3.8 检查点

首先谈一下 VuGen 确定脚本运行成功的判断条件。在录制编写脚本后，通常就会进行回放，如果回放通过没有出现错误，就认为脚本是正确的。究竟 VuGen 怎么区分脚本是否回放正确呢？细心的朋友会发现，基本上所有脚本回放错误都是因为 404 错误，也就是页面无法找到，而只要页面返回了，VuGen 都不会提示任何错误。例如前面登录 Web Tours 网站的脚本，由于没有做关联操作而导致登录失败，但是脚本还是执行成功了，在 Test Results 中显示为 PASS 状态。

VuGen 判断脚本是否执行成功是根据服务器返回的状态来确定的，如果服务器返回的 HTTP 状态为 200 OK，那么 VuGen 就认为脚本正确地运行了，并且是运行通过的。在绝大多数系统出错时会返回错误页面吗？不会，一般系统都会返回一个消息提示框，来提升用户感受。例如：“网站忙，请稍候”。其实这个时候网站已经无法正确响应用户请求了，但是 VuGen 脚本无法识别，会错误地认为网站还能正确访问，导致分析错误。所以这时需要一种检查点函数帮助验证请求发送出去后，服务器的返回是不是期望的内容，如果不是，那么就说明服务器无法提供正常的服务了。

检查点函数和关联函数比较相似，同样也是一个注册型函数 web_reg_find()。该函数能够对服务器返回的内容进行检查。

例如：Web Tours 的注册操作，在注册成功后，服务器会返回以下内容:

```
Thank you, test1, for registering and welcome to the Web Tours family. We hope
we can meet all your current and future travel needs. If you have any questions,
feel free to ask our support staff. Click below when you're ready to plan your
dream trip...
```

这里带下画线的是注册用户名，只要服务器返回以上的内容那么就说明注册成功了，否则就是注册失败，脚本回放失败。先录制一个注册用户的脚本，如下所示：

```
Action()
{
                        web_url("WebTours",
                         "URL=http://127.0.0.1:1080/WebTours/",
                         "TargetFrame=",
                         "Resource=0",
                         "RecContentType=text/html",
                         "Referer=",
                         "Snapshot=t1.inf",
                         "Mode=HTML",
                         LAST);

                        web_url("sign up now",
                        "URL=http://127.0.0.1:1080/WebTours/login.pl?
username=&password=&getInfo=true",
                         "TargetFrame=body",
                         "Resource=0",
```

```
                            "RecContentType=text/html",

                           "Referer=http://127.0.0.1:1080/WebTours/home.html"
,
                            "Snapshot=t2.inf",
                            "Mode=HTML",
                            LAST);

                           web_submit_data("login.pl",
                            "Action=http://127.0.0.1:1080/WebTours/login.pl",
                            "Method=POST",
                            "TargetFrame=info",
                            "RecContentType=text/html",
                           "Referer=http://127.0.0.1:1080/
WebTours/login.pl?username=&password=&getInfo=true",
                            "Snapshot=t3.inf",
                            "Mode=HTML",
                            ITEMDATA,
                            "Name=username", "Value=test2", ENDITEM,
                            "Name=password", "Value=test2", ENDITEM,
                            "Name=passwordConfirm", "Value=test2", ENDITEM,
                            "Name=firstName", "Value=", ENDITEM,
                            "Name=lastName", "Value=", ENDITEM,
                            "Name=address1", "Value=", ENDITEM,
                            "Name=address2", "Value=", ENDITEM,
                            "Name=register.x", "Value=34", ENDITEM,
                            "Name=register.y", "Value=9", ENDITEM,
                            LAST);
    return 0;
    }
```

回放这个脚本，查看 Test Results 中显示的结果是运行通过，但是如果认真看一下最后一步的截图，就会发现回放的脚本根本就没有执行成功，因为 test2 用户已经存在了，回放不会注册任何新的用户，但是 VuGen 不会判断注册失败的错误，这里必须通过检查点函数来解决。

现在在 web_submit_data()函数前添加检查点函数（注册型函数一定要写在请求前）。打开 Insert 菜单下的 New step 选项，找到 web_reg_find()函数并进行添加，如图 3.155 所示。

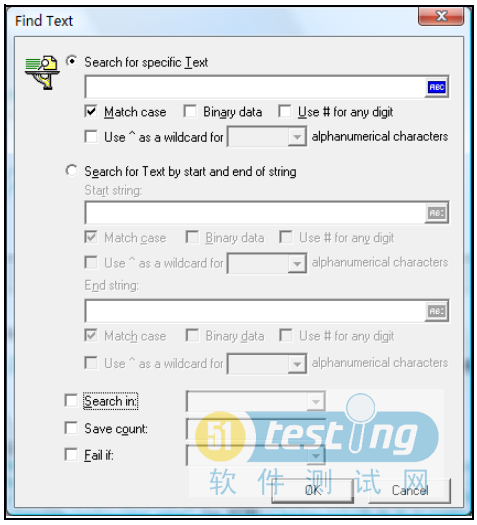


图 3.155　web_reg_find()函数设置界面

注 意

我们在进行性能测试而不是功能测试，虽然检查点也可以很好地实现功能测试，但在大多数时候并不需要太在意功能点是否全部正确，只要确保基本业务操作正确即可（如果希望测试系统在高负载情况下的功能是否正常可以在运行场景中调用 QTP 实现，具体内容参考第 4.4 节）。

