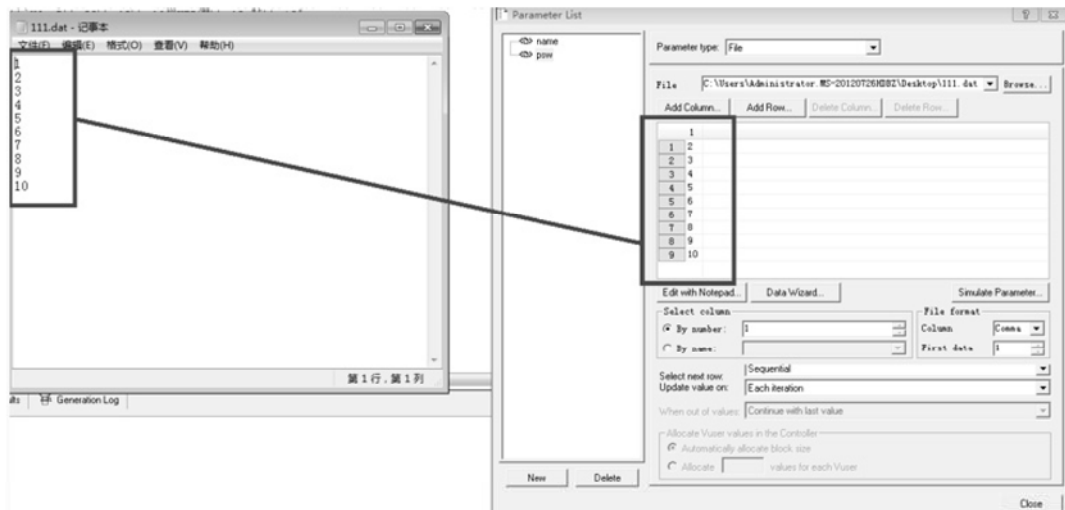


图 5-35 在参数化设置界面中导入数据(2)

(2) 编辑参数的取值范围后, 下一步需要设置数据取值方式与更新方法, 如图 5-36 所示。

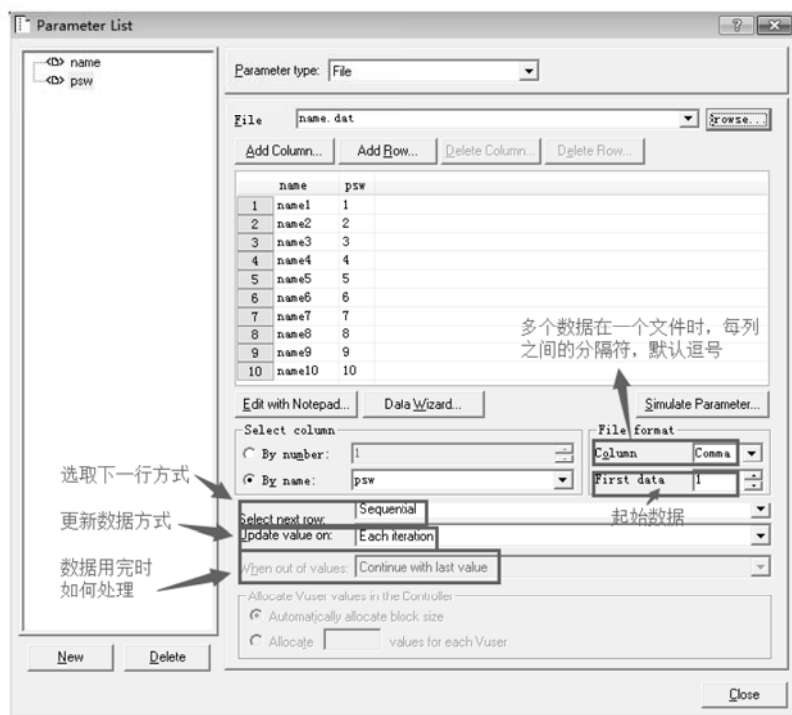


图 5-36 参数化设置界面(数据处理与格式)

这里涉及三个重要参数的设置。

① Select next row: 选取下一行方式, 如图 5-37 所示。

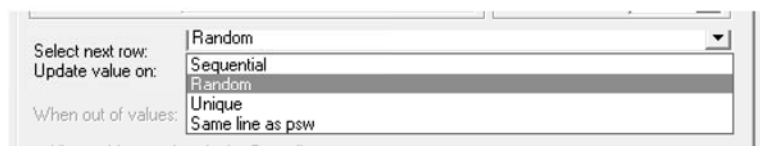


图 5-37 设置选取下一行的方式

- Sequential: 默认值, 按照参数化的数据顺序, 从上往下一个一个地取值。
- Random: 随机取, 参数化中的数据, 每次随机从中抽取数据。
- Unique: 唯一地向下取值, 只能被用一次。
- Same line as psw, 和 psw 列取同一行的值, (行相同)步调一致。

例如, 参数取值范围是:

a b c d e f g ...

现有 3 个用户(vu1, vu2, vu3)取值, 迭代 2 次。

- 顺序方式: vu1(a, b) vu2(a, b) vu3(a, b)
- 唯一方式: vu1(a, b) vu2(c, d) vu3(e, f)

如果是注册, 采用唯一方式。另外, 对于单用户来说, 顺序和唯一取值序列是相同的。

② Update value on: 更新数据方式, 如图 5-38 所示。

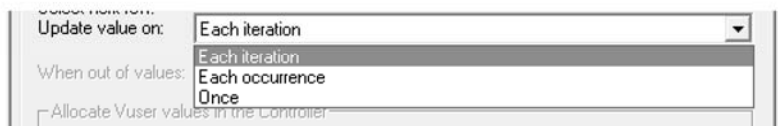


图 5-38 设置数据更新方式

- Each iteration: 默认, 每次迭代时更新取值(常用)。
- Each occurrence: 每次遇到该参数时取值。
- Once: 取值仅一次, 一次选择, 终身不变。

这里的迭代次数, 指的是 Action 运行的次数。vuser_init 和 vuser_end 只运行一次, 如图 5-39 所示。

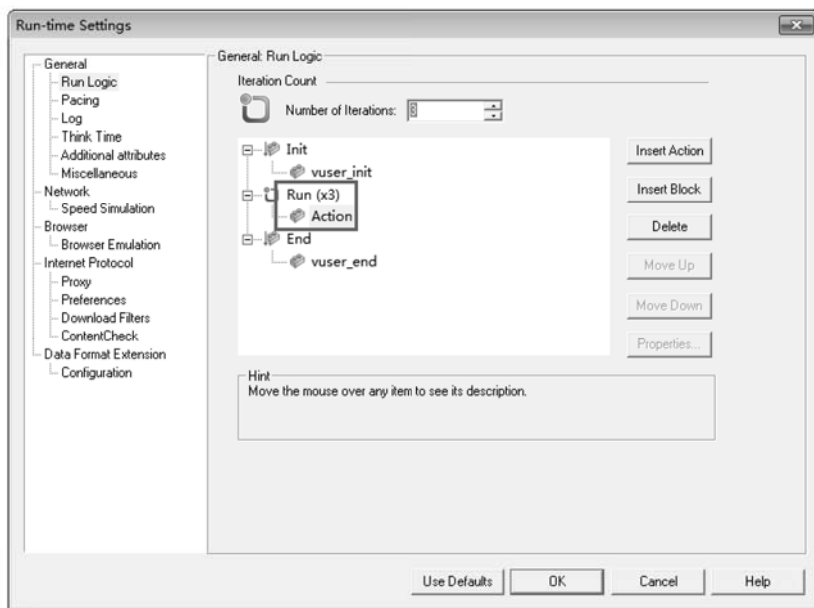


图 5-39 在 Run-time Settings 对话框中设置迭代次数

③ When out of values: 数据不足时的处理方式。

在①中选择 Unique 时才需要考虑这个选项, 表示取值越界后的处理方式, 如图 5-40

所示。

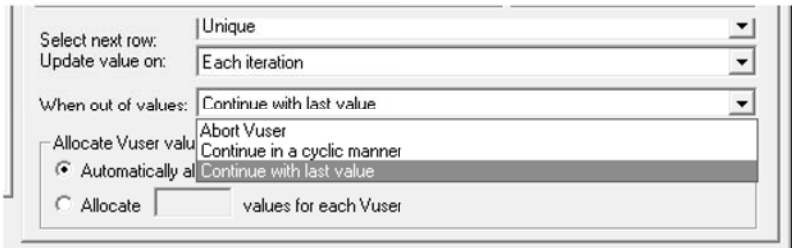


图 5-40 When out of values 的设置

- Abort Vuser：放弃虚拟用户，不再取值。
- Continue in a cyclic manner：以循环的方式继续，当参数化文件中的值取完最后一个值后，又从参数化文件的第一行开始取值。
- Continue with last value：当参数化文件中的值取完最后一个值后，一直持续到最后

