

QTP 技术集锦

文档

Version 1.0

自动化测试	Version:	1.3
文档		2006-5-17

Revision History

Date	Version	Description	Author
2007-1-22		first draft	Wiston Li
2007-3-22		Document migration	Wiston Li

自动化测试	Version:	1.3
文档		2006-5-17

2. 描述性编程 2	15
3. 函数	17
3.1. QTP 中几个截取字符串的函数	18
4. "+" "&" 的区别	20
5. QTP 场景恢复	21
6. QTP 中测试环境的初始化	27
7. QTP 将数据库中的数据导入到 excel	29
8. 在 QTP 中将数据库的数据导入到 EXCEL	29
9. QTP 将数据测试驱动	31
10. QTP 中设置共享对象库	32
11. QTP ActiveX 报错;	35
12. 数据驱动在测试中的应用	35
13. 在 Excel 中增加超链接的方式	39
14. 登陆脚本设计	40
15. QTP 中 Wait 与同步点的区别	41
16. QTP tips	41
17. Action 调用	45
18. Action 参数调用	47
19. QTP 中根据日期计算年龄的解决办法	48
20. 测试报告知多少	48
21. 调用共享文件	49
22. 识别 grid 中的 button	52
23. How to parameterize your link	52

自动化测试	Version:	1.3
文档		2006-5-17

24. QTP 中的步骤生成器	54
25. QTP 感想	56
26. QTP 参数化实例	60
27. 对象库的管理与合并	62
28. QTP 中虚拟对象设置	66
29. QTP 删除新建文本文件没有权限的问题	68
30. 关于 Filter 如何使用的疑问	69
31. Call Windows API	69
32. Global 和 local 区别	72
33. QTP itself API	73
34. QTP 需探讨的问题	74
在我的文章中也提到过了,描述性编程能让 QTP 脱离对象库来执行脚本,在某些特殊应用的时候,可能就更方便点.比如:要针对某个对象的参数化等等.而且描述性脚本因为脱离了对象库,所以增加了脚本的可移植性.同时脚本也更具有可读性.当然也给手工写脚本提供了一个平台.	74
35. 正则表达式的应用	75
36. 数据驱动测试: EXCEL	81
37. Action parameters	92
38. 数据驱动测试: 文件	95
39. 相对路径 action path	101
40. 两种方法参数化 link	101
41. WScript	102
42. QTP Plus 的应用	109
43. 一个万能的键盘模拟函数	110

自动化测试	Version:	1.3
文档		2006-5-17

44.	XML 参数化	115
45.	Oracle QTP	115
46.	对象库的管理与合并	116
47.	change sysdate	120
48.	Import Excel	120
49.	Call Dll	121
50.	Call Function	123
51.	Global and local 区别	124
52.	Shell	124
53.	对文件系统操作	125
54.	对象同步	125
55.	正则表达式应用模板	125
56.	Wscript.shell 的应用	126
57.	Create log file	126
58.	对象同步	127
59.	异常控制	127
60.	Check if the object.exist	128
61.	QTP 场景恢复	129
62.	QTP 步骤生成器	135
63.	IFRAME issues	138
64.	Iframe issues 续	142
65.	QTP 解析 PDF	144

自动化测试	Version:	1.3
文档		2006-5-17

66.	QTP 常见问题解答一	144
67.	QTP 常见问题解答二	146
68.	QTP 常见问题解答三	149
69.	QTP 常见问题解答四	150
70.	QTP 设计开发脚本	151
71.	设置共享对象库	152
72.	检查网页上文字显示的颜色	155
73.	QTP 录制如何识别编辑框	156
74.	How to search the string from the html source	157
75.	ADORecord.set 操作	159
76.	Object Property 参数化操作	161
77.	Quick Test Plus 工具	163
78.	Schedule the QTP tasks	164
79.	General the radom code	165
80.	ChildObjects 和 webTable 的使用	166
81.	怎样在 QTP 中打开右键菜单	167
82.	CaptureBitmap 方法	169
83.	校验对象是否存在 object.exist	170
84.	QTP 中&和_的用法。	171
85.	选取 webList 中某一项	172
86.	object.SetCaretPos 方法	172
87.	Similar to watir sample using description programming	173

自动化测试	Version:	1.3
文档	2006-5-17	

88. QTP 使用 outlook 发邮件

173

自动化测试	Version:	1.3
文档		2006-5-17

Technology Tips

1. QTP 描述性编程

【摘要】自动化功能测试是一种企业级的用于检验应用程序是否如期运行的功能性测试工具。通过自动捕获，检测，和重复用户交互的操作，能够辨认缺陷并且确保那些跨越多个应用程序和数据库的业务流程在初次发布就能避免出现故障，并且保持长期可靠运行。在市场上用的比较多的主要包括Mercury公司的WinRunner, QuickTest Professional 和 IBM 的 Rational Robot。笔者对于QuickTest Professional 相对较为熟悉，希望有机会向大家逐步介绍QuickTest Professional 中的一些要点及技巧。在本文里主要介绍了QuickTest Professional 中的一项核心内容Descriptive Programming，希望对大家有所帮助。在文中，为了方便起见，将QuickTest Professional 简称QTP。(本文是基于Quick Test Professional V8.0 而写)。

【关键词】

描述性编程 Descriptive Programming

功能测试 Functional Test

专家视图 Expert View

关键字视图 Keyword View

对象模型 Object Model

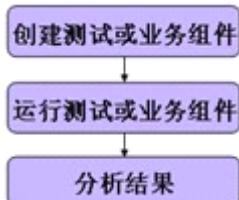
运行时对象 Run-Time Object

测试对象 Test Object

【正文】

. QTP 功能测试基本方法

我们简单介绍一下有关功能测试的基本方法，这实际上对于所有自动化功能测试产品来说都是一样的。一般情况下，用QTP来进行功能测试的基本方法主要包括三个主要阶段：



1、创建测试或组建

首先可以通过在应用程序或网站上录制会话，或者建立对象库并使用关键字驱动功能向关键字视图中手动添加步骤来创建测试或组件。在QTP里面我们可以通过两种方式添加步骤来创建测试或组件：

- 在应用程序或网站上录制会话。
- 建立对象库并使用这些对象在关键字视图或专家视图中手动添加步骤

然后在在测试或组件中插入检查点，检查页面、对象或文本字符串中的特定值或特征，通过它可以标识网站或应用程序是否正常运行。或是通过用参数替换固定值扩展测试或组件的范围。提供数据表中的数据，定义环境变量和值，定义测试、组件或操作参数和值，或者使用QTP生成随机数字或当前用户和测试数据等。

最后如果需要的话使用QTP中众多的功能测试功能来增强测试或组件或添加编写语句来实现更复杂的测试目标。

自动化测试	Version:	1.3
文档		2006-5-17

2、运行测试和组建

控制运行会话，帮助标识和消除测试或组件中的缺陷。使用“单步执行”、“单步跳过”和“单步退出”命令逐步运行测试或组件，或设置断点使测试或组件在预定点暂停。每当测试或组件在断点处停止时，可以在“调试查看器”中查看其变量的值。

3、分析结果

在运行测试或组件之后，通过两种方式可以查看其结果：在“结果”窗口中查看结果；自动报告在运行会话过程中检测到的缺陷，可能的话并上报到其他缺陷管理产品中。

. 试图与对象模型

在介绍 QTP 中的 Descriptive Programming 前，我们有必要先介绍一下 ExpertView 及在 ExpertView 里进行编码的一些基本知识。

在 QTP 里面提供了两种视图，第一种我们把它称为 KeywordView（关键字视图，在早期的版本中称为 TreeView），第二种把它成为 ExpertView（专家视图），这两种视图分别是针对两种类型的人进行使用的。

1、KeywordView（关键字视图）

通过关键字视图，QTP 提供了一种模块化的表格格式创建和查看测试或组件的步骤。每个步骤在关键字视图中都是一行，这样用户可以轻松的修改任何一部分组成。

在录制会话过程中，用户在应用程序上执行的每个步骤在关键字视图中记录为一行。例如，在 51testing 的页面上执行的下列三个步骤：

- 在“用户名”编辑框中输入 zhoda02。
- 在“密码”编辑框中输入加密字符串 41c630a213508cd49eb35089db1b893144b9。
- 单击“登录”按钮。

那么，关键字视图将包含下列行：



很显然，关键字视图非常直观有效，使用的人可以很清晰的看到被录制对象的录制层次及运行步骤，比较适合那些对于业务操作流程熟悉的人员使用。但是，如果需要一些增强型的操作，那就需要切换到专家视图里进行了。

2. ExpertView（专家视图）

QTP 在关键字视图中的每个节点在专家视图中对应一行脚本。上面例子对应的脚本如下：（删除后一句是因为前后重复，一句可以说明问题）

```
Browser("51Testing 软件测试网：软件测试的专业网站").Page("51Testing 软件测试网：软件测试的专业网站")
  .WebEdit("username").Set "zhoda02"
```

```
Browser("51Testing 软件测试网：软件测试的专业网站").Page("51Testing 软件测试网：软件测试的专业网站")
  .WebEdit("password").SetSecure "41c630a213508cd49eb35089db1b893144b9"
```

```
Browser("51Testing 软件测试网：软件测试的专业网站").Page("51Testing 软件测试网：软件测试的专业网站")
  .WebButton("登录").Click
```

对于 QTP 来说，其核心编码语言是 Visual Basic Script，因此，如果用户熟悉 VBScript，可以运用自如的添加和更新语句，并通过编程方式增强测试和脚本，而这一切必须在专家视图中完成。

自动化测试	Version: 1.3
文档	2006-5-17

更为重要的是，有些操作是必须在专家视图中才可以完成的，例如：要处理动态对象、客户化报告、获取对象运行时的属性值（Run-time Value）等等，这些都必须通过专家视图中的 VBScript 编码完成。

然而，QTP 里所有的操作都是基于对象进行的，所以我们必须对对象模型有一个基本了解，才可以在专家视图内进行 Descriptive Programming。

3、测试对象模型

测试对象模型是一大组对象类型或类，QTP 用这些对象类型或类来表示应用程序中的对象。每个测试对象类都有一个可以唯一标识属于该类的对象的属性列表，以及一组 QTP 可以对其进行录制的方法。它包括测试对象（Test Object）和运行时对象（RunTime Object）。

测试对象是 QTP 在测试或组件中创建的用于表示应用程序中的实际对象的对象。QTP 存储有关该对象的信息，这些信息有助于它在运行会话期间标识和检查该对象。

运行时对象是网站或应用程序中的实际对象，在运行会话期间执行针对该对象的方法。

如果录制时执行应用程序的相应操作，则一般情况下 QTP 将完成以下操作：

- 标识 QTP 测试对象类（表示执行了操作的对象），并创建相应的测试对象。
- 读取应用程序中对象属性的当前值，然后将属性和属性值列表与测试对象一起存储。
- 选择该对象的唯一名称，一般使用该对象某个重要属性的值。
- 使用适当的 QPT 测试对象方法录制对对象执行的操作。

例如，假定使用以下 HTML 源代码单击“查找”按钮：

```
<INPUT TYPE="submit" NAME="Find" VALUE="Find">
```

QTP 将单击的对象标识为 WebButton 测试对象。它将创建一个名为 Find 的 WebButton 对象，然后为该 Find WebButton 对象录制下列属性和属性值，同时还会录制对 WebButton 的 Click 方法。

属性	值
type	Find
name	reset
html tag	INPUT

在关键字视图及专家视图中显示内容分别为：

项	操作	文档
Action1		
Mercury - Business Tech...		
Mercury - Business ...	search_right Click	单击 "search_right" image.

```
Browser("Mercury Interactive").Page("Mercury Interactive").WebButton("Find").Click
```

