

3.7.6 关联函数的高级使用

上面详细介绍了关联的作用和关联函数的详细选项,那么在工作中除了要使用关联函数获得服务器返回以外,还能做什么呢?

例如论坛一个版面中有 20 个帖子,如何实现随机单击其中某一个帖子的操作呢?回想一下关联选项 Ord=All 的时候关联出来的结果是不是一个参数数组?既然是参数数组,怎么从参数数组中取出一个随机的值呢?

在不同的 LoadRunner 版本中处理这个问题使用不同解决方法,现在来分别了解一下具体的方案(这里关联后的参数名为 link,设置 Ord 为 All)。

LoadRunner 9 系列

在 LoadRunner 9 中做这个操作非常简洁,因为有了参数数组函数,所以只需要这样写就可以了:

```
char * siteval;  
siteval =lr_paramarr_random(link)
```

即直接从参数数组 link 中取一个随机的值。

问题: 如果需要随机选择前 10 个帖子怎么办?

Lr_paramarr_random()函数的随机范围其实是根据 lr_paramarr_len()决定的。比如数组长度是 20,随机值介于 1~20 之间,现在手动将这个参数数组设置得小一些,问题就解决了:

```
char * siteval;  
lr_save_string("10","link_count");  
siteval =lr_paramarr_random(link);
```

思考: 如果需要随机获取关联结果中的第 5 至第 10 个对象,该如何处理呢?

在这种情况下,需要引入随机数,生成随机值为 5~10 的正整数,再调用 lr_paramarr_idx() 函数进行处理即可。

如果使用的是 LoadRunner9 以前的版本,没有这个参数数组函数怎么办?

LoadRunner 8 以前的版本

假设存在关联后的参数数组为{link},数组记录总个数为 20,需要取得其中的一个随机关联值可以这样写:

```
char linkname[100],num[100];  
int randnum;  
//关联和请求操作省略  
randnum=rand()%atoi(lr_eval_string("{link_count}))+1;//获得关联参数数目内的  
随机数字  
strcpy(linkname,"{link_");  
//lr_error_message("%s",linkname);  
itoa(randnum,num,10);  
strcat(linkname,num);  
strcat(linkname,"}");  
//lr_error_message("%s",linkname);  
lr_save_string(lr_eval_string(linkname),"temp");
```

这段代码看起来是比较头疼的，因为涉及了很多新的函数。

`atoi()`类型强制转换函数的作用是将字符串型的内容转化为整数型。

`atoi(字符串);`

由于使用求余操作是对一个数字进行操作，所以需要使用 `atoi` 将参数转化为正整数。

使用 `Rand()%atoi(lr_eval_string("{link_count}"))`可以得到 0~19 的随机正整数。

`strcpy()`字符复制函数就是将一个字符串复制到一个变量中去。

`strcpy(变量名,需要复制的字符串内容);`

所以 `strcpy(linkname,"{link_}")`的作用是将"`{link_}`"这个内容保存到变量 `linkname` 中。

`itoa()`也是一个强制类型转换函数，和 `atoi()`相反，它是将整数型内容转化为字符串型。

`itoa(数字型,字符串,转化格式);`

这里转化格式使用的是十进制。为了拼接参数数组，需要生成"`{link_2}`"这样的参数，通过前面的随机函数已经生成 1~20 的随机正整数，现在需要把这个数字拼接上去，这里使用 `strcat()`来实现，由于 `strcat()`必须使用字符串，所以需要将随机整数 `randnum` 转化成字符串型的 `num`。

`strcat()`是一个字符添加函数，它将一个字符串附加在一个变量后。

`strcat(变量名,字符串);`

接着将上面生成的字符串继续拼接到 `linkname` 变量中。

这样就拼接出了 `linkname="{link_2}"`，在讲述参数化的时候提到过参数和变量的调用，当变量这样写的时候就可以直接读取到参数名的值。所以使用 `lr_eval_string()`可以将这个变量对应的参数值取出，再通过 `lr_save_string()`将值存放到另一个参数 `temp` 中去，最后参数 `temp` 就存放了 `{link_2}` 参数所对应的值了。

如果需要得到 1~10 的随机记录呢？这个时候只需要在随机数生成的时候做点手脚就行了。

`randnum=rand()%atoi(lr_eval_string("{link_count}"))+1;//获得关联参数数目内的随机数字`

把这段代码修改为：

`randnum=rand()%10+1;//得到 1~10 的随机数字`

在这里最好先做一个判断，避免出现帖子少于 10 个的问题。

如果需要得到 5~10 的随机记录呢？这个时候还是随机数生成的操作，如何生成一个范围内的随机数呢？通过公式 `rand()%(max-min+1)+min` 能生成从最小值到最大值之间的随机数，所以只需要写为以下形式即可：

`Rndnum=rand()%(10-5+1)+5;`

LoadRunner 8 系列

到了 LoadRunner 8.x 系列，VuGen 提供了一个新的函数来帮助我们快捷地处理类似的类型转化操作。这个函数就是 `sprintf()`，`sprintf()`和 C 语言中的 `printf()`函数十分相似，使用它可以生成带格式的字符串，从而帮助我们快捷地完成一个特殊格式拼接过程。

sprintf(变量名,格式,值)

例如可以这样写:

```
char temp[100];  
sprintf(temp, "welcome %dtesting", 51);  
lr_error_message(temp);
```

可以看到结果是"welcome 51testing", 通过这个函数将数字 51 拼接到了这个字符串中。还是前面的脚本, 看看在 LoadRunner 8.x 中怎么写, 现在可以将脚本改为以下形式:

```
char linkname[100];  
int randnum;  
randnum=rand()%atoi(lr_eval_string("{link_count}))+1;//获得关联参数数目内的  
随机数字
```

```
    sprintf(linkname, "{link_%d}", randnum);  
    lr_save_string(lr_eval_string(linkname), "temp");
```

这样直接就把随机的 randnum 变量放在了 linkname 变量中, 并且生成 linkname="{link_2}" 这样的变量。后面的按照 LoadRunner 7 系列的做法就行了, 是不是方便了 很多呢?