一个值。

为帮助理解上述取值遍历方式，举例说明。

前提条件如下。

虚拟用户：vu1、vu2、vu3；

数据：a、b、c；

脚本中参数出现 3 次，迭代 3 次。

为说明遍历方式，Select next row 取 Unique 时，给数据表中加入数据 d、e、f、g、h、i。

Select next row 和 Update Value on 两个参数组合的九种情况，结果取值如表 5-1 所示。

表 5-1 参数组合

序号	Select next row	Update Value on	结 果
1	Sequential	Each Iteration	第一次迭代，无论参数任何时候出现，vu1、vu2、vu3 取 a； 第二次迭代，无论参数任何时候出现，vu1、vu2、vu3 取 b； 第三次迭代，无论参数任何时候出现，vu1、vu2、vu3 取 c
2	Sequential	Each Occurrence	第 N 次迭代，参数第一次出现，vu1、vu2、vu3 取 a； 第 N 次迭代，参数第二次出现，vu1、vu2、vu3 取 b； 第 N 次迭代，参数第三次出现，vu1、vu2、vu3 取 c
3	Sequential	Once	第 N 次迭代，无论参数任何时候出现，vu1 取 a，vu2 取 b，vu3 取 c

续表

序号	Select next row	Update Value on	结 果
4	Random	Each Iteration	第 N 次迭代, 无论遇到该参数多少次, vu1 都只取 a, 或者 b, 又或者 c, 本次迭代不再更新; 第 N 次迭代, 无论遇到该参数多少次, vu2 都只取 a, 或者 b, 又或者 c, 本次迭代不再更新; 第 N 次迭代, 无论遇到该参数多少次, vu3 都只取 a, 或者 b, 又或者 c, 本次迭代不再更新; 在 N+1 次迭代, 每个 vu 重新随机抽取数据
5	Random	Each Occurrence	第 N 次迭代, 第一次遇到该参数, vu1、vu2、vu3 在 a、b、c 中随机抽取一个; 第 N 次迭代, 第二次遇到该参数, vu1、vu2、vu3 重新在 a、b、c 中随机抽取一个; 第 N 次迭代, 第三次遇到该参数, vu1、vu2、vu3 重新在 a、b、c 中随机抽取一个; 在 N+1 次迭代, 每个 vu 继续保持每遇到一次参数, 就重新抽取一次数据
6	Random	Once	第 N 次迭代, 无论遇到该参数多少次, vu1 都只取 a, 或者 b, 又或者 c; 第 N 次迭代, 无论遇到该参数多少次, vu2 都只取 a, 或者 b, 又或者 c; 第 N 次迭代, 无论遇到该参数多少次, vu3 都只取 a, 或者 b, 又或者 c; 在 N+1 次迭代, 每个 vu 不会重新抽取数据
7	Unique	Each Iteration	第一次迭代, 无论参数出现多少次, vu1 取 a, vu2 取 d, vu3 取 g; 第二次迭代, 无论参数出现多少次, vu1 取 b, vu2 取 e, vu3 取 h; 第三次迭代, 无论参数出现多少次, vu1 取 c, vu2 取 f, vu3 取 i
8	Unique	Each Occurrence	第一次迭代, 第一次出现该参数, vu1 取 a, vu2 取 d, vu3 取 g; 第一次迭代, 第二次出现该参数, vu1 取 b, vu2 取 e, vu3 取 h; 第一次迭代, 第三次出现该参数, vu1 取 c, vu2 取 f, vu3 取 i

续表

序号	Select next row	Update Value on	结 果
9	Unique	Once	无论进行多少次迭代，无论参数任何时候出现，vu1 取 a，vu2 取 b，vu3 取 c； 数据 d、e、f、g、h、i 不取

注：

(1)Select next row 取 unique 时，必须注意数据表有足够多的数，在此我们给数据表中加入数据 d、e、f、g、h、i；

(2)实际使用中可能用不到很复杂的组合情况。

5.4.4 Parameterization 实例

对于 5.3 节中录制的登录脚本，选择登录用户名和密码对其参数化。选择需要参数化的字段，右击，在弹出的快捷菜单中选择 Replace with a Parameter 命令，如图 5-41 所示，弹出 Select or Create Parameter 对话框，在 Parameter name 处输入变量名 name，单击 OK 按钮，对密码的参数化重复以上的操作。



图 5-41 选择 Replace with a Parameter 命令

把用户名改成变量 name，密码改成变量 psw，如图 5-42 所示。

```

web_submit_data("login",
  "Action=http://192.168.1.100:80/person/personal/login",
  "Method=POST",
  "TargetFrame=",
  "RecContentType=application/json",
  "Referer=http://192.168.1.100:80/person/",
  "Snapshot=t141.inf",
  "Mode=HTML",
  ITEMDATA,
  "Name=username", "Value={name}", ENDITEM, |
  "Name=password", "Value={psw}", ENDITEM,
  "Name=randCheckCode", "Value=13", ENDITEM,
  LAST);

```

图 5-42 将用户名和密码替换成变量

下面设置这两个参数的取值：

```

name: name1-6;
psw: psw1-6;

```

Select next row 参数选择 Sequential, Update value on 参数选择 Each iteration。并设置 3 次迭代, 3 用户。

我们的登录脚本在 vuser_init.c 中, 与迭代次数无关, 3 用户执行脚本时, 取值如表 5-2 所示(以 3 用户为例)。

表 5-2 执行登录脚本

用户	name	psw
vu1	Name1	Psw1
vu2	Name2	Psw2
vu3	Name3	Psw3

其他数据未取到。

5.4.5 关联

我们通过一个场景来理解关联的含义和作用。

比如, 我们去坐飞机, 登机前需要在检票点出示机票, 检票登机。那么检票人员会检查哪些东西呢? 机票是否真实, 航班是否正确等信息, 验证通过即可登机。

过了几天我们又去坐飞机, 还是拿同样一张票去登机, 检票人员再一次核对信息, 发现机票已经过期了, 自然就不能成功登机了。

在脚本中也存在大量类似的情况, 录制的时候, 服务器会给一个唯一的认证码来进行操作, 当再次回放脚本的时候服务器会给一个全新的认证码, 而脚本录制是死的, 还是拿

老的认证码提交，从而导致脚本执行失败。

例如，常见系统中的登录功能，在登录后服务器会返回 SessionID，登录后的操作都需要提交该 SessionID 确认身份。使用 Vugen 录制时，将会记录服务器返回的 SessionID 并且原封不动地在下一个请求中发给服务器，如图 5-43 所示。



图 5-43 登录成功过程

待到回放的时候，服务器会在接收到用户名和密码后返回新的 SessionID，而脚本仍然发送旧的 SessionID 给服务器，最终因 SessionID 错误，导致脚本回放失败，如图 5-44 所示。