运行测试或组件时，QTP 通过其测试对象类及其描述（一组用于唯一标识该对象的测试对象属性和属性值）来标识应用程序中的每个对象。测试对象及其属性和属性值的列表存储在对象库中。例如在上例中，QTP 将在运行会话期间在对象库中搜索 WebButton 对象，通过名称 Find 来查找其描述。QTP 根据找到的描述，在应用程序中查找 WebButton 对象，该对象带有 HTML 标记 INPUT、类型为 submit、值为 Find。找到对象后，它将对其执行 Click 方法。

在这样一组对象模型的基础上，QTP 为各类应用对象都提供了一组方法和属性，例如 Web Objects, Windows Objects, SAPGUI Objects, ActiveX, Java 等。下面是一些 Web Objects 的方法和示例：

对象	方法
----	----

自动化测试	Version:	1.3
文档		2006-5-17

Browser	Check
Frame	Click
Image	Exist
Link	GetCellData
Page	GetProperty
WebArea	GetROProperty
WebButton	Mouseover
WebCheckBox	RowCount
WebEdit	Select
WebList	Set
WebRadioGroup	SetProperty
WebTable	Submit

例 1：获取单元格中的值

```
thisText = Browser(...).Page(...).Frame.(...).WebTable("sample").GetCellData(2, 1)
```

例 2：获取图片的名称

```
ObjectName = Browser(...).Page(...).Image("Find").GetProperty("Name")
```

例 3：检查某个对象是否存在，如果存在弹出对话框说明对象存在。

```
If Browser("Browser").Page("Page").Applet("login.html").JavaEdit("username").Exist Then
```

```
    MsgBox("The object exists.")
```

```
End if
```

. 描述性编程 (descriptive programming)

1、descriptive programming 概述

通常情况下，当在录制一个操作时，QTP 会将被操作对象加入到对象库里 (Object Repository)。一旦对象存在于对象库里，我们就可以在专家视图里通过添加相关的对象方法来对该对象进行操作。我们可以通过引用层次型对象库里的对象描述 (Object Description) 来添加相应的方法。

因为 QTP 对象库中的每个对象都具有唯一名称，所以在引用时对象名是必须需要指定的。然后在测试运行期间，QTP 在对象库中根据这个对象的名称和父对象来查找对象，并使用为这个测试对象存储的测试对象描述，在网站或应用程序中标识该对象。

例如我们用 QTP 录制 Yahoo Mail 登录情况时我们需要输入用户名，于是在录制时我们就会录下一个 WebEdit 对象，它的缺省逻辑名为“login”，该编辑字段位于名为“Yahoo! Mail – The best”的页面上，并且该页面在浏览器中使用名称 Yahoo! 进行录制。如图所示，即为录制时的对象库的内容：

自动化测试	Version:	1.3
文档		2006-5-17



那么如果我们想要应用该对象，就可以在专家视图输入以下信息：

```
Browser("Yahoo!").Page("Yahoo! Mail - The best").WebEdit("login").Set "xxx"
```

或者我们也可以调用一些方法，获取改对象在运行时的对象名，如：

```
Browser("Yahoo!").Page("Yahoo! Mail - The best").WebEdit("login").GetROProperty("name")
```

然而，我们可以发觉到，上面的例子在处理对象时，对象已经存在于对象库里，因此我们可以应用这个对象的逻辑名。实际使用中，情况往往并非如此简单，我们经常会遇到很多在页面上动态产生的对象，换而言之，对象库里没有这些对象，我们也无法引用。因此我们必须采用其他的技术来完成这类操作，这也就是我们需要讲解的 Descriptive Programming。

为了满足上面提到的动态对象的处理问题，QTP 允许用户通过将对象属性编码到测试脚本里来动态识别对象，这就是我们通常意义上称为的 Descriptive Programming。通过这种方式，我们可以指示 QTP 不通过引用对象库和对象名来对实际对象进行操作。具体操作中，我们只需要为 QTP 提供对象的一组属性和值，这样 QTP 就可以识别相应的对象并对其进行操作。这相当于，告诉 QTP 要识别对象的一些关键特征，根据这些特征 QTP 就可以一一匹配然后识别出来这个对象。

而且，更为重要的是，通过这种 Descriptive Programming 的方式，还可以让 QTP 识别具有某些相同属性的对象。我们先来举个例子来看一下：我们假设当前的 Windows 系统中打开了若干的 Yahoo 主页面（多于一个），现在我们要关闭所有的正在浏览 Yahoo 主页面的浏览器。

对于上面那个例子来说，我们先看一个简单一点的情况，假设只有且仅有一个 Yahoo 主页面：那么我们可以用下面的方法来

```
Window("Text:=Yahoo! - Microsoft Internet Explorer").Close
```

我们可以看到语句里我们要查找的对象是 Window 窗口标题为 “Yahoo! - Microsoft Internet Explorer”，然后把它关闭，具体的语法说明我们稍后为解释。但是上面的语句仅仅适合前面提到的条件“只有且仅有一个 Yahoo 主页面”，如果有多个同样的窗口就会出错，原因是通过语句可以匹配到多个对象，而 QTP 不知道应该对哪个对象进行关闭动作。我们需要进一步的缩小匹配范围：

```
Dim i
i = 0
while (Window("Text:=Yahoo! - Microsoft Internet Explorer", "index:=" & i).exist)
    Window("Text:=Yahoo! - Microsoft Internet Explorer", "index:=" & i).close
    i = i + 1
wend
```

这里我们可以看到，对于具有相同属性的对象，我们可以通过 index 参数来对其进行区别，第一个对象为 index=0，第二个为 index=1 等等，依次类推。当然我们还可以通过 CreationTime 和 Location 参数来定位对象，这里就不详细叙述了。

通过上面的例子，我们对 Descriptive Programming 有一个基本了解了，下面我们详细讲解一下 Descriptive Programming：在具体实现中，我们有两种类型的 Descriptive Programming 方法。可以列出直接在测试语句中描述对象的属性和值的集合；或者向 Description 对象中添加属性和值的集合，然后在语句中输入 Description 对象的名称。下面我们分别举例介绍。

2、直接在语句中输入编程描述

自动化测试	Version:	1.3
文档		2006-5-17

通过多个指定描述对象的 `property:=value` 对，可以直接在语句中描述对象，这是最直接有效的方法。

常规语法为：

```
TestObject("PropertyName1:=PropertyValue1", "...", "PropertyNameX:=PropertyValueX")
```

`TestObject` – 测试对象的类。

`PropertyName:=PropertyValue` – 测试对象的属性及其值。各个 `property:=value` 对之间应用逗号和引号分开。

例如：以下语句指定 Mercury Tours 页面中名为 author 且索引值为 3 的 WebEdit 测试对象。当测试运行时，QTP 将查找具有匹配属性值的 WebEdit 对象，并输入文本 jojo。

```
Browser("Mercury Tours").Page("Mercury Tours").WebEdit("Name:=Author", "Index:=3").Set "Mark Twain"
```

我们也可以从描述中的特定位置（从 `Page` 对象描述开始）开始使用 Descriptive Programming。

```
Browser("Mercury Tours").Page("Title:=Mercury Tours").WebEdit("Name:=Author", "Index:=3").Set "jojo"
```

此外，如果我们希望在一个测试或组件中多次使用相同的 Descriptive Programming，则可以将创建的对象赋值给变量，这样使用会方便很多。

例如：我们需要完成下面一系列操作

```
Window("Text:=HyperSna").WinButton("Caption:=日期").Click
```

```
Window("Text:=HyperSna").WinButton("Caption:=时间").Click
```

```
Window("Text:=HyperSna").WinButton("Caption:=确定").Click
```

那么，为了方便其见，我们可以将 `Window("Text:=HyperSna")` 赋值到一个变量，然后再使用，参见下面的代码：

```
Set WinHyper = Window("Text:=HyperSna")
```

```
WinHyper.WinButton("Caption:=日期").Click
```

```
WinHyper.WinButton("Caption:=时间").Click
```

```
WinHyper.WinButton("Caption:=确定").Click
```

如果使用了 VBScript 里面的 `With` 语句，还可以简化为以下代码：

```
With Window("Text:=HyperSna")
```

```
    .WinButton("Caption:=日期").Click
```

```
    .WinButton("Caption:=时间").Click
```

```
    .WinButton("Caption:=确定").Click
```

```
End With
```

下面我们来看一个更为详细的例子，在 QTP 产品缺省安装里面自带了一个网上订机票的示例称为 Mercury Tour，我们看一下在订票过程中何时需要用 Descriptive Programming。

首先登入系统后，如果需要订票，就要先搜索航班，此时系统要求输入订票乘客的数量，假设我们在第一次录制脚本时选择了 1 个 Passenger，并成功完成订票。然后，我们需要参数化乘客数量来测试订票系统，我们会发现回放会失败。原因在于，不同的乘客的数量导致在订票时需要输入每个乘客的姓名，而录制时，只输入了一个乘客的姓名。而乘客姓名的输入框是随着乘客数量的变化而动态生成的，我们不可能从对象库里得到没有录制的对象，因此必须使用 Descriptive Programming。

自动化测试	Version:	1.3
文档		2006-5-17

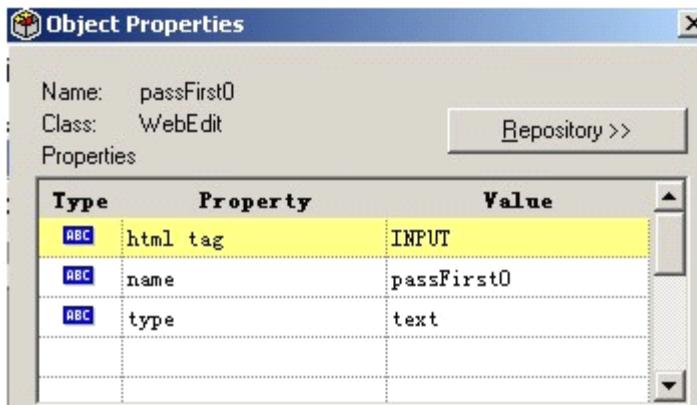
The screenshot shows a flight booking form titled "FLIGHT FINDER". The "Flight Details" section includes fields for Type (Round Trip selected), Passengers (set to 2), Departing From (Acapulco), On (Dec 20), Arriving In (Zurich), and Returning (Dec 21). Below this is a "Passengers" section containing two sets of First Name and Last Name fields. At the bottom is a "Credit Card" section.