3.8.1　文本检查点

web_reg_find()文本检查点函数提供了一种对服务器返回内容进行查询的功能，和关联的不同之处在于检查点函数只能返回检索到内容的次数。

- Search for specific Text

需要查询的标准文本，该功能和 Word 中的查找功能十分相似。在服务器返回的内容中查找特定的字符串，内容支持参数化，并且支持和关联相同的通配符和识别方式。

在这里输入需要查找的字符串即可，检查点函数将会找出服务器返回中是否存在需要查找的内容。

- Search for Text by start and end of string

文本检查点函数也同样提供了根据左右边界进行查找的功能，选项参考关联函数。

- Search in

设置在服务器返回的哪部分数据中进行查询。这里提供了 All、Headers、Body 三个选项，和关联函数相同，Headers 是指 HTTP 返回的包头部分；而 Body 是指返回 HTTP 的正文部分，一般来说需要检查的内容都存放在 Body 中。

- Save count

这是文本检查点很特别的功能，它将记录查找内容的出现次数并且存放到一个参数中，这里可以填写一个参数名称来存放计数结果。

- Fail if

设置在什么情况下文本检查点函数错误，提供了两个选项：Found、NotFound。如果选择 Found 也就是说如果在服务器返回中存在需要检查的对象，那么文本检查点函数出错；选择 NotFound 则相反，如果没有找到对应的内容，那么文本检查点函数出错。

检查点函数的错误会导致整个脚本运行结果的失败，通过这个功能可以方便地定位脚本

运行中的逻辑错误。

接着来完成注册用户的脚本，实现脚本每次运行必须完成用户的注册，否则脚本运行失败。在脚本中添加检查点函数。在 Search for specific Text 中输入以下内容：for registering and welcome to the Web Tours family，然后设置 Search in 为 Body、Save count 为 regcheck、Fail if 为 Not Found，如图 3.156 所示。

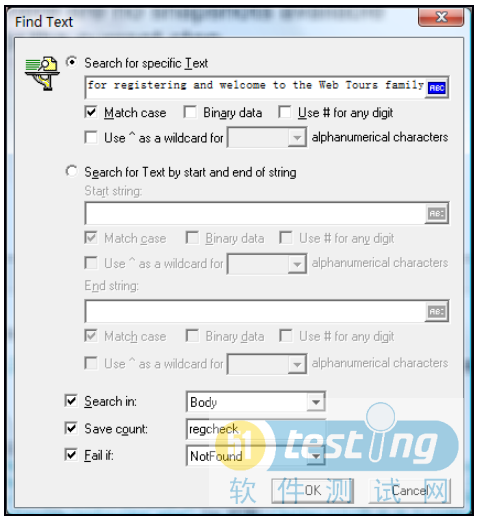


图 3.156　设置 web_reg_find 函数

确定后，得到检查点函数：

```
web_reg_find("Fail=NotFound",
 "Search=Body",
 "SaveCount=regcheck",
 "Text=for registering and welcome to the Web Tours family",
 LAST);
```

现在再回放一下脚本看看效果，会得到以下错误信息：

```
Action.c(36): Error -26366: "Text=for registering and welcome to the Web Tours family" not found for web_reg_find     [MsgId: MERR-26366]
Action.c(36): web_submit_data("login.pl") highest severity level was "ERROR", 2630 body bytes, 226 header bytes  [MsgId: MMSG-26388]
```

在服务器的返回中没有发现我们想要搜索的东西，所以脚本回放错误，整个 Test Results 是以 FAIL 结束的。接着修改一下注册的用户信息，看看检查点函数是否能查询得到相关信息。修改用户名后，再次运行脚本。

```
Action.c(36): Registered web_reg_find successful for "Text=for registering and welcome to the Web Tours family" (count=1)     [MsgId: MMSG-26364]
Action.c(36): web_submit_data("login.pl") was successful, 1020 body bytes, 343 header bytes           [MsgId: MMSG-26386]
```

可以看到 web_reg_find()函数成功地查询到了需要的内容，并且提示 count=1，说明找到了 1 次该内容，脚本成功运行，Test Results 状态为 PASS。

如果打开了参数存取值日志选项，可以看到 regcheck 参数的值，如果没有查询到那么

regcheck 的值为 0，反之 regcheck 存放 count 对应的记录条数。检查点函数有效地对脚本运行进行了监控，确保了脚本操作的正确性。另外一方面也可以通过它来进行一些逻辑分支的判断和页面的内容检查工作。

3.8.2 自动检查点

有些时候我们需要确保每个请求中都包含一个检查信息，然而手动添加检查点函数非常麻烦，这个时候可以使用自动检查点规则，通过建立一个新的规则，可以在回放时对所有的返回内容进行检查工作。