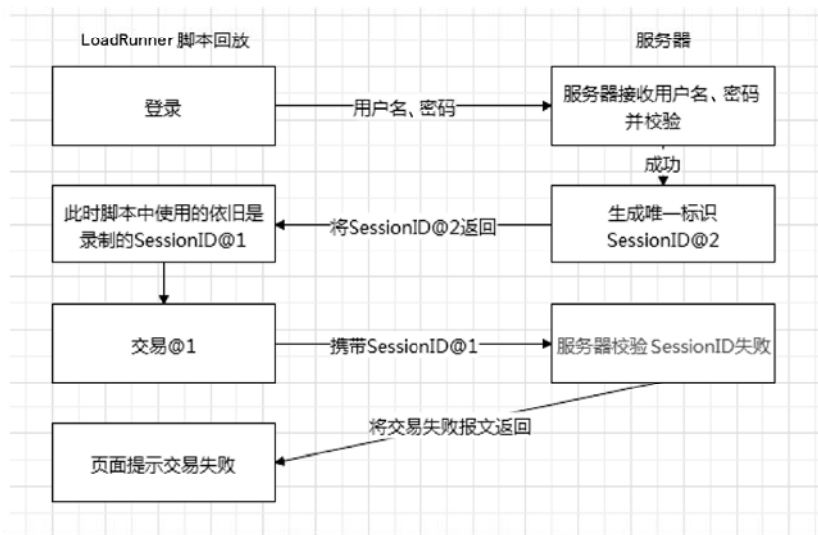


图 5-44 登录失败过程

录制和回放过程如下：

为了确保脚本回放成功，我们需要获得服务器每次返回的动态 SessionID，再将这个动态数据发回给服务器。而关联能够帮助我们将服务器返回的数据进行处理并保存为参数。

通常情况下我们使用关联的步骤流程如下：

在脚本回放过程中，客户端发出请求，通过关联函数所定义的左右边界值(也就是关联规则)，在服务器所响应的内容中查找，得到相应的值，以变量的形式替换录制时的静态值，从而向服务器发出正确的请求。

简单地说，关联就是对服务器的返回做处理的过程。它分为三种方式实现：

- 自动关联；
- 手动关联；
- 一边录制一边关联。

下面重点介绍自动关联和手动关联。

1. 自动关联

自动关联是 Vugen 提供的自动扫描关联处理策略，它的原理是对同一个脚本运行和录制时的服务器返回进行比较，来自动查找变化的部分，并且提示是否生成关联。自动关联的步骤如下：

- (1) 开启自动关联选项。
- (2) 录制脚本。
- (3) 回放脚本。
- (4) 系统自动弹出检测关联对话框，或手动启动关联检测对话框。
- (5) 查看系统检测出的关联点进行关联设置。
- (6) 回放脚本检查关联的正确性。

具体示例步骤如下。

(1) 启动 Vugen，通过以下两种方式打开 Recording Options 对话框：一是选择 Tools→Recording Options 命令，如图 5-45 所示；二是在 Start Recording 对话框中单击 Options 按钮，如图 5-46 所示。

(2) 启动 Recording Options 后，在打开的 Recording Options 对话框中，选中 Enable correlation during recording 复选框，以启动自动关联，如图 5-47 所示。

(3) 录制登录退出脚本，如图 5-48 所示。

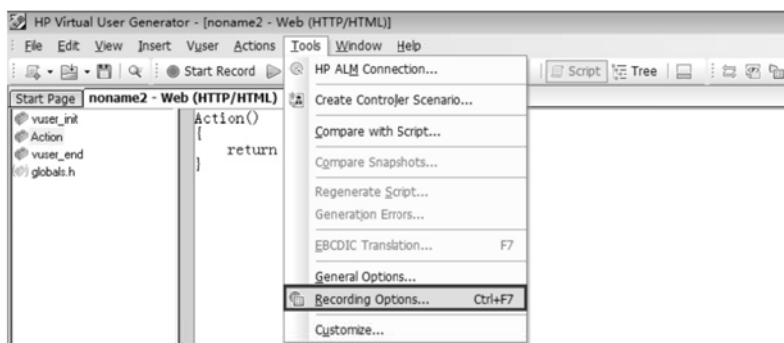


图 5-45 选择 Recording Options 命令

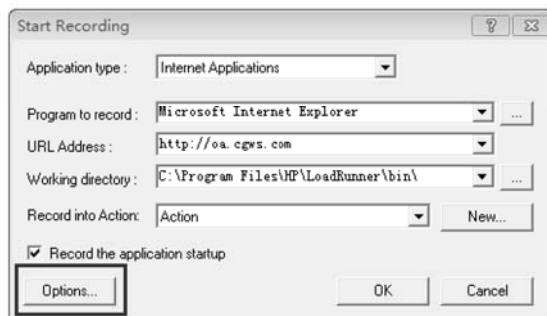


图 5-46 单击 Options 按钮

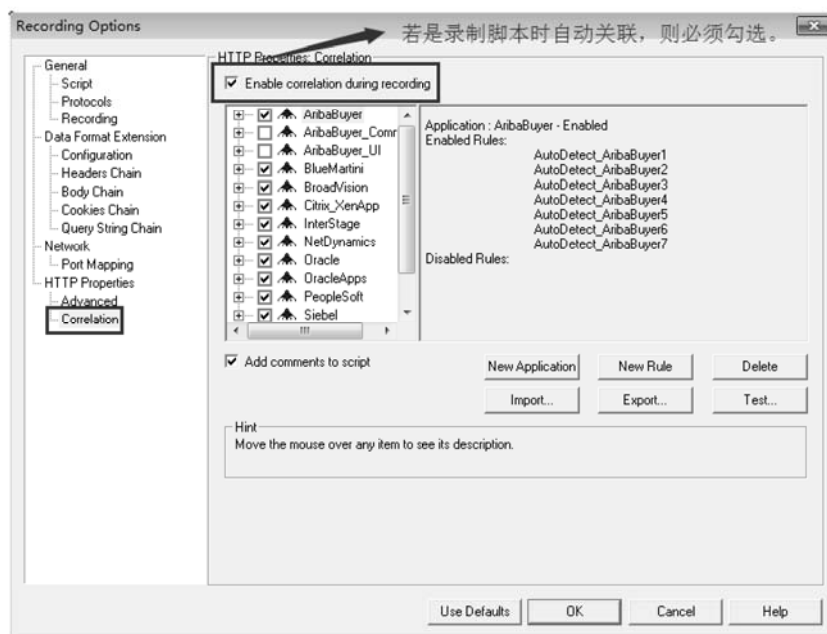


图 5-47 启动自动关联


```

Action()
{
    web_url("WebTours",
    web_custom_request("q.cgi",
        "URL=http://masterconn.qq.com/q.cgi",
        "Method=POST",
        "Resource=1",
        "Referer=",
        "Snapshot=t39.inf",
        "EncType=",
        "BodyBinary=\\x00\\x02\\x00X\\x00\\x00\\x00\\x01\\x00\\x00\\x00j\\x00\\x00\\x07\\xD
        LAST);

    lr_think_time(5);

    web_submit_data("login.pl",
        "Action=http://127.0.0.1:1080/WebTours/login.pl",
        "Method=POST",
        "ContentType=text/html",
        "Referer=http://127.0.0.1:1080/WebTours/nav.pl?in=home",
        "Snapshot=t40.inf",
        "Mode=HTML",
        ITEMDATA,
        "Name=userSession", "Value=116454.604050223fQQAVDDpzQfIDDDDDfQtAplAcVHf", ENDITEM,
        "Name=username", "Value=bigbig", ENDITEM,
        "Name=password", "Value=0000000", ENDITEM,
        "Name=JSFormSubmit", "Value=on", ENDITEM,
        "Name=login.x", "Value=31", ENDITEM,
        "Name=login.y", "Value=3", ENDITEM,
        LAST);

    web_image("SignOff Button",
    return 0;
}

```

图 5-48 录制登录退出脚本

(4) 此时，回放脚本出错，如下所示。

```

Action.c(41): Error -27987: Requested image not found
[MsgId: MERR-27987]
Action.c(41): web_image("SignOff Button") highest severity level was "ERROR",
0 body bytes, 0 header bytes
[MsgId: MMSG-26388]

```

(5) 在菜单栏中选择 Vuser → Scan Script for Correlations 命令(或使用快捷键 Ctrl+F8)，弹出如图 5-49 所示的对话框。