在录制单个乘客时，我们得到的录制语句是：

```
Browser("Welcome: Mercury Tours").Page("Book a Flight: Mercury").WebEdit("passFirst0").Set "Michael"
```

```
Browser("Welcome: Mercury Tours").Page("Book a Flight: Mercury").WebEdit("passLast0").Set "Wang"
```

显然 WebEdit("passFirst0") 和 WebEdit("passLast0") 是录制时产生的对象并存放到对象库里的。通过对象库，我们可以看到对象的属性如下



系统对于发生多个 FirstName 时，命名规则是 passFirst0, passFirst1…依次类推。因此只要通过简单的 Descriptive Programming 就可以完成动态 FirstName 与 LastName 的识别工作。这里我们假设参数化的乘客数已经赋值给 intPassNum，下面是脚本中的关键语句：

```
counter = 0
For i = 0 to (intPassNum)
  Browser("Find a Flight:").Page("Book a Flight:").WebEdit("name:=passFirst"&i).Set "Michael"
  Browser("Find a Flight:").Page("Book a Flight:").WebEdit("name:=passLast"&i).Set "Wang"
```

自动化测试	Version:	1.3
文档		2006-5-17

counter = counter + 1

Next

3、使用 description 对象

使用 Description 对象可以返回包含一组 Property 对象的 Properties 集合对象。Property 对象由属性名和值组成。然后，可以在语句中指定用返回的 Properties 集合代替对象名。（每个 property 对象都包含一个属性名和值）。

要创建 Properties 集合，可以使用以下语法输入 Description.Create 语句：

```
Set MyDescription = Description.Create()
```

创建 Properties 对象（例如，以上示例中的 MyDescription）后，就可以输入语句，以便在运行会话期间在 Properties 对象中添加、编辑、删除或检索属性和值。这样，就可以在运行会话期间，使用动态方法确定哪个属性以及多少个属性应包含在对象描述中。

在 Properties 集合中填充一组 Property 对象（属性和值）后，可以在测试语句中指定用 Properties 对象代替对象名。

例如，假设我们需要完成以下一个操作：

```
Window("Error").WinButton("text:=OK", "index:=1").Click
```

我们可以通过 Description 对象来实现同样的功能，参加下面的代码：

```
Set MyDescription = Description.Create()
MyDescription("text").Value = "OK"
MyDescription("index").Value = 1
Window("Error").WinButton(MyDescription).Click
Set MyDescription = Nothing
```

【小结】以上是对 QTP 中有关处理动态对象中的 Descriptive Programming 的简单介绍，希望对大家能够有所帮助，就总体而言，如果能够熟练掌握 Descriptive Programming，那么有很多实际中的问题就可以迎刃而解。当然 Descriptive Programming 只是 QTP 中的一个功能，QTP 在实际功能测试中还有很多强大的功能，作为 QTP 学习的一个系列有机会我会一一介绍。

2. 描述性编程 2

最近有几位朋友问到描述性编程的问题，那么我们今天就来探讨一下 QTP 中的描述性编程吧。当我们并不想 QTP 依靠对象库来识别对象时，我们就可以借用描述性编程的方式。把对象的属性及属性值直接搬到代码中去，让 QTP 直接通过脚本中的这些属性及属性值就能识别到这个对象，而不必再去匹配对象库。总的来说，如果您希望在未存储在对象库中的对象上执行操作，则这种编程描述将是非常有用的。或者还可以使用编程描述在具有某些相同属性的多个对象上执行相同的操作，或者在其属性与运行会话期间动态确定的描述相匹配的对象上执行操作。比如：在我的[QTP 参数化实例](#)文章中，已经运用过描述性编程了。由于参数化的时候受到了对象库的制约，所以也采用了描述性编程的形式，让要参数化的对象不再受到对象库的制约。

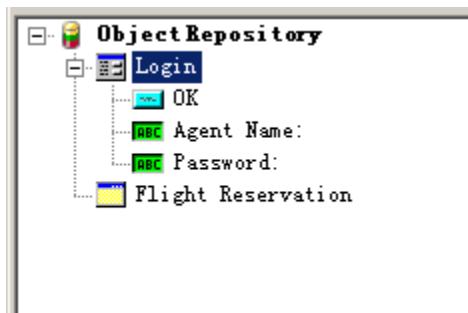
接下来，通过一个具体的例子，再来看看描述性编程的形式。打开 QTP 录制 Fight 的登录框。代码如下：

```
Dialog("Login").WinEdit("Agent Name:").Set "mercury"
```

自动化测试	Version:	1.3
文档		2006-5-17

```
Dialog("Login").WinEdit("Password:").SetSecure "45a1e193f9dac3bf91f3ba7c1dd250a48a33eabf"
Dialog("Login").WinButton("OK").Click
Window("Flight Reservation").Close
```

对应的对象库, 如图:



然后我们把对象库中, 这些对象的属性及属性值从对象库搬到脚本中, 代码修改成为:

```
Dialog("text:=Login").WinEdit("attached text:=Agent Name:").Set "mercury"
Dialog("text:=Login").WinEdit("attached text:=Password:").SetSecure
"4414d23c42046da1ea3a895f1518b9c476b2b225"
Dialog("text:=Login").WinButton("text:=OK").Click
Window("regexpwndtitle:=Flight Reservation").close
```

把对象库中对象删除掉, 如下图:



再来执行脚本, QTP 执行成功. 这就是描述性编程. 当然值得注意的是, 如果父对象用描述性编程表示了, 那么后面的对象都要用描述性编程表示了. 如:

```
Dialog("text:=Login").WinEdit("attached text:=Agent Name:").Set "mercury"
```

这边 Dialog("text:=Login")用描述性编程表示了, 那么 WinEdit("attached text:=Agent Name:")就必须用描述性编程表示了. 因为在对象库中, 子对象是依靠父对象存在的. 所以如果 Dialog("text:=Login")这个父对象从对象库中删除了, 那么后面的子对象也同时被删除了. 反过来子对象删除就不影响父对象. 所以子对象可以单独用描述性编程表示, 而不需牵连其他对象.

自动化测试	Version:	1.3
文档		2006-5-17

此文来源于 51testing 博客，转载请注明出处为 51testing 博客

原始链接: http://blog.51testing.com/?26649/action_viewspace_itemid_2625.html

3. 函数

自动化测试	Version:	1.3
文档	2006-5-17	

自动化测试	Version:	1.3
文档	2006-5-17	

3.1. TP 中几个截取字符串的函数

自动化测试	Version:	1.3
文档		2006-5-17

Left 函数 (Right 函数就是从右边开始算起)

返回指定数目的从字符串的左边算起的字符。Left(string, length)

参数

string: 字符串表达式，其最左边的字符被返回。如果 string 参数中包含 Null，则返回 Null。

Length: 数值表达式，指明要返回的字符数目。如果是 0，返回零长度字符串 ("")；如果大于或等于 string 参数中的字符总数，则返回整个字符串。

例子：

```
Dim MyString, LeftString
MyString = "VBscript"
LeftString = Left(MyString, 3) ' LeftString contains "VBS".
*****
```

Mid 函数

从字符串中返回指定数目的字符。

Mid(string, start[, length])

参数：

string: 字符串表达式，从中返回字符。如果 string 包含 Null，则返回 Null。

Start: string 中被提取的字符部分的开始位置。如果 start 超过了 string 中字符的数目，Mid 将返回零长度字符串 ("")。

Length: 要返回的字符数。如果省略或 length 超过文本的字符数（包括 start 处的字符），将返回字符串中从 start 到字符串结束的所有字符。

例子

```
Dim MyVar
MyVar = Mid("VB script is fun!", 4, 6) ' MyVar contains "script".
*****
```

InStr 函数 (InStrRev 函数相反从最后向前起)

返回指定的字符串在另一字符串中最先出现的位置。

InStr([start,]string1, string2[, compare])

参数：

start: 起始位置，默认从第一位

string1: 主体字符串，从左向右查找。如果 string1 为 Null，则返回 Null。

string2: 查找的字符串，如果 string2 为 Null，则返回 Null。找不到就返回 0。

compare: 0 是二进制比较，1 是文本比较。0 为缺省值。个人感觉区别就是在大小写。

例子

```
Dim SearchString, SearchChar, MyPos
SearchString ="XXpXXpXXPXXP" ' String to search in.
SearchChar = "P" ' Search for "P".
MyPos = Instr(4, SearchString, SearchChar, 1) ' A textual comparison starting at position 4. Returns 6.
MyPos = Instr(1, SearchString, SearchChar, 0) ' A binary comparison starting at position 1. Returns 9.
MyPos = Instr(SearchString, SearchChar) ' Comparison is binary by default (last argument is omitted).
Returns 9.
MyPos = Instr(1, SearchString, "W") ' A binary comparison starting at position 1. Returns 0 ("W" is not found).
*****
```

Split 函数

在指定的 delimiter 参数出现的所有位置断开 String 对象，将其拆分为子字符串，然后以数组形式返回子字符串。

Split(expression[, delimiter[, count[, compare]]])

参数

expression: 主体字符串，也就是要被拆分处的字符或字符串。

delimiter: 拆分元素，默认是(" ")

count:Number [可选] 要放入数组中的项目数。

compare: 0 是二进制比较，1 是文本比较。0 为缺省值。

自动化测试	Version:	1.3
文档		2006-5-17

例子

```
Dim MyString, MyArray, Msg
MyString = "VBscrīptXisXfun!"
MyArray = Split(MyString, "x", -1, 1)
' MyArray(0) contains "VBscrīpt".
' MyArray(1) contains "is".
' MyArray(2) contains "fun!".
Msg = MyArray(0) & " " & MyArray(1)
Msg = Msg & " " & MyArray(2)
MsgBox Msg
```

4. "+" "&"的区别

今天在录脚本的时候，偶尔发现，以前一直使用的"+"

"+"和"&"的区别：

"+": 加法运算符

如果两个表达式都是 numeric，则相加；

如果两个表达式都是 strings，则连接；

如果一个表达式是 numeric，另一个是 string，把 string 转换成 numeric，再相加。

"&": 字符串连接符

把两个表达式都转换成 string，然后连接。

举个例子就很明白了，注释的部分是结果。

```
msgbox "1" & 3 '13
msgbox "1"+3 '4
```

自动化测试	Version:	1.3
文档		2006-5-17

```
msgbox "你好" & 3 '你好 3  
msgbox "你好" + "3" '你好 3  
msgbox "你好" + 3 'Type mismatch: [string: "你好"]
```

5. QTP 场景恢复

[QTP Recovery Scenario 简介](#)

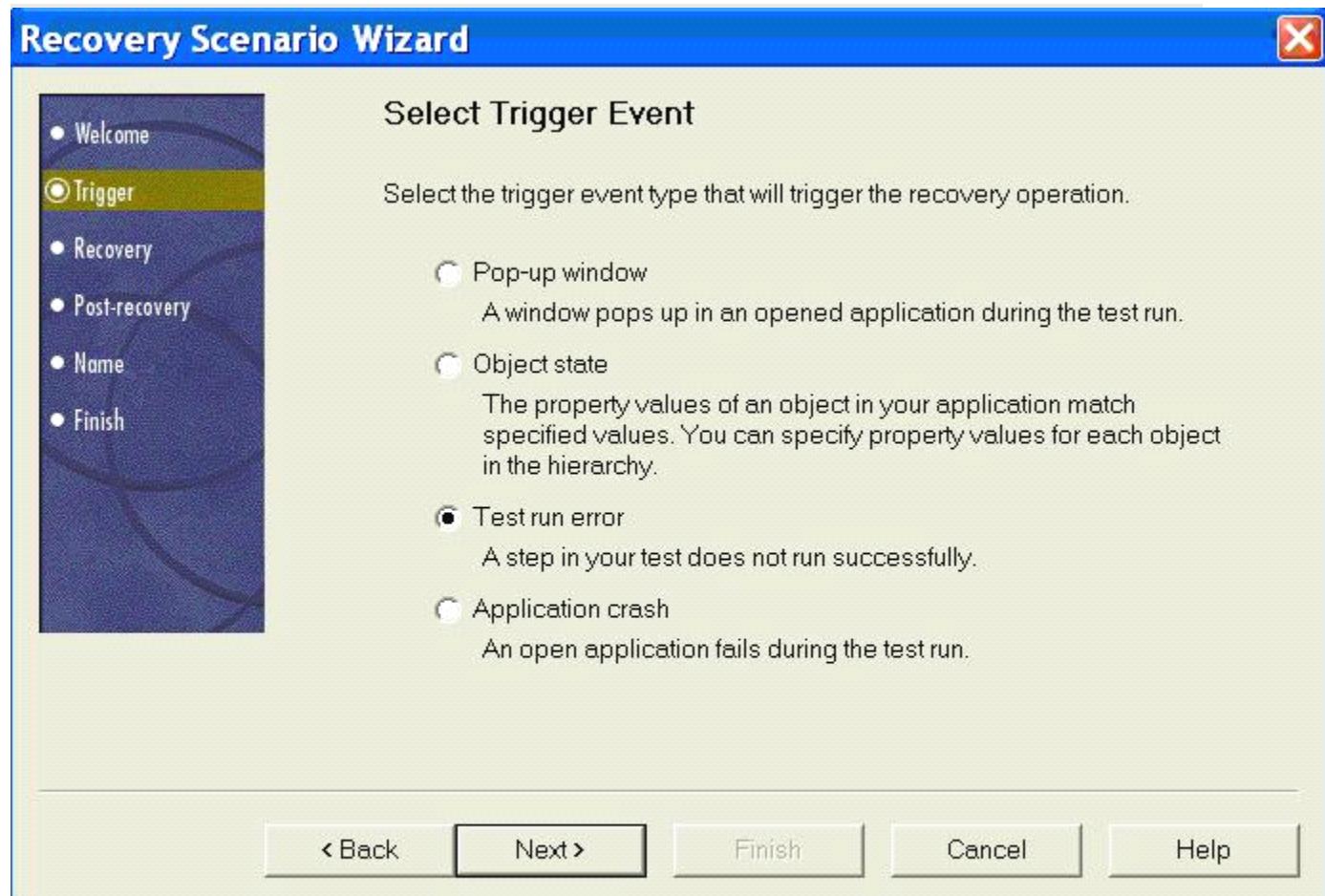
场景恢复可以用于应对[测试脚本](#)在运行的过程中出现的异常，在预估可能出现的异常状况下，添加对应的场景恢复，可以使脚本运行的更加通畅。以下是[QTP Recovery Scenario](#) 的简单介绍，希望对初学者有所帮助。

