[**loadrunner关联及web\_reg\_save\_param方法浅析**](http://www.cnblogs.com/Automation_software/p/3482094.html)

一、什么是关联

关联（correlation）：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，已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，这种动态获得服务器响应内容的方法被称作关联。也是把脚本中某些写死的数据，转变成动态的数据。  
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，那么这个内容需要通过关联来询问服务器，获得该数据的变化结果。例如：  
1.登录字符串。带有会话 ID 或时间戳等动态数据的登录字符串。  
2.日期/时间戳。使用日期或时间戳或者其他用户凭据的任意字符串。  
3.常见前缀。后跟字符串的常见前缀，如 SessionID 或 CustomerID

二、web\_reg\_save\_param函数说明

语法：

int web\_reg\_save\_param(const char \*ParamName, <list of Attributes>, LAST);

参数说明:

· ParamName: 存放得到的动态内容的参数名称

· list of Attributes: 其它属性，包括：Notfound, LB, RB, RelFrameID, Search, ORD, SaveOffset, Convert, SaveLen。属性值不分大小写

o Notfound: 当在返回信息中找不到要找的内容时应该怎么处理

o Notfound=error: 当在返回信息中找不到要找的内容时，发出一个错误讯息。这是缺省值。

o Notfound=warning: 当在返回信息中找不到要找的内容时，只发出警告，脚本也会继续执行下去不会中断。

o LB( Left Boundary ) : 返回信息的左边界字串。该属性必须有，并且区分大小写。

o RB( Right Boundary ): 返回信息的右边界字串。该属性必须有，并且区分大小写。

o RelFrameID: 相对于URL而言，欲查找的网页的Frame。此属性质可以是All或是数字，该属性可有可无。

o Search : 返回信息的查找范围。可以是Headers，Body，Noresource，All(缺省)。该属性质可有可无。

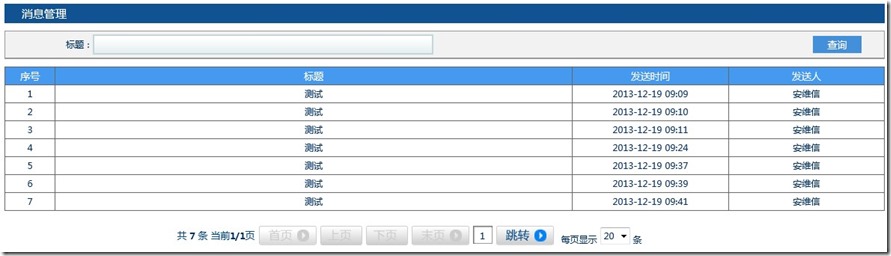
o ORD : 说明第几次出现的左边界子串的匹配项才是需要的内容。该属性可有可无，缺省值是1。如为All，则将所有找到的内容储存起来。

o SaveOffset : 当找到匹配项后，从第几个字元开始存储到参数中。该属性不能为负数，缺省值为0。

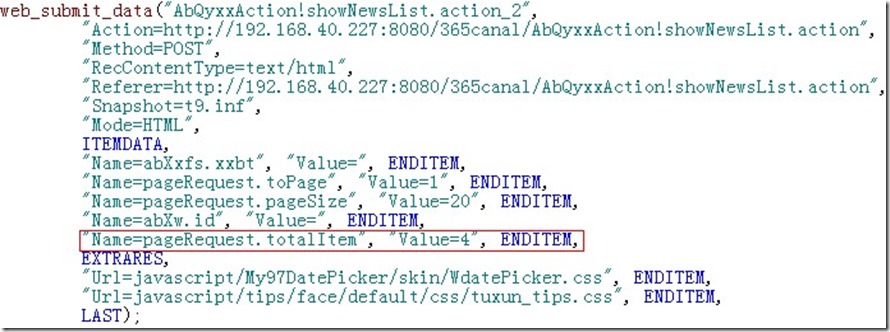
o SaveLen ：当找到匹配项后，偏移量之后的几个字元存储到参数中。缺省值是-1，表示一直到结尾的整个字串都存入参数。