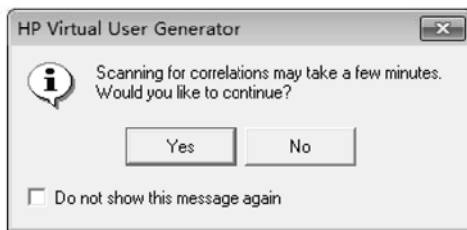


图 5-49 选择是否扫描相关性

(6) 单击 Yes 按钮，开始扫描相关性，可能需要几分钟时间。选中出现的需要关联的提示，单击 Correlate 按钮，提示前分别出现对号。回放脚本，无报错，自动关联成功，如图 5-50 所示。



图 5-50 自动关联成功

2. 手动关联

自动关联是 LoadRunner 通过对比录制和回放时服务器响应的不同，而提示用户是否进行关联，用户可自己创建关联规则，这个功能可以方便地使我们获得需要关联的部分，但同时也存在一定的问题，如：自动关联所检测到的关联点不一定真的需要进行关联，这需要我们根据实际情况进行判断；有些需要关联的动态数据自动关联无法找到，这时就需要做手动关联。

手动关联的主要步骤如下。

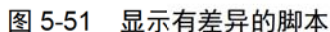
- (1) 录制测试脚本，录制两遍。
- (2) 使用 WinDiff 工具找出两次脚本的不同，判断是否需要进行关联。
- (3) 确定插入关联的位置。
- (4) 在 VIEW TREE 中使用 web_reg_save_param 函数手动建立关联。
- (5) 将脚本中有用到关联的数据，用参数代替。
- (6) 验证关联的正确性。

下面结合实例详细介绍。

- (1) 录制测试脚本，录制两遍。相同的操作，录制两份，分别保存。
- (2) 使用 WinDiff 工具协助找出需要关联的数据。在第二份脚本中，选择 Vugen 的 Tools→Compare with Vuser，并选择第一份脚本。
- (3) WinDiff 会开启，同时显示两份脚本，并显示有差异的地方。WinDiff 会以一整行黄色标示有差异的脚本，如图 5-51 所示。

查看两份脚本中有差异的部分，每一个差异都可能是需要做关联的地方。找到不同的部分后，复制，然后打开 Recording Log 或是 Generation Log，(一般情况录制脚本时选择 Single Protocol(单协议)，在 Recording Log 里找；录制脚本时选择 Multiple Protocol(多协议)，在 Generation Log 里找)按 Ctrl+F 键，在查找的界面中粘贴差异部分的内容，单击“查找”按钮找到后，查看该部分的信息，确认是客户端的请求信息还是服务器回应的信息。

如果出现在\$\$\$\$\$\$ Request Header For Transaction With Id 3 Ended \$\$\$\$\$\$这个部分，那证明是客户端发出的请求，这里是不需要做关联的。



① 记录这个不同数据之前的内容和之后的内容。

② 记录这个不同数据出现的位置，是 Header 还是 Body。(如果出现在如“*****

右边界

左边界

(4) 确认插入关联的位置。在日志中找到了两次脚本的不同点的位置，根据这个位置，

再确定是在哪个请求之后产生的，也就是说要定位发生不同点的 Response(响应)是由哪个 Request(请求)产生的，找到了这个请求的函数位置，就知道要往哪里做关联了。

一般情况下关联函数写到发出请求的函数之前，服务器已经回应的要进行关联的数据脚本之后，在发出请求之前进行关联操作(调用 web_reg_save_param 函数中存放得到的动态内容的参数名称)(在发出请求之前就要将有变化的地方进行参数化，所以要放在发出请求之前)。

(5) 插入关联函数。在插入关联函数前，先介绍关联函数 web_reg_save_param。先举一个 web_reg_save_param 函数的例子。

```
web_reg_save_param ("userSession ",  
    "LB=input type=hidden name=userSession value= ",  
    "RB=> ",  
    "Search=Body",  
    LAST);
```

参数说明如下。

- ParamName: 存放得到的动态内容的参数名称。
- list of Attributes: 其他属性。包括: Notfound、LB、RB、RelFrameID、Search、ORD、SaveOffset、Convert、SaveLen。属性值不分大小写。
- LB(Left Boundary): 返回信息的左边界字符串。该属性必须有，并且区分大小写。
- RB(Right Boundary): 返回信息的右边界字符串。该属性必须有，并且区分大小写。
- Search: 返回信息的查找范围。可以是 Headers、Body、Noresource、All(默认)。该属性值可有可无。

插入关联函数的方法如下。

① 将 Vugen 切换到 View tree 模式下，在左边的列表中，找到在上一步发出请求的函数并右击，在弹出的快捷菜单中选择 Insert before 命令。

② 在弹出的 Add step 对话框的 Find function 中输入 web_reg_save_param，单击 OK 按钮，在 Parameter name 中输入关联函数的名称，这里最好有含义: userSession。

③ 在 Left boundary 中输入刚才记录下的不同点字符串的左面的几个字符，定义左边界: input type=hidden name=userSession value=。

④ 在 Right boundary 中输入刚才记录下的不同点字符串的右面的几个字符，定义右边界，在 Search in 中选择 body。单击 OK 按钮。

⑤ 回到脚本编辑模式，查看该函数插入是否正确。在发出请求的函数前应该看到:

```
web_reg_save_param ("userSession ",  
    "LB= input type=hidden name=userSession value=",
```

```
"RB=>",
"Search=Body",
LAST);
```

(6) 将脚本中有用到关联的数据, 用参数代替。如发出请求的参数如下, 那么将原来服务器返回的动态值使用 {userSession} 来替换:

```
web_submit_data("login.pl",
    "Action=http://127.0.0.1:1080/WebTours/login.pl",
    "Method=POST",
    "RecContentType=text/html",
    "Referer=http://127.0.0.1:1080/WebTours/nav.pl?in=home",
    "Snapshot=t3.inf",
    "Mode=HTML",
    ITEMDATA,
    "Name=userSession", "Value={userSession}", ENDITEM,
    "Name=username", "Value=erer", ENDITEM,
    "Name=password", "Value=123456", ENDITEM,
    "Name=JSFormSubmit", "Value=off", ENDITEM,
    "Name=login.x", "Value=79", ENDITEM,
    "Name=login.y", "Value=12", ENDITEM,
    LAST);
```

(7) 回放脚本, 验证关联的正确性。

5.5 回放脚本

通过录制一系列典型用户操作(例如预订机票), 已经模拟了真实用户操作。将录制的脚本合并到负载测试场景之前, 回放此脚本以验证其是否能够正常运行。回放过程中, 可以在浏览器中查看操作并检验是否一切正常。如果脚本不能正常回放, 可能需要添加关联。回放脚本之前, 可以配置运行时设置, 用来帮助设置 Vugen 的行为。

5.5.1 设置运行时行为

通过 LoadRunner 运行时设置(Runtime Settings), 可以模拟各种真实用户活动和行为。例如, 可以模拟一个对服务器输出立即做出响应的用户, 也可以模拟一个先停下来思考, 再做出响应的用户。另外还可以配置运行时设置来指定 Vugen 应该重复一系列操作的次数和频率。有一般运行时设置和专门针对某些 Vugen 类型的设置。例如, 对于 Web 仿真, 可以指示 Vugen 在 Netscape 而不是 Internet Explorer 中回放脚本。

适用于所有类型脚本的一般运行时设置。其中包括以下几项。

- 运行逻辑(Run Logic)。即重复次数。
- 步(Pacing)。即两次重复之间的等待时间。
- 思考时间(Think Time)。即用户在各步骤之间停下来思考的时间。
- 日志(Log)。即希望在回放期间收集的信息的级别。

注意,也可以在 LoadRunner Controller(控制器)中修改运行时设置。

1. 打开运行时设置对话框

(1) 确保 Task(任务)窗格出现(如果未出现,请单击 Task 按钮)。单击 Task 窗格中的验证回放。

(2) 在说明窗格内的标题运行时设置下单击打开 Runtime Settings(运行时设置)超链接,如图 5-53 所示。也可以按 F4 键或单击工具栏中的运行时设置按钮。这时将打开运行时设置对话框。