打开 Run-time Settings/ContentCheck，如图 3.157 所示。

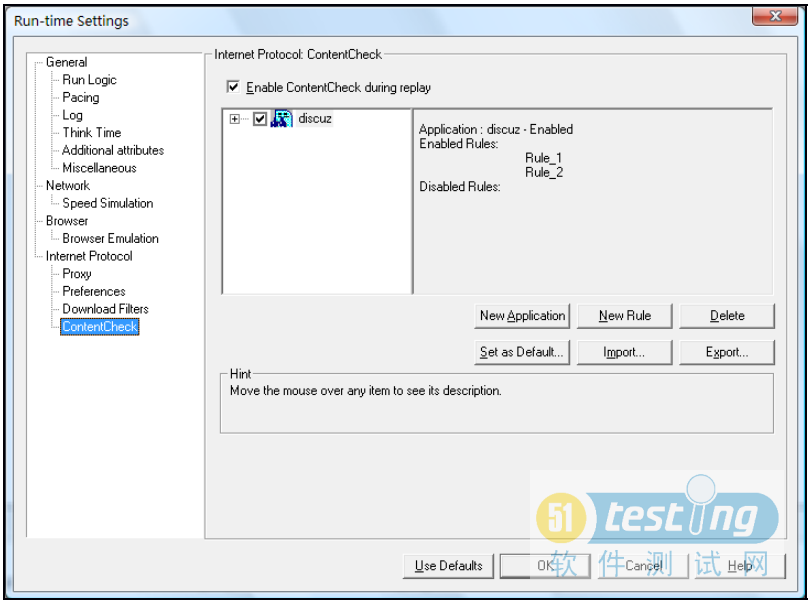


图 3.157 ContentCheck 设置窗口

在这里单击 New Application 按钮来添加一个应用，然后在这个应用下添加对应的规则，单击 New Rule 按钮。这里可以在 Search for Text 中填写需要检查的内容，也可以在下面的 Search by prefix and suffix 中填写需要检查的边界，大小写需要区分的话将 Match case 选中，最后确定该规则在什么情况下失败。

如果希望脚本检查每个页面是否包含 logo.gif 这个图片，那么可以在这里添加一个图片检查规则。在 Search for Text 中填写 images/logo.gif，设置 Fail if 为 Not Found，如图 3.158 所示。

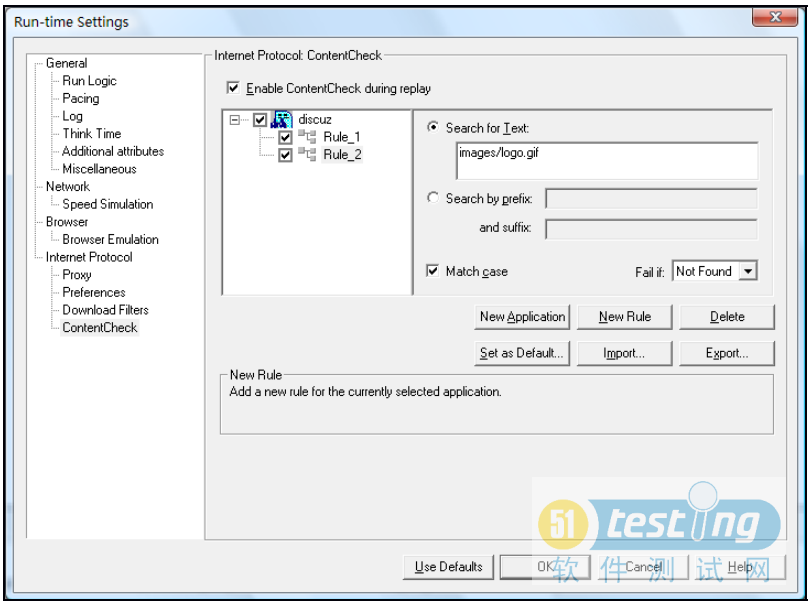


图 3.158　设置全局检查点规则

运行脚本时，如果有页面返回的代码中没有包含 images/logo.gif 时，就能看到以下错误：

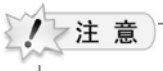
```
Action.c(3): Error -26370: ContentCheck Rule "Rule_2" in Application "discuz"
triggered. Text "images/logo.gif" not matched      [MsgId: MERR-26370]
Action.c(3): web_url("mainpage") highest severity level was "ERROR", 114808
body bytes, 4592 header bytes  [MsgId: MMSG-26388]
```

如果返回的代码中包含 images/logo.gif 则不会有任何提示出现。通过自动检查点，可以轻松地对页面中固定出现的内容进行校验工作。也可以通过 web_global_verification()函数实现全局检查点功能。

3.8.3　图片检查点

通过 web_image_check()函数可以检查页面上的图片。例如：

```
web_image_check("函数标题","ALT=图片说明", LAST);
```



图片检查点函数必须在 Run-time Settings 中打开 Preferences 下的 Enable Image and text check 才有效。