添加一个新的场景恢复，通过菜单 [Tools->Recovery Scenario Manager](#) 进入，主要分为以下四个步骤。

[步骤一：Trigger]

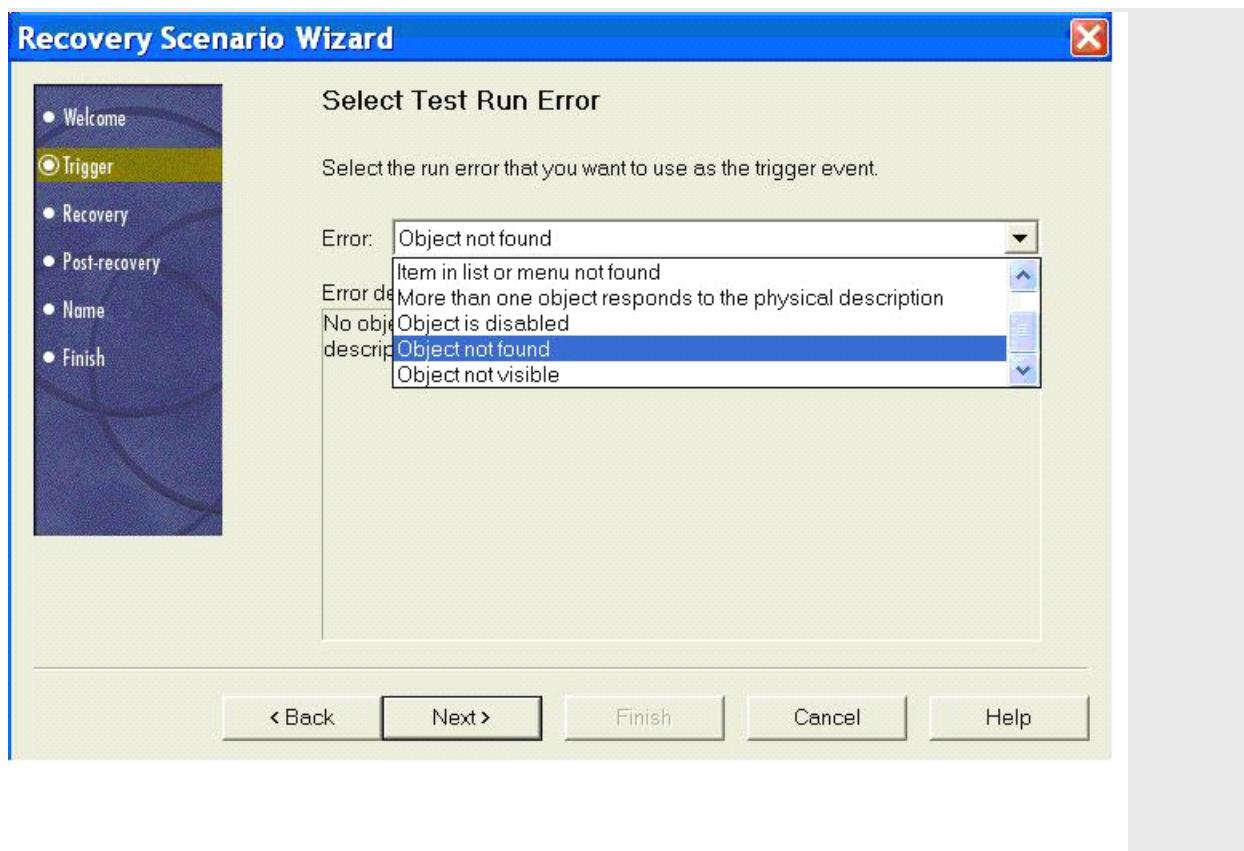
场景恢复机制提供了四种类型的触发事件，分别用来识别：弹出对话框、对象的特殊属性值、运行错误、[应用程序失败](#)。

自动化测试	Version:	1.3
文档	2006-5-17	



可以根据具体的需求来添加各个类型的恢复场景，每种类型的选项可以在添加向导中选择，如下图。

自动化测试	Version:	1.3
文档		2006-5-17

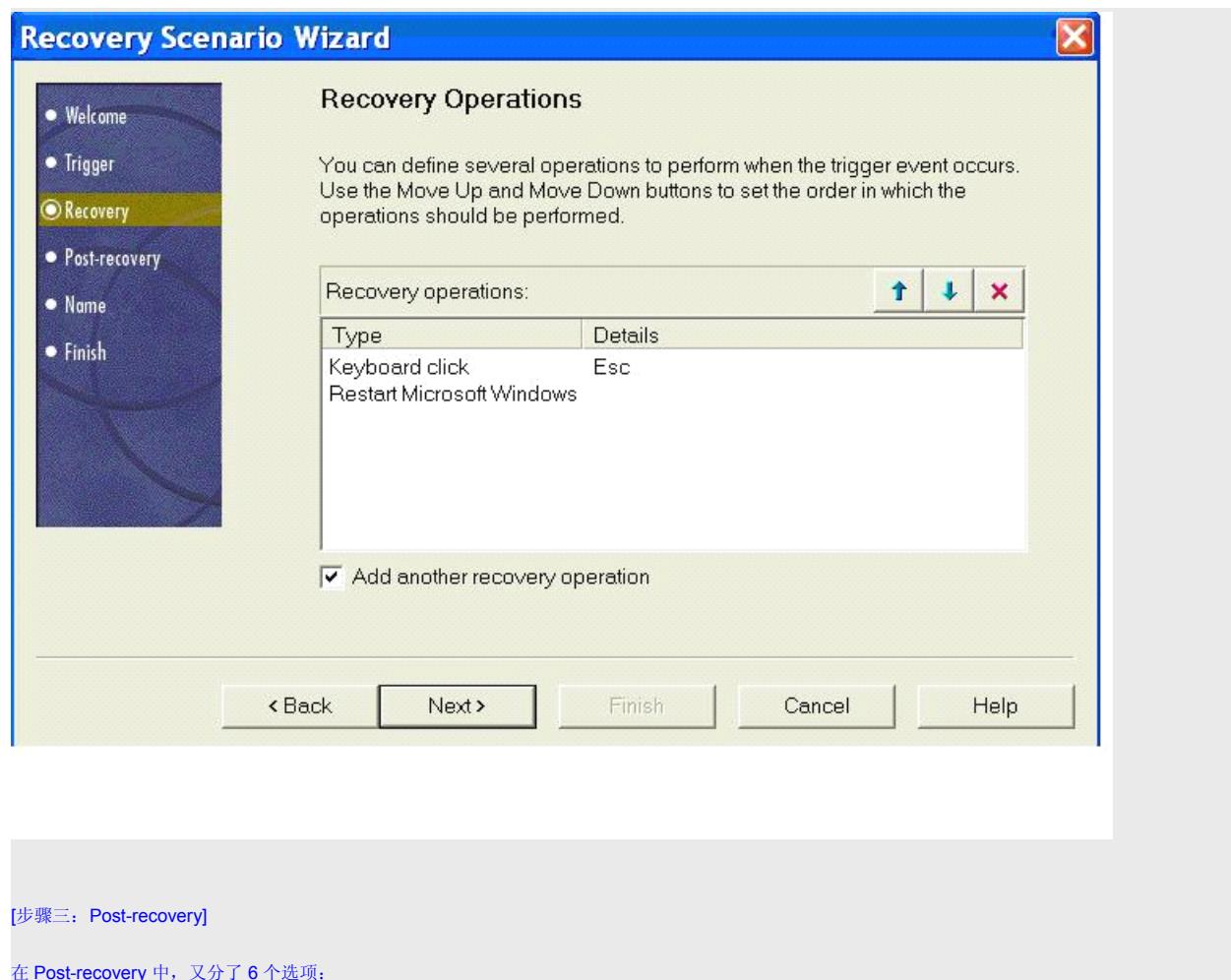


[步骤二: Recovery]

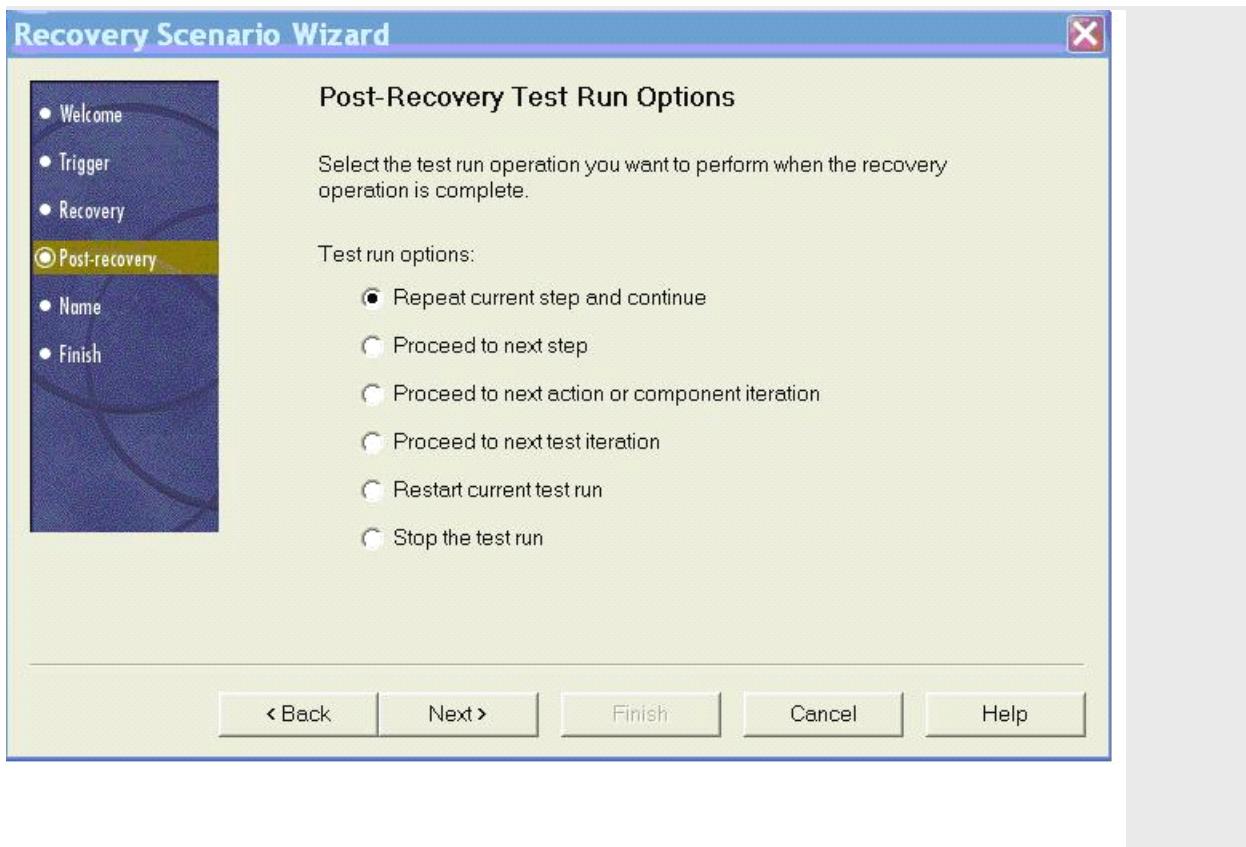
恢复的操作可以是自定义按钮操作，函数调用，关闭应用程序进程，重起机器等等，几乎涵盖了所有 QTP 的正常操作，按照向导进行设置，操作很方便。也可以添加多个恢复操作，且调整执行顺序，注意：重起系统总是排在最后一个。