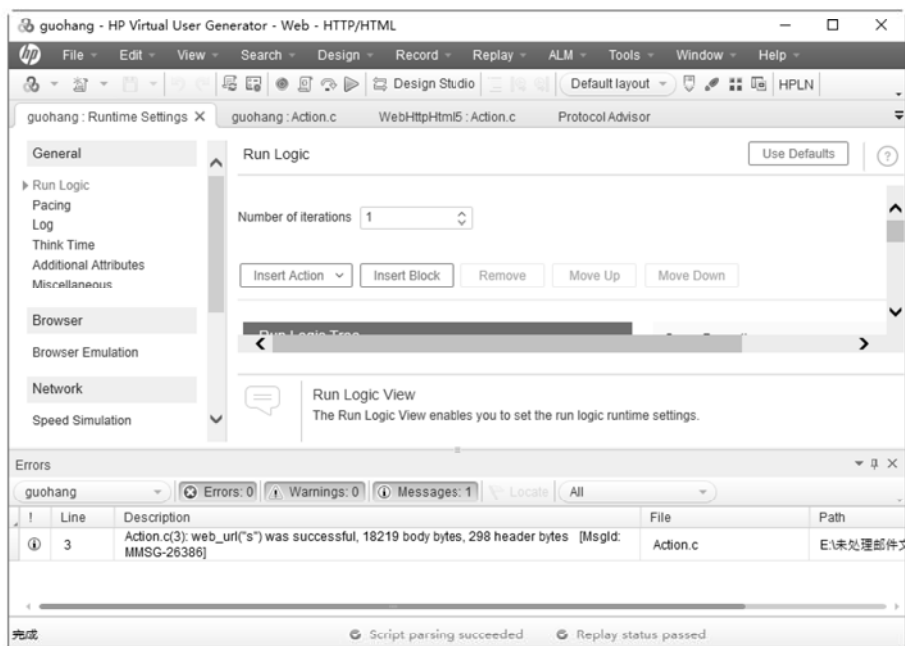


图 5-53 运行时设置

2. 设置运行逻辑

在左窗格中选择 Run Logic(运行逻辑)节点,设置迭代次数或说连续重复活动的次数,将迭代次数设置为 2,如图 5-54 所示。



图 5-54 设置迭代次数

3. 迭代时间间隔的设置

在左窗格中选择 **Pacing(步)**节点，此节点用于控制迭代时间间隔，如图 5-55 所示。可以指定一个随机时间，这样可以准确模拟用户在操作之间等待的实际时间。但使用随机时间间隔时，很难看到真实用户在重复之间恰好等待 60 秒的情况。选择第三个单选按钮并选择下列设置：时间随机，间隔 60.000 到 90.000 秒。

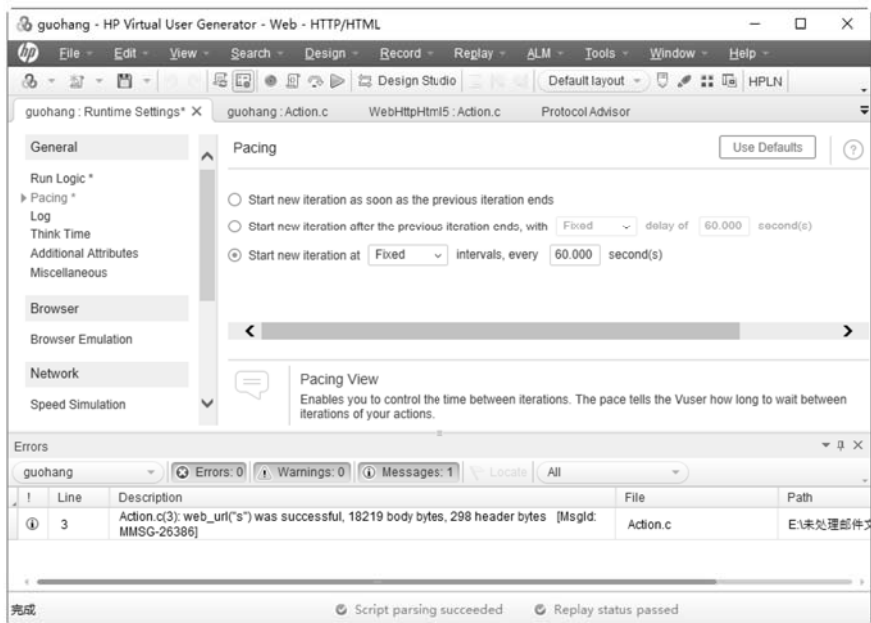


图 5-55 设置迭代时间间隔

4. 配置日志设置

在左窗格中选择 **Log(日志)**节点，可以配置日志，如图 5-56 所示。日志设置指出要在运

行测试期间记录的信息量, 开发期间, 可以选择启用日志记录来调试脚本, 但在确认脚本运行正常后, 只能记录错误或禁用日志功能。选择扩展日志并启用参数替换。

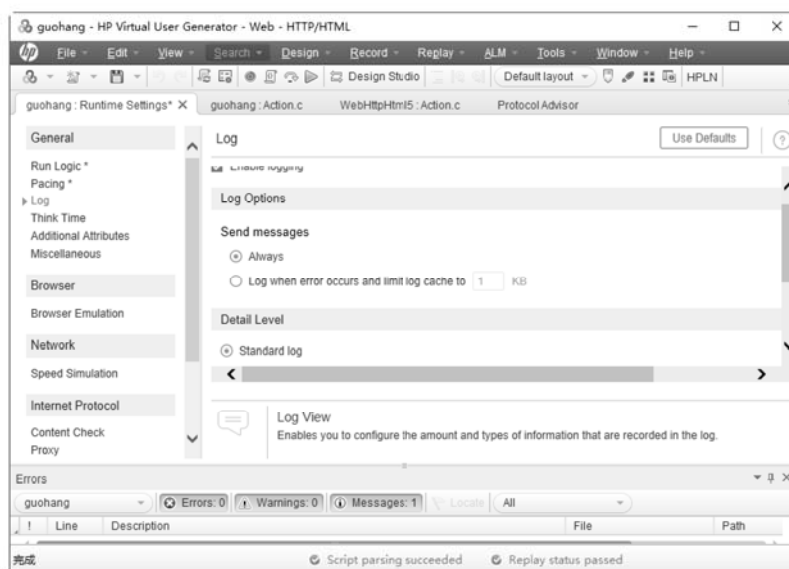


图 5-56 配置日志

5. 查看思考时间设置

如图 5-57 所示, 可以在左窗格中单击 Think Time(思考时间)节点。

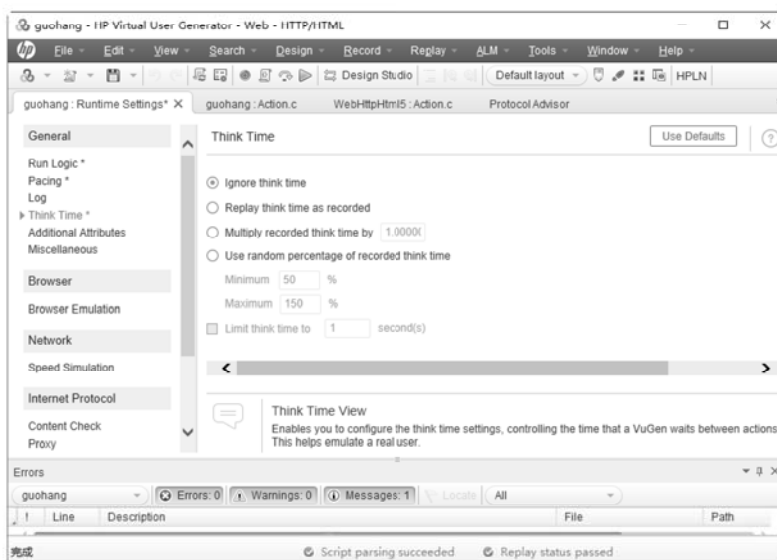


图 5-57 设置 Think Time 节点

备注：请勿进行任何更改。可以在 Controller(控制器)中设置思考时间。注意，在 Vugen 中运行脚本时速度很快，因为它不包含思考时间。

5.5.2 实时查看脚本运行情况

回放录制的脚本时，Vugen 的运行时查看器功能实时显示 Vugen 的活动情况。默认情况下，Vugen 在后台运行测试，不显示脚本中的操作动画。但在本教程中，将学习让 Vugen 在查看器中显示操作，从而能够看到 Vugen 如何执行每一步。查看器不是实际的浏览器，它只显示返回到 Vugen 的页面快照。此处就是回放时显示的快照，否则没有快照。