三、实例解析

例1：

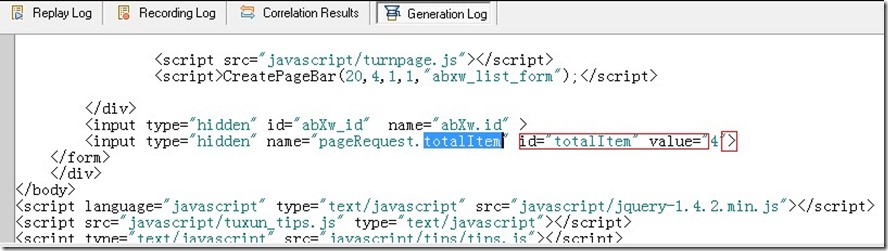
[](http://images.cnitblog.com/blog/219750/201312/19132434-bbe29af05b4c442a82c47792d01f7f6c.jpg)

如图所示，不输入查询条件，直接点击【查询】按钮。生成的数据提交脚本如下：

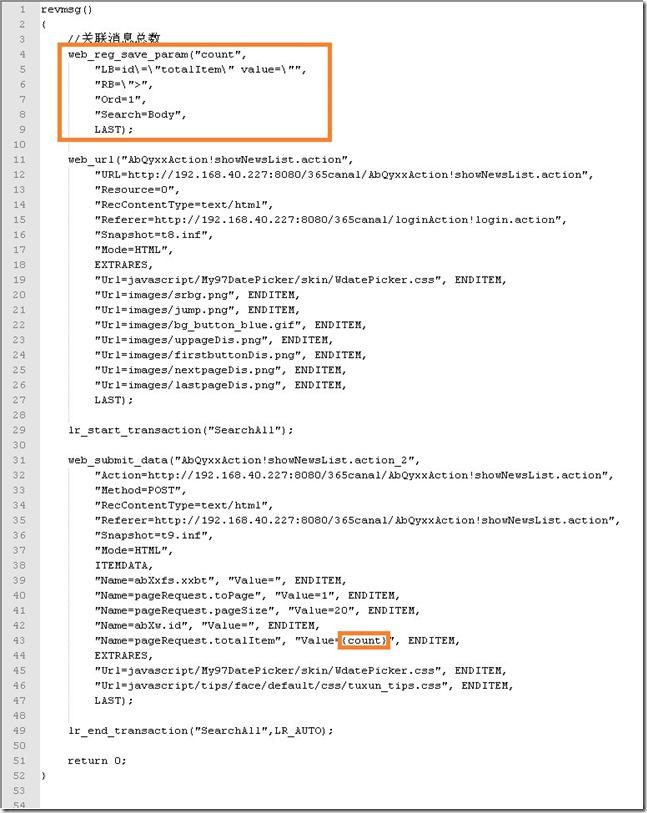
[](http://images.cnitblog.com/blog/219750/201312/19132435-cab207846d0a42b4bf73ba5acb0710ee.jpg)

该value值为请求的数据总数，该值是动态变化的。为了正确的发送请求并得到正确的结果，需要对该值进行关联。

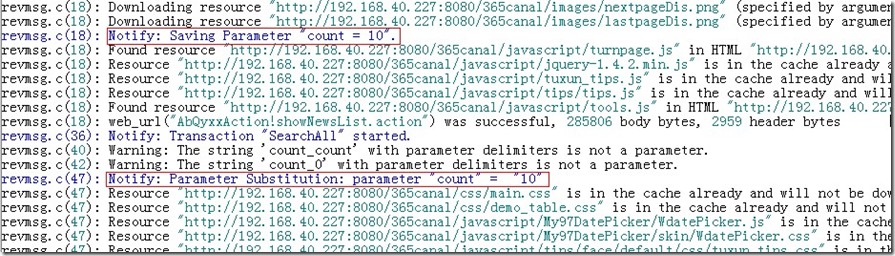
在Generation Log中搜索“totalItem”，如图所示，从而可以确定该value值的左右边界。

[](http://images.cnitblog.com/blog/219750/201312/19132435-39eac565e0b74e6a94e63962574199c6.jpg)

在脚本的请求前插入web\_reg\_save\_param方法，并在提交数据请求的时候使用｛count｝来替代录制时实际的值。如下图：

[](http://images.cnitblog.com/blog/219750/201312/19132436-24020708a0074ab5b2d8c19c4de2c6c0.jpg)

打开扩展日志，运行脚本，可以看到正确的关联出了结果。

[](http://images.cnitblog.com/blog/219750/201312/19132624-b2b960d4b1844156bec836e2009de2e9.jpg)