添加操作完成，把 Add another recovery operation 前的按钮去掉，才可以进入下一步。

自动化测试	Version:	1.3
文档		2006-5-17



自动化测试	Version:	1.3
文档		2006-5-17



[Repeat current step and continue](#): 重复当前步骤然后继续向下

[Proceed to next step](#): 处理下一步

[Proceed to next action or component iteration](#): 处理下一个 Action, 或者组件的下一个循环

[Proceed to next test iteration](#): 处理该测试的下一个循环

[Restart current test run](#): 重新启动当前的测试

[Stop the test run](#): 终止测试运行

这些操作的差别在具体实践中体会吧，这里就不赘述了。

[步骤四: Name]

一切都设置好以后，就给你的场景恢复起个唯一的名字吧。也可以添加一些描述方便维护工作。

开始使用场景恢复，在 [Test->Settings->Recovery](#) 中添加设置好的恢复场景，然后选择激活方式（On every step/On error/Never）。

自动化测试	Version:	1.3
文档		2006-5-17

Test Settings

The dialog box has tabs at the top: Properties, Run, Resources, Parameters, Environment, Web, Recovery. The Recovery tab is selected.

Scenarios list:

- Scenario Name: testSce... Z:\CommonLib\MyRecoveryScenario.qrs

Scenario description: A large text area for describing the scenario.

Activate recovery scenarios: A dropdown menu currently set to "Never". The options are:

- On every step (highlighted)
- On error
- Never

Buttons at the bottom: OK, Cancel, Apply, Help.

Text below the dialog:

On every step, 只要出现恢复的场景，就执行场景恢复中的动作，然后继续。

On error, 在出现错误的时候，才查找是否符合待恢复的场景，如果是则执行恢复操作。

Never, 无论如何，都不运行场景恢复机制。

提供两个执行场景恢复的小例子：

例 1：

[配置]

1, 在 Recovery Scenario Properties 中，Post-Recovery Operation 选择 Repeat current step and continue
 2, 激活方式为 On every step

自动化测试	Version:	1.3
文档		2006-5-17

[现象]

只要出现恢复的场景，就执行场景的动作，并且重复执行出现该场景的那条语句，然后继续。适用于重复尝试必将成功的场景，否则死循环。

例 2：

[配置]

1，在 Recovery Scenario Properties 中，Post-Recovery Operation 选择 Repeat current step and continue

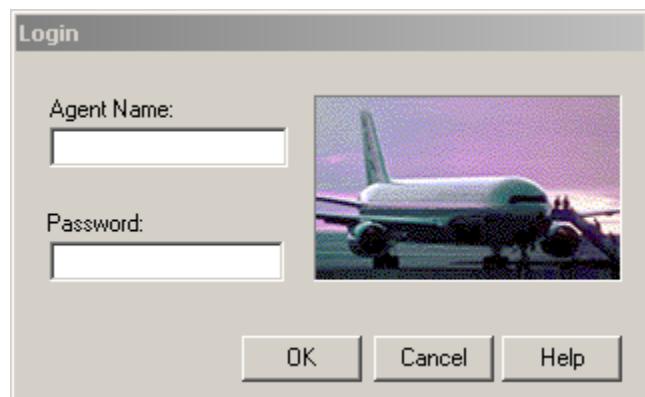
2，激活方式为 On error

[现象]

出现错误后，进入场景恢复机制，就执行场景的动作，（并没有重复执行出现该场景的那条语句，不知道为什么）直接执行下一条语句。

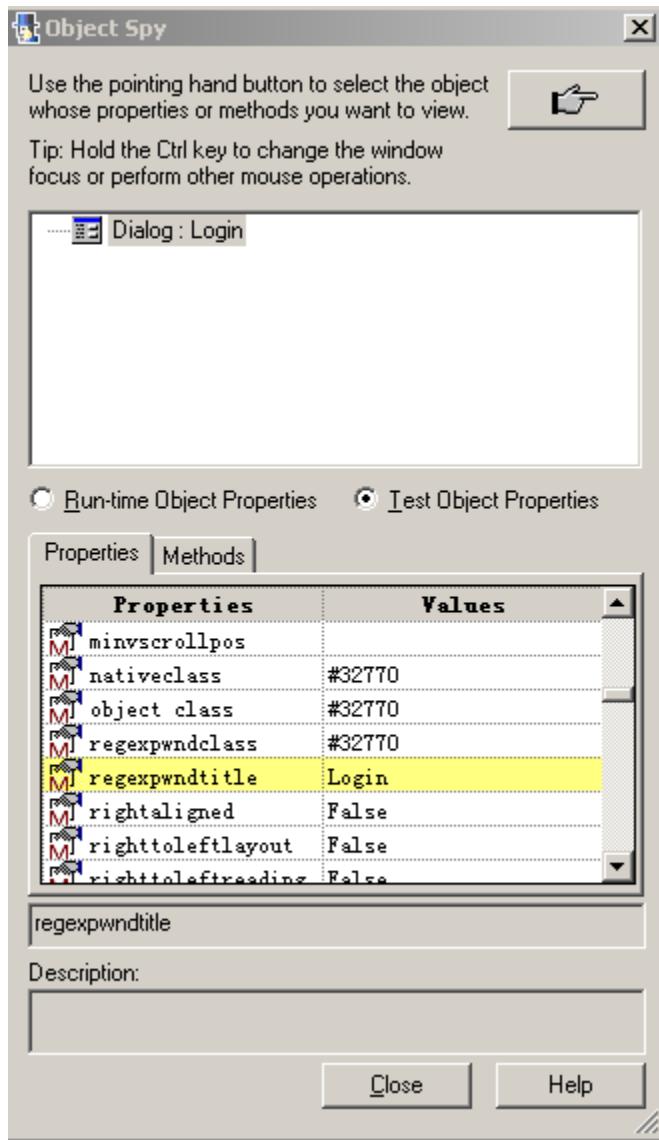
6. QTP 中测试环境的初始化

在我们运行脚本前，有时候忘了清理环境，可能被测对象已经存在了，QTP 再次运行时又重新打开了个相同的对象，导致后续步骤，由于 [QTP](#) 根据对象库中的对象匹配到了两个相同的对象，不知操作哪个对象了，于是报错。那我们有没有办法避免呢？答案当然是肯定的。我们可以在运行 [测试](#) 脚本前，先写段脚本来判断我们要测试的对象是否存在。如果存在可以先关闭，再执行重新打开的操作。接下去以 QTP 自带的 Flight 的登录框为例。



首先我们用 QTP 中的 `object spy` 来看一下这个窗口的属性，找到我们可以利用的属性

自动化测试	Version: 1.3
文档	2006-5-17



这边我想利用的就是这个窗口的名称属性及值，利用这个名称属性来告诉 QTP 该关闭哪个窗口属性.

代码如下：

```

Dim loginwin

Set loginwin=description.Create()          ' 创建对象名为 loginwin 的对象

loginwin( "regexpwndtitle").value="Login"   , 对象属性及值

Set child=desktop.ChildObjects (loginwin)

For i=1 to child.count

```

自动化测试	Version:	1.3
文档		2006-5-17

child(i-1).close ' 循环寻找这个对象，一次关闭

Next

需要更改的地方，我用蓝色表示了，大家也可以自己试试。不过有一点要注意就是，要执行关闭窗口的关系，这个也完全依照 windows 的特性来做。最上面的处在激活状态的先关，然后依次关闭其他的窗口。否则脚本会无法执行。可以把这段脚本放在脚本的最前面，实现测试环境的初始化

7. QTP 将数据库中的数据导入到 excel

8. 在 QTP 中将数据库的数据导入到 EXCEL

[QTP](#)

昨天遇到这个问题.自己也有点迷茫,在 51 上发帖后得到高人指点,在这里我将代码写出来,留个足迹,呵呵,谢谢风过无痕.

Set excelObj = CreateObject("Excel.Application") '创建一个对象

xlsUrl = "C:\QTP\0110\sql-excel-qtp\data.xls" '指定一个已经存在的 XSL 文件路径

excelObj.workbooks.open(xlsUrl) '打开 EXCEL

Set sheetNew = excelObj.sheets.item(2) '设置将数据放在 EXCEL 中的那一页上(索引从 1 开始)

'相关的**数据库**操作

Set conn = CreateObject("adodb.connection") set rs = createobject("adodb.recordset") conn.open

"Provider=SQLOLEDB.1;Password=qaqc;Persist Security Info=True;User ID=test;Initial Catalog=test;Data

Source=192.168.0.53" **sql**="select * from test" rs.open sql,conn

自动化测试	Version:	1.3
文档		2006-5-17

Dim i while not rs.eof

user_id = rs("user_id").value

user_name = rs("user_name").value

sheetNew.cells(i+1,1).value = user_id '列付值

sheetNew.cells(i+1,2).value = user_name '列付值

i = i + 1

rs.movenext

wend

excelObj.activeworkbook.saveas "C:\QTP\0110\sql-excel-qtp\data.xls" '保存数据

excelObj.quit '关闭对象

set excelObj = nothing '清空对象

rs.close '关闭数据库连接操作

set rs=nothing

conn.close

set conn = nothing

-----这就是导出到 E X C E L 的全部过程：），后面如果要用到这些数据的话，直接导入进来就可以了

自动化测试	Version:	1.3
文档		2006-5-17

9. QTP 将数据测试驱动

事件的场景：在[测试](#)过程中，我们可能用到[数据库](#)中一些初始的数据进行测试，所以我们就希望把数据按照我们的预先设计导入到一个 excel 里面，然后在后面的测试中引入进来，用于测试。

具体操作：

'创建 excel 的对象

```
Set ExcelObj = CreateObject("Excel.Application")
```

指定一个 excel 的格式，这个需要我们事先定制好

```
aatype="F:\templet.xls"
```

'打开 excel

```
ExcelObj.Workbooks.Open(aatype)
```

'设置操作的工作表

```
Set NewSheet = ExcelObj.Sheets.Item(2)
```

'以下是数据库操作

```
Dim rs,sq,pkey
```

```
set conn=createobject("adodb.connection")
```

```
set rs=createobject("adodb.recordset")
```

```
set rs1=createobject("adodb.recordset")
```

```
conn.open= "Provider=OraOLEDB.Oracle.1;Password= *;Persist Security Info=True;UserID= *;SERVER= *;Data Source= *;DBQ= *;"
```

```
sql="select * from table"
```

```
rs.open sql,conn
```

```
i=1
```

```
do while not rs.eof
```

```
values1=rs("pkey")
```

```
values2=rs("name")
```

```
NewSheet.Cells(i+1,1).Value =values1
```

```
NewSheet.Cells(i+1,2).Value =values2
```

自动化测试	Version:	1.3
文档		2006-5-17

i=i+1

rs.movenext

Loop

'保存 excel 数据

ExcelObj.ActiveWorkbook.Saveas "F:\data.xls"

ExcelObj.Quit

Set ExcelObj = Nothing

rs.close

set rs=nothing

rs1.close

set rs1=nothing

conn.close

set conn=nothing

这样的话，我们在别的[脚本](#)里面直接把 excel 的数据 import 进来就能使用了。

10. QTP 中设置共享对象库(9.0)

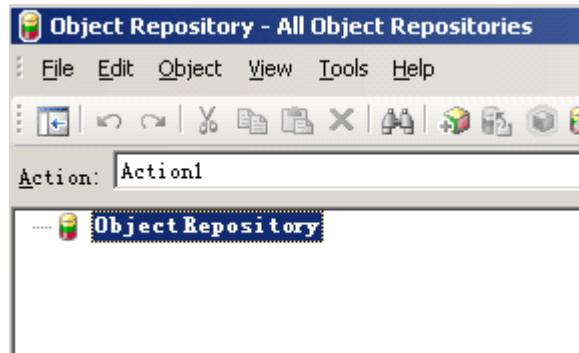
很多时候我们总是觉得管理 [QTP](#) 的[脚本](#)比较烦.因为除了要对代码的管理之外，还要保证 QTP 对象库的完整. 每一个用例脚本的生成，同时又都会生成一个对象库，那么是不是可以做一个公共的对象库来给各个用例脚本调用呢？接下来，我们就来实现共享对象库的应用.