(1) 选择 Tools(工具)→Options(选项)命令，打开 Options 对话框，然后选择 General(常规)选项卡，如图 5-58 所示。

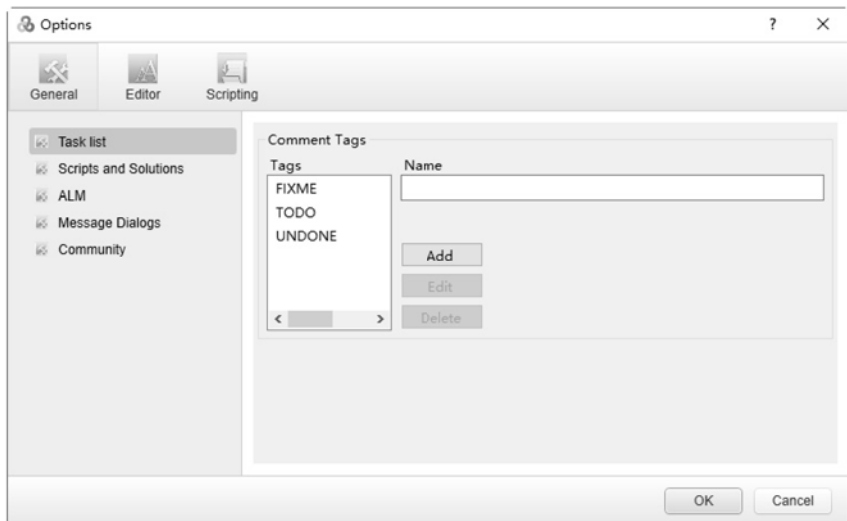


图 5-58 设置显示选项

(2) 单击 OK 按钮关闭 Options 对话框。在任务栏(Task)中单击验证回放(2.Replay)，然后单击说明窗格底部的开始回放按钮，或者按 F5 键，或者单击工具栏中的运行按钮，如图 5-59 所示。

(3) 如果打开选择结果目录的对话框，并询问要将结果文件保存到何处，请接受默认名称并单击 OK 按钮。稍后 Vugen 将打开运行时查看器，并开始运行脚本视图或树视图中的脚本(具体取决于上次打开的脚本)。在运行时查看器中，可以直观地看到 Vugen 的操作。注意回放的步骤顺序是否与录制的步骤顺序完全相同。回放结束后，会出现一个消息框提示是否扫描关联。

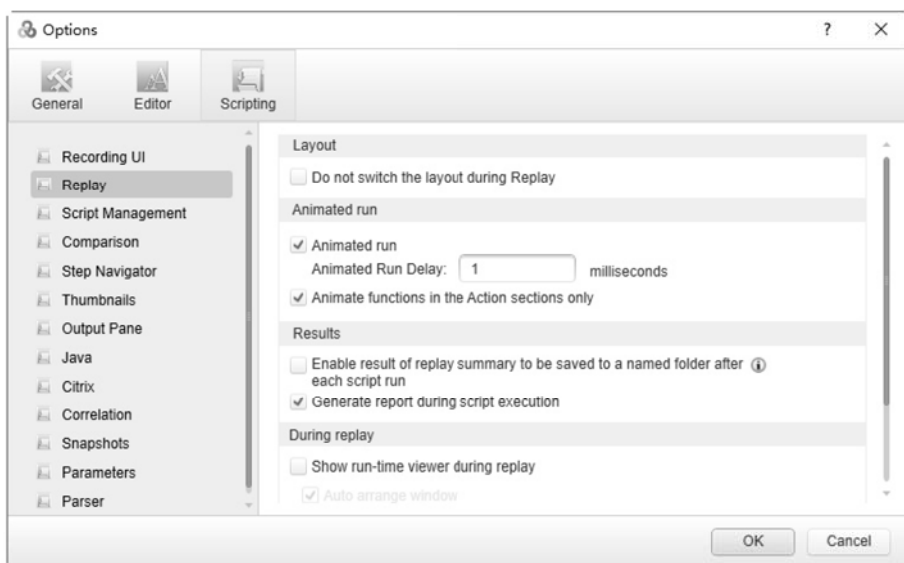


图 5-59 开始回放

5.5.3 查看回放信息

1. 查看概要信息

当脚本停止运行后，可以在向导中查看关于这次回放的概要信息。要查看上次回放概要，请单击验证回放。

上次回放概要列出检测到的所有错误，并显示录制和回放快照的缩略图。可以比较快照，找出录制的内容和回放的内容之间的差异。也可以通过复查事件的文本概要来查看 Vugen 操作。输出窗口中 Vugen 的“回放日志”选项卡用不同的颜色显示这些信息。

2. 查看回放的日志

单击说明窗口中的回放日志超链接。也可以单击工具栏中的显示/隐藏输出按钮，或者在菜单栏中选择视图→输出窗口，然后选择回放日志选项卡，如图 5-60 所示。

在回放日志中按 Ctrl+F 键打开 Find(查找)对话框，找到下列内容。

A: 启动和终止。脚本运行的开始和结束 - 虚拟用户脚本已启动、Vuser 已终止。

B: 迭代。迭代的开始和结束以及迭代编号(橙色字体部分)。

Vugen 用绿色显示成功的步骤，用红色显示错误。例如，如果在测试过程中连接中断，Vugen 将指出错误所在的行号并用红色显示整行文本。

双击回放日志中的某一行。Vugen 将转至脚本中的对应步骤，并在脚本视图中突出显

示此步骤。

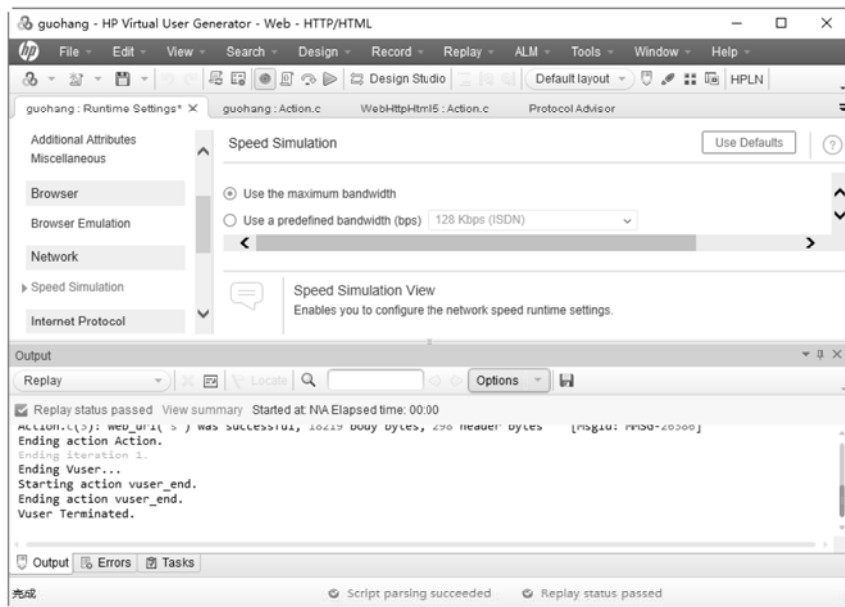


图 5-60 查看回放日志

5.5.4 确定测试已通过

回放录制的事件后，需要查看结果以确定是否全部成功通过。如果某个地方失败，则需要知道失败的时间以及原因。

查看测试结果的步骤如下。

- (1) 要返回到向导，单击任务窗格(Task)中的验证回放。
- (2) 在标题验证下的说明窗格中，单击可视测试结果超链接。也可以选择 **Replay→Replay Summary** 命令，打开如图 5-61 所示的窗口，单击其中的 **The Test Results** 链接，即可打开如图 5-62 所示的测试结果界面。