通过上面的方法就能处理常见的一些在关联后需要进行随机处理的情况。例如: 现在需要这样的脚本, 访问论坛首页, 登录后检查所有在线用户, 然后随机给所有的在线普通会员发一条广告短信息, 那么这个脚本怎么去做呢?

分析一下业务: 作为一个用户, 首先要登录系统, 然后查看到所有的在线用户, 依次给在线的用户发送短信息。录制一个用户执行以上操作的脚本。如何获得在线的用户信息? 在首页的最下面可以看到在线用户的信息, 通过 HTML 代码可以发现普通用户和管理员的区别在于用户名前的图片名。

```
<li>  
<a href="userinfo-8409.aspx" target="_blank" title="时间: 2009/3/19 9:52:16  
操作: ">admin00011</a>  
</li>
```

普通用户使用的图片是 member.gif, 而管理员使用的是 admin.gif, 所以可以通过这个信息来做一个关联操作, 得到所有用户信息前是 member.gif 的用户 id。

```
web_reg_save_param("member",  
    "LB=member.gif\" /><a href=\"userinfo-\",  
    "RB=.aspx",  
    "Ord=ALL",  
    "Search=NoResource",  
    LAST);  
web_url("fristpage", "URL=http://192.168.0.200", LAST);
```

按照代码填写关联的边界, 运行后会发现关联失败, 原因来自于 member.gif" />和<a href=中其实还有一些符号, 遇到这种无法直接写出左右边界的情况, 应该先将边界设置得更大一点, 修改关联函数。

现在再运行一下这个脚本，可以得到对应的关联内容。

通过这个关联结果，进一步尝试补全整个左边界。

通过这个代码,就可以将所有是在线的普通用户 id 通过关联函数保存到 member 这个参数数组中。

```
web_submit_data("usercppostpm.aspx",  
  
                "Action=http://192.168.0.200/usercppostpm.aspx?msg  
toid=1",  
  
                "Method=POST",  
                "TargetFrame=",  
                "RecContentType=text/html",  
  
                "Referer=http://192.168.0.200/usercppostpm.aspx?ms  
gtoid=1",  
  
                "Snapshot=t14.inf",  
                "Mode=HTML",  
                ITEMDATA,  
                "Name=msgto", "Value=admin", ENDITEM,  
                "Name=subject", "Value=广告测试", ENDITEM,  
                "Name=message", "Value=广告测试", ENDITEM,  
                "Name=sendmsg", "Value=立即发送", ENDITEM,  
                LAST);
```

1. 用前面的 `strtok()` 进行关联拆分。
2. 另一种做法是，得到了用户 `id` 后再通过用户 `id` 去获得用户名。这里使用第 2 种做法，所以可以得到下面的最终代码。

51Testing 软件测试网作品系列

```
web_reg_save_param("username",
    "LB=<td class=\"navname\">",
    "RB=</td>",
    "Ord=1",
    "Search=NoResource",
    LAST);

web_url("userinfo","URL=http://192.168.0.200/userinfo
-{userid}.aspx",LAST);

web_submit_data("usercp postpm.aspx",

    "Action=http://192.168.0.200/usercp postpm.aspx?msgtoi
d={userid}",

    "Method=POST",
    "TargetFrame=",
    "RecContentType=text/html",

    "Referer=http://192.168.0.200/usercp postpm.aspx?ms
gtoid=1038",

    "Snapshot=t14.inf",
    "Mode=HTML",
    ITEMDATA,
    "Name=msgto", "Value={username}", ENDITEM,
    "Name=subject", "Value=广告测试", ENDITEM,
    "Name=message", "Value=广告测试", ENDITEM,
    "Name=sendmsg", "Value=立即发送", ENDITEM,
    LAST);
}

Return 0;
}
```

补充:

表 3.14 关联中常用的转义内容

转 义 内 容	说 明
\b	Backspace 键
\f	换页
\n	换行
\r	回车
\t	水平制表符
\v	垂直制表符
\'	单引号标记
\"	双引号标记
\\	反斜杠
\?	文本问号



注意

关联的使用要点:

1. 什么内容需要关联

当脚本中的数据每次回放都发生变化时,并且这个动态数据在后面的请求中需要发送给服务器,那么这个内容就需要通过关联来询问服务器,获得该数据的变化结果。当确认关联的内容后,要确认所需要的数据是在哪个请求中返回的,并且定位该数据的位置和特征,以及左右边界。

2. 关联的边界如何设置

左右边界是定位动态数据的关键,但是想要精确定位数据的左右边界并不是一件容易的事,可以通过先松后紧的方式来达到这个目标。首先在做关联的时候设置一个比较明确固定的值,确保能够将所需要的内容整体保存为参数,其次再根据返回的内容进一步细化边界值,一步步达到准确关联结果的目的。