第一步：把需要加到共享对象库中的各个用例脚本的对象库，分别导出成. tsr 文件.

操作方法:先用 QTP 打开已经录制完毕的脚本后，选择 Resources--> Object Repository .

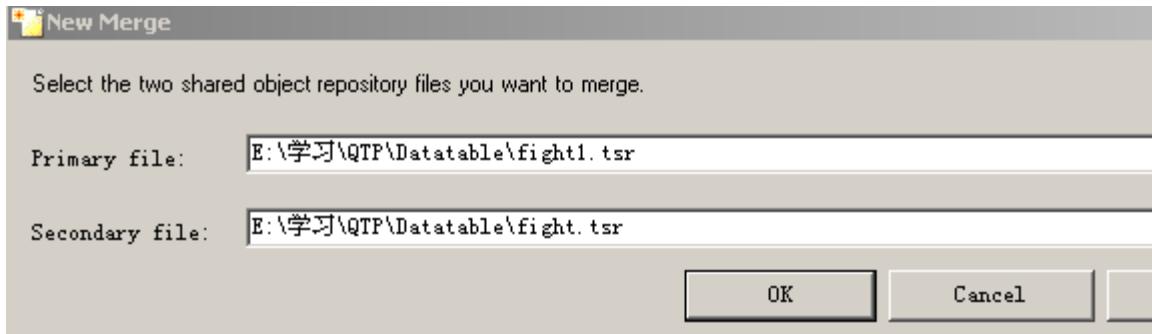
然后 file-->export....取名导出文件，如下图:

自动化测试	Version:	1.3
文档		2006-5-17



第二步：把需要加入到共享对象库中的各个用例脚本的对象库，合并对象及对象属性，形成一个大的共享对象库。

操作方法:打开 Resources -->Object Repository Manager. 找到 Tools 菜单，然后选择打开 Object Repository merge tool.选择两个要合并的对象库文件，进行对象合并。一次只能合并两个。所以如果这里你要合并多个对象库文件的话，先合并两个后，保存成一个新的。tsr 文件。然后再重复上面的操作，选择这个新的。tsr 文件和另一个准备好的对象库合并。如下图：

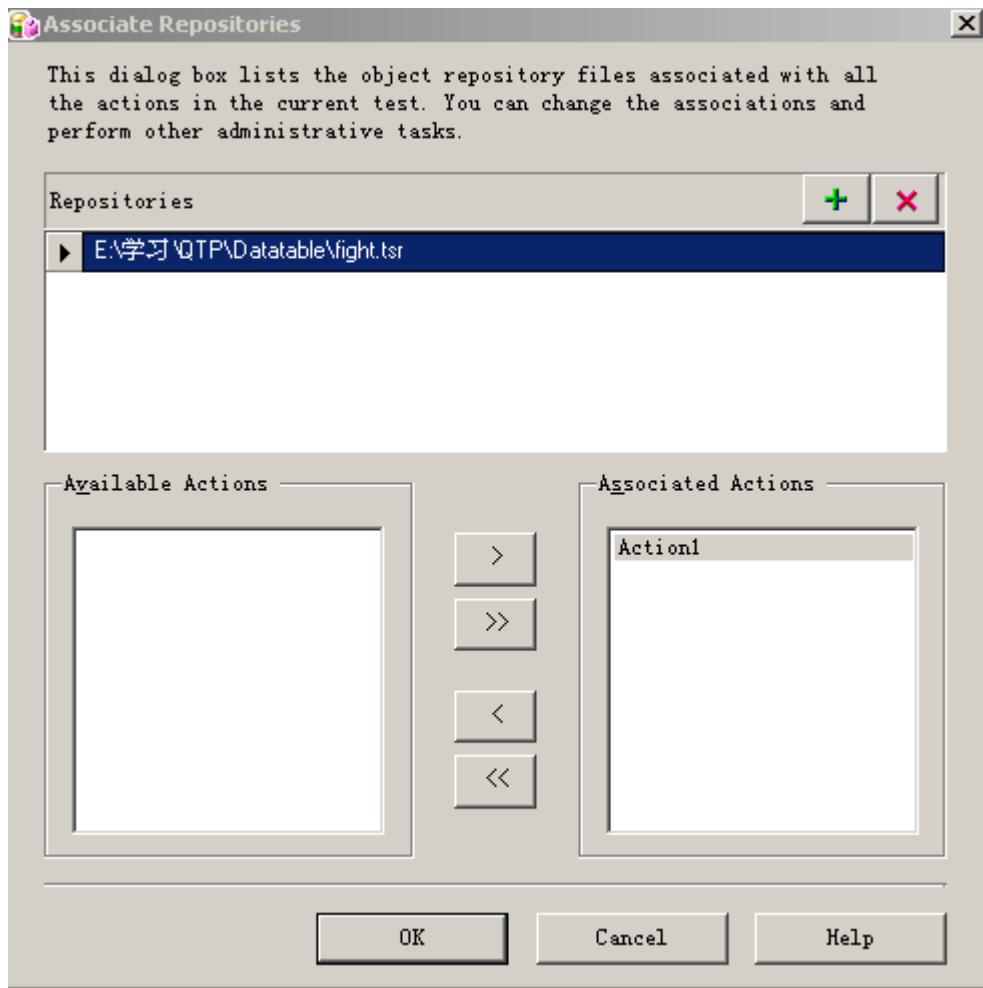


合并的时候，你会发现，在两个对象库文件中，相同的对象合并成一个，不同的对象，全部被完整增加进去。然后形成一个大的对象库。现在只要保存这个共享对象库。如果以后有新增的对象，可以重复上述操作。

第三步：调用上面保存好的共享对象库，给新的脚本使用。

操作步骤：Resources -->Associate Repository. 选择上面保存好的共享对象库的。tsr 文件。加入到 Associate Repository 中。如下图：

自动化测试	Version: 1.3
文档	2006-5-17



完成这一步后，其实整个共享对象库的操作已经完成。你在对象库中可以看到以后的对象。在录制新脚本的时候，如果对象已经存在，就不会再被记录，只有在这个对象库中没有的对象才会被记录进去。回放脚本，对象识别成功。

(注：此操作只适合与装有 QTP9.0 的版本。对于 8.2 版本不适合)

附：8.2 版本中的共享对象库操作

QTP8.2 的共享方法

1. 选择“**测试**”>“**设置**”>“**资源**”选项卡。
2. 在“对象库类型”区域，“共享”。

请指定要用作对象库文件的共享对象库文件。要指定文件，请输入对象库文件名，或单击浏览按钮并从“打开”对话框中选择资源文件。要新建共享对象库文件，请在“共享”框中输入新文件名。

自动化测试	Version:	1.3
文档		2006-5-17

对象库文件的默认文件扩展名是 **.tsr**, 但是文件可能使用任何扩展名。在“打开”对话框中浏览现有对象库文件时, 在“文件类型”框中选择“所有文件”。

3. 注: 请在新测试开始录制前进行设置, 已有测试的对象库模式不能进行修改。

11. QTP ActiveX 报错;

为什么我在我的第二个帖里点击评论回复后,评论数没有加 1?

所以为了醒目起见我还是上来新发个日志吧...

正如第二条日志中所说我用其它网站来做录制时会有个 ActiveX 的报错:"当前安全设置禁止运行该页中的 ActiveX 控件. 因此,该页可能无法显示"

是在成功录制完后,只要对 **QTP** 进行任何操作就会报错,点确定后倒是可以继续操作的,而且如果不点确定,报错框过子大约十秒就会自动消失,操作过程中这个报错框会时不时就跳出来,很烦...

嗯..找到问题所在了..照以下那样设置就不能再报这个错了:

Tools->Options->Active Screen->Advanced...-> 选中 Disabled 和 Load ActiveX controls->OK

12. 数据驱动在测试中的应用

最近看了不少**朋友**写的 blog 文章后, 就越显自己的文章比较稚嫩. 但我觉得这也是一种**学习**的方式, 现在抱着一种共同学习的态度来探讨一下数据驱动在 **QTP** 中运用. 很希望看我文章朋友能不吝啬的发表一下自己对这个方面的看法, 那真是万分感谢啊.

所谓数据驱动就是用一个数据文件把**测试脚本**驱动起来, 来达到更接近用户化更智能的测试. 其目的是把测试人员从维护复杂的脚本程序中解放出来, 只需维护好数据文件即可, 减少了很多修改脚本的麻烦. 下面讲一下通过四种途径来达到数据驱动.

1. datatable

自动化测试		Page 36
-------	--	---------

自动化测试	Version:	1.3
文档		2006-5-17

QTP 本身程序就给我们提供了这么一个数据表，我们可以把测试数据或测试用例填入这个数据表中。

如：设计用例

```
username    passwd
case1      mercury   mercury
case2      xxxxxxxx  xxxxxxxx
```

录制脚本

```
For i=1 to Datatable.GetRowCount
  Dialog("Login").WinEdit("Agent Name:").Set DataTable("username", dtGlobalSheet)
  Dialog("Login").WinEdit("Password:").Set DataTable("passwd", dtGlobalSheet)
  Dialog("Login").WinButton("OK").Click
  datatable.GlobalSheet.SetNextRow
Next
```

本例是验证一个登录系统，通过 `DataTable` 不同的用例设计，驱动起这段脚本，达到测试的效果。当然上面的例子中还少一个很重要的步骤，那就是结果比较，如果不能进行结果比较的[自动化测试](#)不能够称为自动化测试。

当然我们这里主要讲的是数据驱动，所以不在对上面的例子进行补充。

2 . 文本文件

我们可以把文本文件当成数据文件，通过对文本文件的读写操作，来实现数据驱动。

例：文本文件内的内容

```
mercury,mercuy
```

读文件的代码

```
Function writeorderno(orderno)
  Dim fso, myfile,username,passwd
```

自动化测试	Version:	1.3
文档		2006-5-17

```

Set fso=CreateObject("scripting.FileSystemObject")

Set myfile=fso.openTextFile("C:\testing.txt",1,false)

tmp=split(myfile.readline,",")

username=tmp(0)

passwd=tmp(1)

myfile.close

End Function

```

写文本文件的代码

```

Function writeorderno(orderno)

Dim fso, myfile

Set fso=CreateObject("scripting.FileSystemObject")

Set myfile=fso.openTextFile("C:\result1.txt",8,false)

myfile.writeline orderno

myfile.close

End Function

```

3. EXCEL 文件

我们可以把 EXCEL 文件当成数据文件，通过对 EXCEL 文件的读写操作，来实现数据驱动.