测试结果窗口首次打开时包含两个窗格：“树”窗格(左侧)和“概要”(Summary)窗格(右侧)。“树”窗格包含结果树，每次迭代都会进行编号。“概要”窗格包含关于测试的详细信息以及屏幕录制器视频(如果有的话)。在“概要”窗格中，中间的表格指出哪些迭代通过了测试，哪些未通过。如果进行测试的 **Vugen** 按照原来录制的操作成功执行 **HP Web Tours** 网站上的所有操作，则认为测试通过。下方的表格指出哪些事务和检查点通过了测试，哪些未通过。

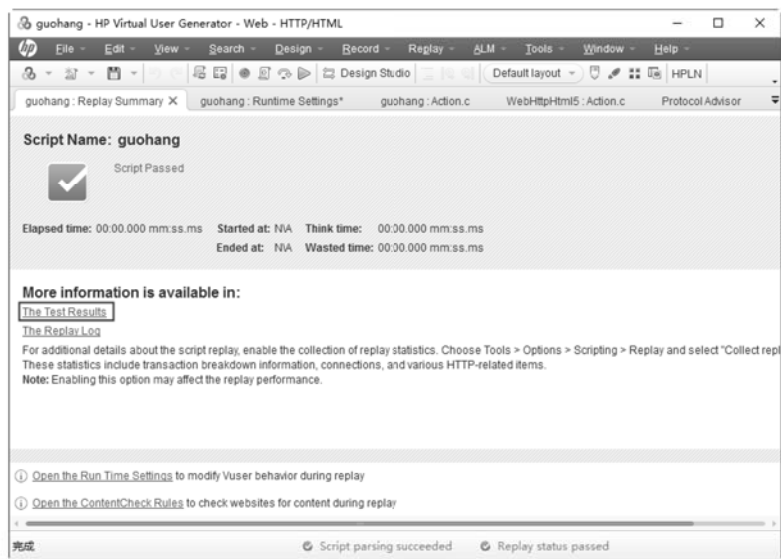


图 5-61 单击 The Test Results 链接

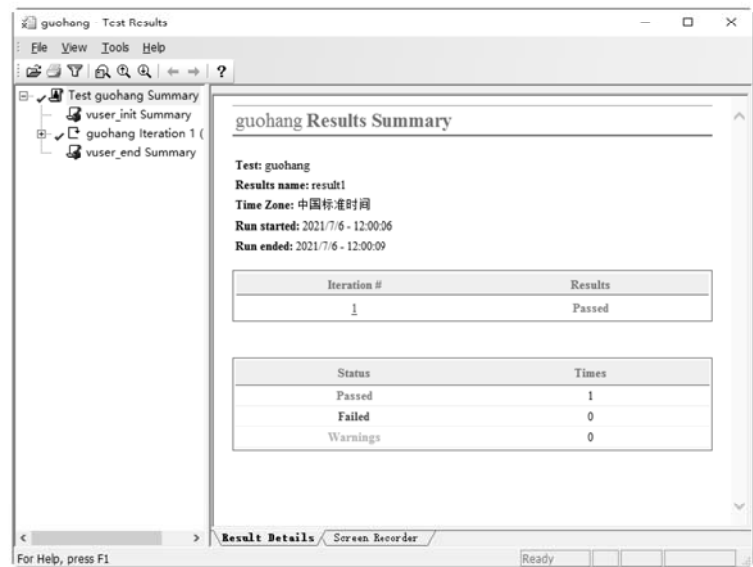


图 5-62 查看测试结果

5.5.5 搜索或筛选结果

如果测试结果表明有些地方失败，可以深入分析测试结果并找出失败的地方。在“树”窗格中，可以展开测试树并分别查看每一步的结果。“概要”窗格将显示迭代期间的回放快照。

1. 在树视图中展开迭代节点

展开节点 `basic_tutorial` 迭代 1, 然后单击加号 (+) 展开左窗格中的 `Action` 概要节点。展开的节点将显示这次迭代中执行的一系列步骤。

2. 显示结果快照

选择 `guohang Iteration 1:vuser_end Summary` 节点, 即可在概要窗格显示与该步骤相关的回放快照, 如图 5-63 所示。

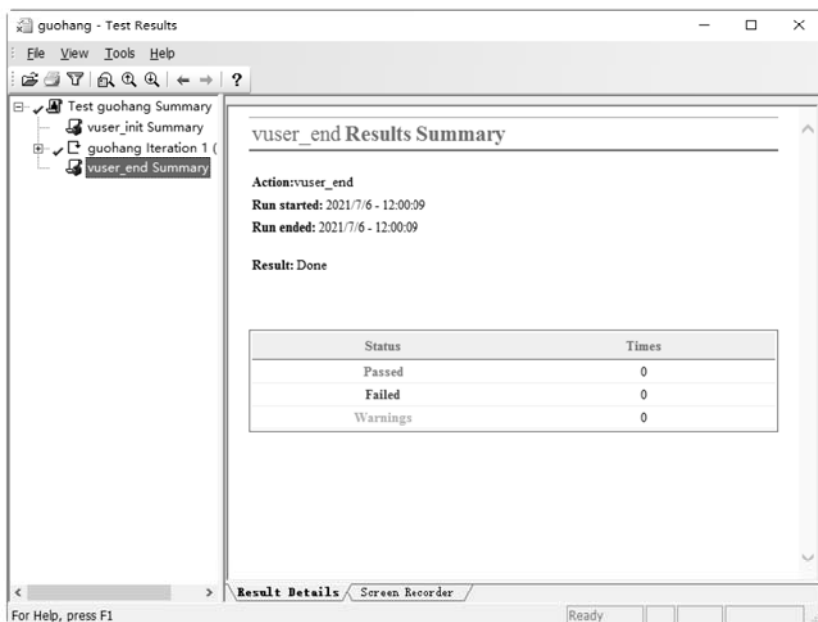


图 5-63 显示结果快照

3. 查看步骤概要

“概要”窗格显示步骤概要信息: 对象或步骤名、关于页面加载是否成功的详细信息、结果(通过(Passed)、失败(Failed)、完成(Done)或警告(Warnings))以及步骤执行时间。

4. 搜索结果状态

可以使用关键字“通过”或“失败”搜索测试结果。此操作非常有用, 例如当整个结果概要表明测试失败时, 可以确定失败的位置。要搜索测试结果, 选择 `Tools→Find` 命令, 或者单击查找按钮, 这时将打开查找对话框, 如图 5-64 所示。

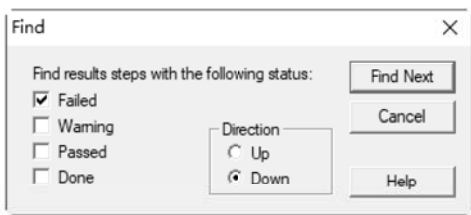


图 5-65 Find 对话框

选中 Done(通过)复选框，确保未选中其他选项，然后单击 Find Next(查找下一个)按钮。
“测试树”窗格突出显示第一个状态为通过的步骤。
注：如果找不到选定状态的步骤，则不突出显示任何步骤。

5. 筛选结果

可以筛选“测试树”窗格来显示特定的迭代或状态。例如，可以进行筛选以便仅显示失败状态。要筛选结果，请选择 View→Filters 命令，或者单击筛选器按钮，这时将打开筛选器对话框，如图 5-65 所示。在状态部分选择失败，不选择任何其他选项。在内容部分选择全部并单击 OK 按钮。因为没有失败的结果，所以左窗格为空。