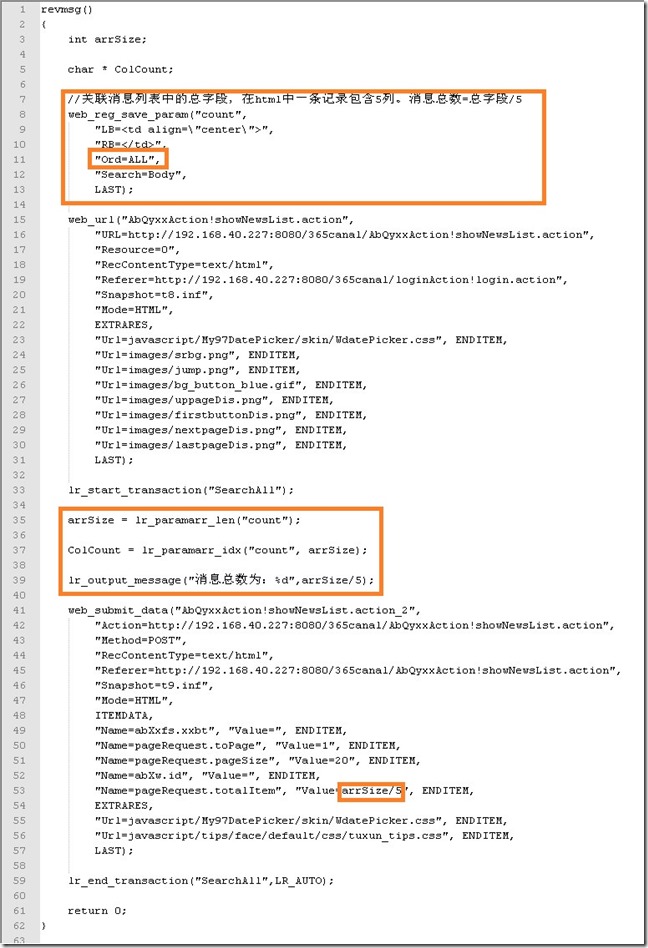
例2：

上面实例是使用web\_reg\_save\_param的ORD=1，只是取第1次出现的左边界子串的匹配项为需要的内容，下面例子使用web\_reg\_save\_param的ORD=ALL，将匹配到的内容保存到数组中。

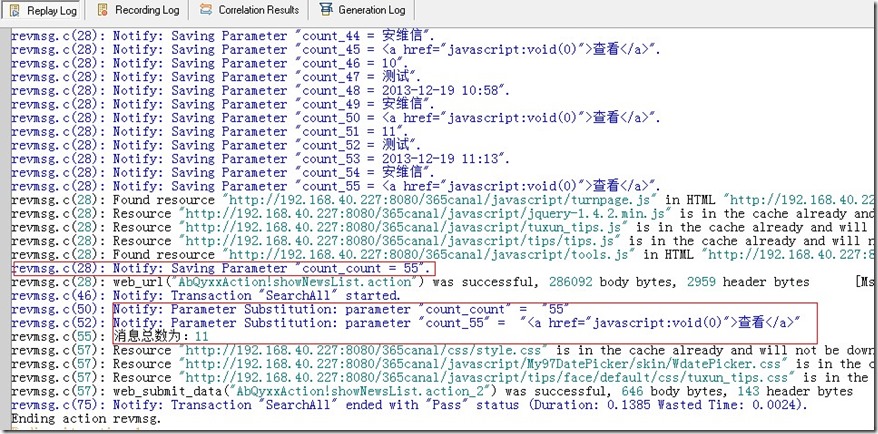
根据消息列表中的记录数来进行关联（只是为了演示，分页不考虑），跟踪Generation Log可以看到，一条记录的内容会包含5列内容。如下图：

[](http://images.cnitblog.com/blog/219750/201312/19132631-6752938f496e4b62b49d41377d329fb4.jpg)

从上图中可以确定其左右边界，在脚本的请求前插入web\_reg\_save\_param方法，并在提交数据请求的时候使用arrSize/5来替代录制时实际的值。如下图：

[](http://images.cnitblog.com/blog/219750/201312/19132702-b31c335c1ad645f7888667ea72a8c246.jpg)

其中lr\_paramarr\_len方法用于取出数字长度，lr\_paramarr\_idx方法用于取数组指定下标的值。打开扩展日志，运行脚本，可以看到正确的关联出了结果。

[](http://images.cnitblog.com/blog/219750/201312/19135905-1d099ae2fa3a47e8b0b90d14b5ba3406.jpg)