可以把 EXCEL 文件当作对象的方式来完成写的操作

```

Dim Excel,ExcelSheet

Set Excel=CreateObject("Excel.Application")

Set ExcelSheet=CreateObject("Excel.Sheet")

ExcelSheet.Application.visible=true

ExcelSheet.ActiveSheet.Cells(1,1).value= 1

ExcelSheet.ActiveSheet.Cells(1,2).value= 2

ExcelSheet.ActiveSheet.Cells(1,3).value= 3

```

自动化测试	Version:	1.3
文档		2006-5-17

Excel.Save "C:\test.xls"

Set ExcelSheet=Nothing

用 ADO 的方式连接 EXCEL 文件来做读的操作

```
Dim conn,input,filename
filename="D:\公基本情况(tb_gsgk)-标准格式.xls"
Set conn= CreateObject("ADODB.Connection")
conn.Open "Provider=Microsoft.Jet.OLEDB.4.0;Persist Security Info=False;Data Source=""" &filename&"";Extended Properties='Excel 8.0;hdr=yes'"
Set input= CreateObject("ADODB.Recordset")
input.Open "select * from [公基本情况-标准格式$]",conn,2,2
input.Close
Set input=nothing
```

4 . 数据库

可以利用设计数据表，把测试数据和测试用例放在数据表里，用 ADO 或者其他任何可以访问连接数据库的方式连接数据库，来实现数据驱动

```
Dim res,cmd,sql
Set Res=createobject("adodb.recordset")
Set Cmd=createobject("adodb.command")
Cmd.activeconnection="Provider=SQLOLEDB.1;Password=111111;Persist Security Info=True;User ID=sa;Initial Catalog=xhq;Data
Source=192.168.191.142"      ' 这句话是连接数据库的数据源，要做修改
Cmd.CommandType = 1
sql="selec t * from 表 where name=username"
Cmd.CommandText = sql
Set res = Cmd.Execute()
Set res = nothing
Set cmd.ActiveConnection = nothing
Set Cmd= nothing
```

自动化测试	Version:	1.3
文档		2006-5-17

以上四种方法都可以帮助我们实现数据驱动，应该说数据驱动在自动化测试中运用的比较的广泛，才有必要拿出来探讨一下

13. 在 Excel 中增加超链接的方式

```

Sub ReportInformation(filename)
    ' create the Excel object
    Set ExcelObj = CreateObject("Excel.Application")
    ' add a new Workbooks and a new Sheet
    ExcelObj.Workbooks.Add
    Set NewSheet = ExcelObj.Sheets.Item(1)
    NewSheet.Name = "Page Information"
    ' customize the Sheet layout
    NewSheet.Cells(1,1).Value = "Tom"
    NewSheet.Cells(2,1).Value = "Sohu"
    NewSheet.Hyperlinks.Add NewSheet.Cells(1,1), "http://www.tom.com/"
    NewSheet.Hyperlinks.Add NewSheet.Cells(2,1), "http://www.sohu.com/"
    ' save the Excel file
    ExcelObj.ActiveWorkbook.SaveAs filename
    ' close the application and clean the object
    ExcelObj.Quit
    Set ExcelObj = Nothing
End Sub

call ReportInformation("c:\test.xls")

```

自动化测试	Version:	1.3
文档		2006-5-17

14. 登陆脚本设计

```

Dim str,MyArray ,Msg

Browser("test").Page("test").Frame("topFrame").WebEdit("name").Set DataTable("Name", dtGlobalSheet)

Browser("test").Page("test").Frame("topFrame").WebEdit("name").Set DataTable("pwd", dtGlobalSheet)

Browser("test").Page("test").Frame("topFrame").Link("登录").Click

if(Browser("test").Dialog("Microsoft Internet Explorer").Exist) then

    str = Browser("test").Dialog("Microsoft Internet Explorer").GetVisibleText

    '假设系统提示信息为“请输入帐号！”，则“GetVisibleText”这个函数返回的值是：

    'Microsoft Internet Explorer

    '请输入帐号！

    '确定

    MyArray = Split(str, vbCrLf, -1, 1)

    'MyArray(0) =Microsoft Internet Explorer

    'MyArray(1) =请输入帐号！

    'MyArray(2) =确定

    str = MyArray(1)

    datatable.Value("result",dtGlobalSheet)=str

    Browser("test").Dialog("Microsoft Internet Explorer").WinButton("确定").Click

Else

    Browser("test").Page("test").Frame("topFrame_2").Link("退出").Click

end if

```

自动化测试	Version:	1.3
文档		2006-5-17

15. QTP 中 Wait 与同步点的区别

先说 wait 函数,当脚本走到 wait 函数时,就开始执行这个函数.如:wait(10),就等待 10 秒钟后再继续执行下面的语句.wait 函数的这个等待的时间,那相对来说是比较固定的.如上例子,一定要等待完 10 秒后再执行.所以写脚本的时候要自己估算一下时间.不然可能造成时间的浪费,或者等待时间的不足.

那同步点呢.等待时间就比较灵活了,它的等待时间是不固定的.设置同步点后,当脚本执行到这句话后,脚本就开始执行等待.脚本会在规定时间内不断的去检查,所同步的对象有没有出现,一但出现,脚本就继续往下执行.不需要等完所有规定时间.如果在规定的时间内,所要同步对象还没有出现,那就提示超时的错误信息.

例如:

```
Window("Flight Reservation").ActiveX("Threed Panel Control").WaitProperty "text", "Insert Done...", 10000
```

当脚本执行到这句话时,就开始执行同步等待时间.这里设置超时时间为 10000.在这个时间内,脚本会不断去查看该对象的 text 属性的属性值 Insert Done...,有没有出现.一但同步到这个属性值,就开始执行下面的脚本.而不需要再继续等待,直到 1000 秒结束为止.那这样的话,这个等待时间不用自己去控制,设置好后由程序自己去判断,就比较灵活,也不会出现浪费时间的情况.能提高脚本的执行率.

16. QTP tips

1. GetCellData 函数

作用:获取单元格的值

例: rowCount = Browser("xxx").Page("xxx").Frame("xxx").WebTable("xxx").RowCount

For counter = 1 To rowCount

```
text = Browser("xxx").Page("xxx").Frame("xxx").WebTable("xxx").GetCellData(counter,1)
```

If (text = "xxx") Then

counter = counter - 1

selectNO = "#" & counter

```
Browser("xxx").Page("xxx").Frame("xxx").WebRadioGroup("xxx").Select selectNO
```

Exit For

End If

Next

2. 把值插入 datatable 里

自动化测试	Version:	1.3
文档		2006-5-17

例: `datatable.setcurrentrow(i)`

```
datatable.value("name","Global")="name"
```

```
datatable.value("passwd","Global")="passwd"
```

3.用代码来启动浏览器

```
Browser1 = "IE"

StartURL = "www.51testing.com

IF Browser1 = "IE" THEN

    set IE = CreateObject("InternetExplorer.Application")

    IE.Visible = true

    IE.Navigate StartURL

END IF
```

4.ExecuteFile 函数

作用:ExecuteFile 可以直接执行 vbs 文件, 而不需要将其导入 resource 中

```
ExecuteFile FileName
```

说明:where FileName is the absolute or relative path of your VBscript file.

例:`ExecuteFile("F:\test.vbs")`

5.Strcomp 函数

作用:比较文本

例:dim strtext1,strtext2,str ,str1,comp1

```
strtext1 = "xxx"

strtext2 = "xxx"

str = VbWindow("xxx").VbWindow("xxx").VbLabel("xxx").GetTOPProperty("text")

str1= VbWindow("xxx").VbWindow("xxx").VbLabel("xxx").GetTOPProperty("text")
```

自动化测试	Version:	1.3
文档		2006-5-17

```

comp1=strcmp(strtext1,str,0)

If comp=0 Then

    msgbox "这两个串相等"

else

    msgbox str

End If

```

6.CaptureBitmap

作用:捕获屏幕

7. GetROProperty

作用:取对象属性值

例:VbWindow("xxx").VbWindow("xxx").VbWindow("xxx").ActiveX("xxx").GetROProperty("TextMatrix(1,0)")

8.ExitAction - 退出当前操作, 无论其循环属性如何。

ExitActionIteration - 退出操作的当前循环。

ExitRun - 退出测试, 无论其循环属性如何。

ExitGlobalIteration - 退出当前全局循环。

9. 如何使用 Excel 对象处理数据?

Dim xl

打开 excel 文件

Function OpenExcelFile(strFilePath)

Set xl = CreateObject("Excel.Application")

xl.Workbooks.Open strFilePath

End Function

获得指定单元格数据

Function GetCellData(strSheet,rwIndex,colIndex)

自动化测试		Page 44
-------	--	---------

自动化测试	Version:	1.3
文档		2006-5-17

```
GetData = xl.WorkSheets(strSheet).Cells(rwIndex,colIndex)
```

```
End Function
```

填充单元格数据

```
Function PutCellData(strSheet,rwIndex,colIndex,varData)
```

```
xl.WorkSheets(strSheet).Cells(rwIndex,colIndex) = varData
```

```
End Function
```

保存并推出

```
Function SaveAndQuit()
```

```
xl.ActiveWorkbook.Save
```

```
xl.Quit
```

```
Set xl = nothing
```

```
End Function
```

10.连接sql数据库

```
例:Dim res,cmd,sql
```

```
Set Res=createobject("adodb.recordset")
```

```
Set Cmd=createobject("adodb.command")
```

```
Cmd.activeconnection="Provider=SQLOLEDB.1;Password=111111;Persist Security Info=True;User ID=sa;Initial Catalog=xhq;Data
```

```
Source=192.168.191.142"    ' 这句话是连接数据库的数据源，要做修改
```

```
Cmd.CommandType = 1
```

```
sql="selec t * from 表 where name=username"
```

```
Cmd.CommandText = sql
```

```
Set res = Cmd.Execute()
```

```
Set res = nothing
```

```
Set cmd.ActiveConnection = nothing
```

```
Set Cmd= nothing
```

自动化测试	Version:	1.3
文档	2006-5-17	

17. Action 调用

本例还是以 **QTP** 安装后自带的 Flight Reservation 为例来录制脚本.首先我们把登录程序系统,然后新建定单,再打开定单,最后关闭系统.分别录制成 5 个 action,设置一个主的 action 为 main 来分别先后调用 login(登录),new order(新建定单),open order(打开定单),logout(退出).

注明:设置拆分 action, 选择菜单功能 split Action . 选择 independent of each other 为并列的两个 action. 选择 Nested 为主次的两个 action.

全部录制完所有 action 后, 在 main action 中代码如下:

```
RunAction "login", onelteration
RunAction "new order", onelteration
RunAction "open order", onelteration
RunAction "logout", onelteration
```

其中 **RunAction** 是一个调用 **action** 的函数, 后面跟 **action** 名和要传递的参数.

说明:

RunAction ActionName, [Iteration , Parameters]

在这里我再引入 action 之间参数传递. 在 login action 中选择菜单选项, 在 action properties 中设置两 input 参数分别先后为 username 和 passwd.

自动化测试	Version:	1.3
文档		2006-5-17

这样我们就通过 `RunAction "login", onelteration,"mercury","mercury"` 把这两个变量分别传给 `username` 和 `passwd`. 这边的变量先后关系分别依次对应了 `action properties` 里 `input` 参数的先后关系.

这样我们就可以在 `login action` 中调用这两个变量了.

```
Dialog("Login").WinEdit("Agent Name:").Set Parameter("username")
Dialog("Login").WinEdit("Password:").Set Parameter("passwd")
Dialog("Login").WinButton("OK").Click
```

下面再来实现从一个 `action` 中传出一个变量给另一个 `action`.

那么我们在 `new order` 中选择菜单选项，在 `action properties` 中设置一个 `output` 参数为 `orderno`

```
Window("Flight Reservation").WinEdit("Order No:").Output CheckPoint("Order No:")
```

通过上面这句话把生成的定单号的值存放到这个 `ouput` 参数中

```
RunAction "new order", onelteration,order
RunAction "open order", onelteration,order
```

再靠 `main action` 中这个 `order` 变量来传递，这个变量正好对应着 `new order` 里的 `output` 参数. 从 `new order` 中传出来，再把这个变量传给 `open order`. 当然在 `open order` 中还要同样设置 `input` 参数. 和这个 `order` 变量对应起来.