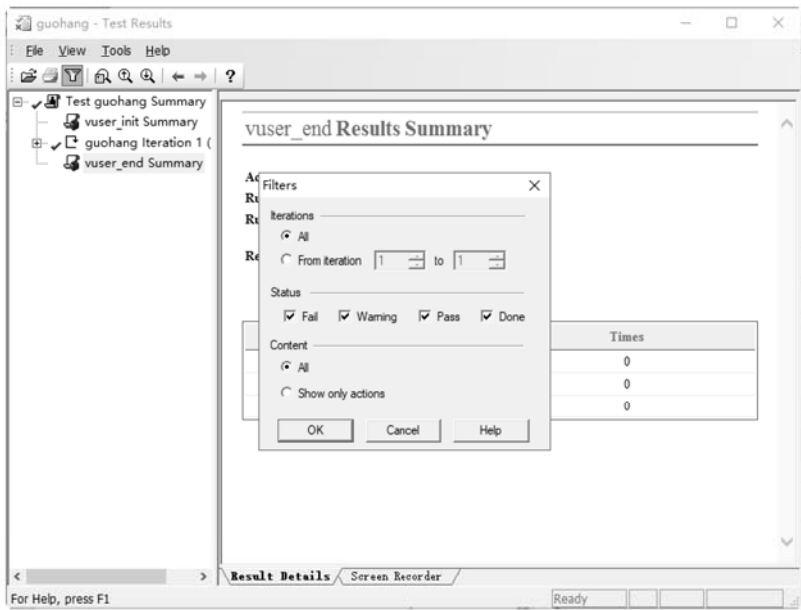


图 5-65 筛选器对话框

最后关闭 Test Results(测试结果)窗口，选择 File(文件)→Exit(退出)命令。

5.6 常见回放问题

创建脚本后，通过在 Vugen 中运行该脚本来对其进行验证。有时虽然操作录制成功，但简单的回放却会失败。许多应用程序都使用动态值，每次使用应用程序时这些值都会变化。例如，有些服务器会为每个新会话分配一个唯一的会话 ID。回放录制的会话时，应用程序创建的新会话 ID 与录制的会话 ID 不同。LoadRunner 通过关联解决了这种问题。关联将动态值(在本例中为会话 ID)保存到参数中。运行模拟场景时，Vugen 并不使用录制的值，而是使用服务器分配的新会话 ID。

5.6.1 设置 HP Web Tours 让其出现回放错误

要演示常见的回放错误，需要修改 HP Web Tours 应用程序中的设置。此设置告诉 HP Web Tours Web 服务器不允许出现重复的会话 ID。

1. 打开 HP Web Tours

选择开始→程序→HP LoadRunner→Samples→Web→HP Web Tours 应用程序。浏览器将打开 HP Web Tours 的主页。

2. 更改服务器选项

(1) 单击 HP Web Tours 主页上的 Administration(管理)链接，将打开 administration(管理)页面。

(2) 选中具有以下标题的复选框：Set LOGIN form's action tag to an error page.(将 LOGIN 表单的操作标记设置为错误页面)，向下滚动到页面底部并单击 Update (更新)。

(3) 向下滚动到页面底部并单击 Return to the Web Tours Homepage(返回到 Web Tours 主页)链接。此设置告诉服务器不允许出现重复的会话 ID。

5.6.2 如何使用唯一的服务器的值

在修改后的 HP Web Tours 配置中，服务器为 Vugen 分配一个唯一的会话 ID。现在如果回放脚本，将会失败。为解决此问题，请使用 Vugen 自动检测是否需要关联会话 ID。运行脚本后，Vugen 会提示扫描脚本，查看需要关联的地方。可以让 Vugen 插入将原始会话 ID

保存到参数中这一步。在每个回放会话中，Vugen 都会将新的唯一会话 ID 保存到参数中。在后面的步骤中，Vugen 使用保存的值而不是原来录制的值。

1. 使用动态值录制新脚本

录制新脚本，并将新脚本保存为 basic_tutorial_Cor。

2. 回放新脚本

在“任务”窗格中单击验证回放，然后单击说明窗格底部的 Start Replay(开始回放)按钮。Vugen 将运行新脚本，会看到输出窗口的回放日志选项卡中有几条显示为红色的错误消息。

3. 查看回放概要

在任务窗格中单击 Check Replay(验证回放)按钮以查看“上次回放概要”，如图 5-66 所示。

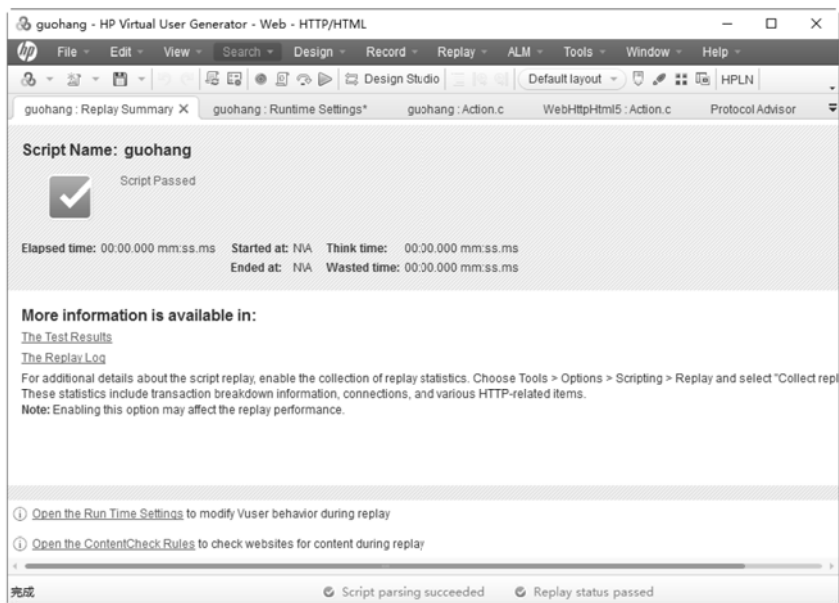


图 5-66 回放概要

4. 扫描脚本以查找需要关联的地方

单击说明窗格中标题“动态服务器值”下的链接显示并解析动态服务器值。Vugen 将扫描脚本，搜索录制值与回放值之间的不同。Vugen 将在 Output(输出)窗口的 Correlation

Results(关联结果)选项卡中显示一系列可能需要关联的差异,如图 5-67 所示。

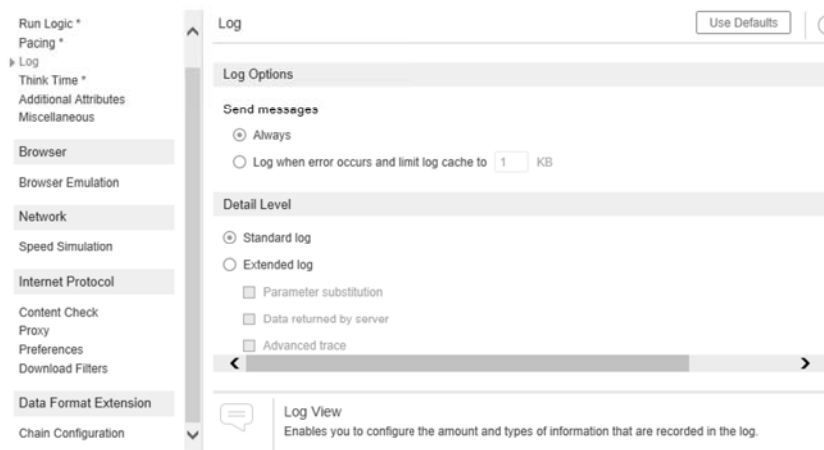


图 5-67 显示关联结果

5. 关联会话 ID

选择 Correlation Results(关联结果)选项卡中的第一个条目,单击 Correlate(关联)按钮。Vugen 将在脚本的顶部插入新步骤,将原始会话 ID 保存到参数中。在每个回放会话中,Vugen 都会将新的唯一会话 ID 保存到参数中。在后面的步骤中,Vugen 使用保存的值而不是原来录制的值。

6. 检查关联语句的语法

选择 View(视图)→Script View(脚本视图)命令,查看脚本中的关联语句。Vugen 添加到脚本中的语句如下:

```
web reg save param("WCSPParam Diff1,
"LB=userSession value=",
"RB=>"
"Ord=1",
"RelFrameId=1.2.1",
"Search=Body",
LAST);
```

该语句的意思是检查以下两个字符串之间数据的服务器响应。

- 左边界: userSession value=
- 右边界: >

该语句指示 Vugen 将首次出现的此数据保存到参数 WCSPParam_Diff1 中。