这样就可以在 `open order` 中使用这个变量了.

自动化测试	Version:	1.3
文档		2006-5-17

Window("Flight Reservation").Dialog("Open Order").WinEdit("Edit").Set Parameter("orderno")

当然其实两个 action 之间传递参数还可以更简单点。比如一个脚本中 datatable 和环境变量都是全局的，在一个脚本中的任何 action 都能使用。所以可以借用这两个来传递。

通过这个例子最主要讲的是两点，拆分 action 和 action 之间的参数传递。

18. Action 参数调用

Action 级别的参数调用：

1、设置参数

Action Properters >> Action Parameter

增加 input 的参数和 output 的参数

2、Action 1 的脚本：

```
parameter("output")="Action1 Output"
msgbox("Action1 Input:&parameter("input"))
msgbox("Action1 Output:&parameter("output"))
```

3、Action 2 的脚本：

```
RunAction "Action1", onelteration,"Action2 Input",action2
msgbox("Output in Action2:&action2")
```

我们不难发现在 QTP 的 Action 之间关系

作为被调用 Action1，都是使用 parameter("参数名") 来使用的。

作为调用 Action1 的 Action2 而言，input 参数是以往被调用 Action1 的操作输入的参数。Output 的参数是从被调用的 Action1 返回的值。

自动化测试	Version:	1.3
文档		2006-5-17

注：调试脚本

(http://blog.51testing.com/?3528/action_viewspace_itemid_1412.html)

19. QTP 中根据日期计算年龄的解决办法

在论坛上看到有关 **QTP** 中根据日期计算年龄的解决办法，所以写了一个 function 来处理。

涉及函数：

- 1、date：取当前系统时间；
- 2、isdate：判断输入的是否日期型的；
- 3、year：取得年的值。

具体的脚本：

```
Public Function old(MyDate)
If Isdate(MyDate) Then
    MyYear = Year(MyDate)
    CYear=Year(date)
    If (MyYear>CYear) Then
        MsgBox "出生年("&MyYear&")不可以大于当前年"
    Else
        Old=CYear-MyYear
    End If
Else
    MsgBox "错误的日期"
End If
End Function
ddate="October 19, 1962"
old1=old(ddate)
MsgBox("您的岁数是：" & old1)
```

20. 测试报告知多少

在运行过某一个 Test 后，会自动弹出一个 Test Results 页面，其含有信息是 Run-Time Data 以及每个步骤的执行结果。

其中结果的状态分为：Fail\Warning\Pass\Done。可以使用过滤器(Filters)进行过滤。

第二个实用的功能就是 Add Defect。

要使用 Add Defect，首先必须连接到 Quality Center。如果本身没有连接到 QC，这里还提供了一个小按钮可以弹出连接的画面，或者直接点击 Add Defect 按钮（是一个红色的带小叉号的图标），会先弹出连接 QC 的画面，连接上以后，点确定，就会出现你所熟悉的录制缺陷的画面了。

自动化测试	Version:	1.3
文档		2006-5-17

在**测试**结果中，对于 **Failed** 状态的信息往往不能肯定一定就是 **bug**（根据实际使用的经验所知）所以需要人工确认，再提交到系统中。如果可以肯定是 **bug** 的话，可以尝试写一个小程序，在 **Test** 运行的过程中就直接往 **QC** 中提报缺陷。

往往在执行 **自动化测试** 的时候，需要连续运行多个 **Test**，比如使用 **Multi Test Manager**。这时要查看 **Test Results** 有些麻烦，需要在 **Multi Test Manager** 的 **Results** 中，手动点开每个 **Test Result** 去查看。

因此 **QTP** 自带的 **Test Results** 比较适合在调试的时候使用，如果在实际的应用自动化测试中，个人推荐把重要的结果信息添加在代码中。

已经实践过的方法是将信息写入文本文件，**Test** 批运行后自动发送测试报告邮件给相关人员。具体的做法就不赘述了，如果大家感兴趣的话，我找时间再写。

最后再说一下怎么删除多余的 **Test Results**。

由于每运行一次 **Test**，就会生成一个 **Test Results** 文件，日积月累有好多都没有用而且占有硬盘空间（我做过的最多的一次，在一天内一个 **Test** 有 200 多个 **Results** 文件），这时候就要考虑删除没有用的结果信息。

这里提供两种删除 **Test Results** 的方法：

1，使用 **QTP** 自带的工具 **Test Results Deletion Tool**。打开界面，选择某个 **test** 或 **test** 所在的 **folder**，在 **Test Results** 栏中会出现所有的 **Results** 记录。可见即可删。

删除某一条记录：单选点 **delete** 键即可。

删除多条记录：按住 **Ctrl** 键选多条记录，再删除。

一次性删除所有记录：按 **Ctrl + A**，选中所有的记录，再删除。

2，手动删除

找到 **test** 所在的目录，看到有 **Res+阿拉伯数字** 组成的文件夹，里面存放的就是 **Test Results** 了，选中这些文件夹直接删除即可。

本人实践过，这种方法对**脚本**没有任何影响。只是注意一点，在删除的时候不要误删了**其他**文件。

21. 调用共享文件

在我们开发**脚本**的时候，发现有些脚本总是重复的出现在各个脚本中。那么我们是不是可以把这些脚本写成一个共享的 **function**，放到一个共享的文件中？然后再给各个脚本调用呢？答案当然是肯定中。其实我在《**QTP**的一些小知识》中也提到了一个方法，就是利用 **ExecuteFile** 函数来调用.vbs 文件。如：一个名为 **test.vbs** 的文件，我们只要在脚本中写上这么一句代码 **ExecuteFile("F:\test.vbs")**，就可以来直接执行这个.vbs 文件了。

自动化测试	Version:	1.3
文档		2006-5-17

现在我再来介绍另一种方法，这种方法也是 QTP 本身所提供的。就是设置菜单选项“setting”中的“resource”选项来做到调用共享文件的效果。在“resource”选项中，有三种文件格式是可以被调用的，分别是*.sql;*.vbs;*.txt 三种后缀文件。这里我用一个*.txt 文件为例。

新建一个名为 **test.txt** 的文本文件，其保存内容如下：（注：保存的内容也必须合法）

```
Function writeorderno(orderno)

Dim fso, myfile

Set fso=CreateObject("scripting.FileSystemObject")

Set myfile=fso.openTextFile("C:\result1.txt",8,false)

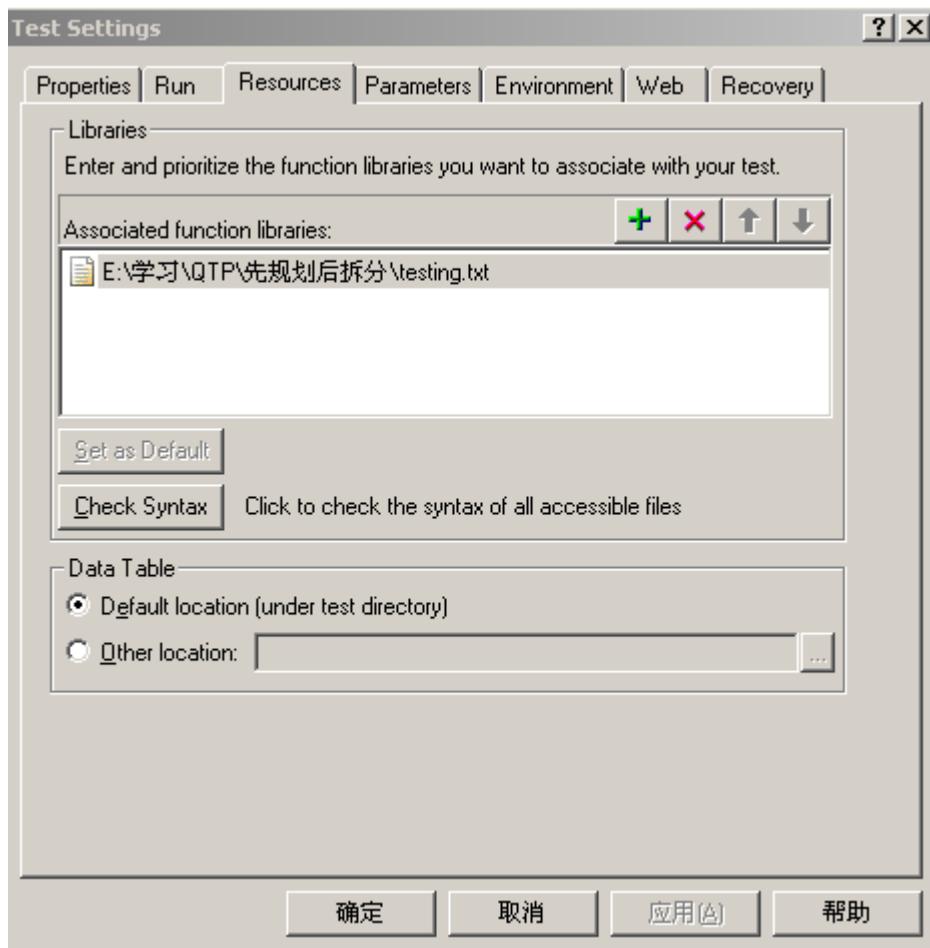
myfile.writeline orderno

myfile.close

End Function
```

一个名为 **writeorderno** 的 Function，其作用是把传进来的值写到 C 盘名为 **result1.txt** 的结果文件中去。（注：结果文件也是事先要准备好的）。共享文件准备完毕，那么就开始设置“resource”选项。打开“setting”窗口，找到“resource”选项。点击“+”的 button。选择前面我们建好的文件。如下：

自动化测试	Version:	1.3
文档		2006-5-17



确定关闭窗口。我们就来试验一下，是不是这个共享文件里的 Function 能够被调用了。在 QTP 专家视图里写上这么几句代码：

Dim order

order=1

writeorderno order

接下去就按 run 执行试验，结果 **通过**。虽然在脚本中根本没对 writeorderno 函数做说明，但是当 QTP 运行到这句代码的时候，在本脚本里没说明的情况下，自己会到“resource”选项里去找。结果 QTP 能正确识别这个 writeorderno 函数。打开 c 盘里的 result1.txt 的结果文件，发现传进函数的值 1，被正确的写进了去。预示着整个调用共享文件的成功。

自动化测试	Version:	1.3
文档		2006-5-17

22. 识别 grid 中的 button

对于 grid 中的按钮的识别，整理了两种方法，供参考。

1, 直接添加按钮对象到对象库

这种方法适合于可以直接回放成功的，这也是最简单的方法。

2, 如果对象识别不出来，或者回放的是否找不到对象，则使用如下方法。

```
aa = Browser("XXX").Page("XXX").Frame("XXX").WbfGrid ("XXX").ChildItemCount(2,3,"WebElement")  
  
set bb = Browser("XXX").Page("XXX").Frame("XXX").WbfGrid ("XXX").ChildItem (2,3,"WebElement",0)  
  
cc = typename (bb) 'ImageLink  
  
msgbox "子集数量: " & aa & "子对象类型: " & cc  
  
ee =bb.object.id 'ctl00_contentBody_gridView_ctl02_IEdit  
  
msgbox "按钮的 id: " & ee
```

本例中，通过找到的 id，可以使用字符串处理把最后的 Edit 找出来，就是要的“编辑”按钮啦。

23. How to parameterize your link

There is a better way to parameterize your link. We could utilize **Description object** since it has 5 method following provided:

Add Method

Count Property

Remove Method

RegularExpression Property

Value Property

自动化测试	Version:	1.3
文档		2006-5-17

Let's read info from a XML / Excel file first, and pass to Data Table.

And then, extract the value from Data Table, put it into Description object, then we could use it.

In this way, we could easily maintain our application configuration (like hostname or URL) when the environment changed.

Here is sample code

```
*****
```

For intLoop = 1 to N

```
strText=DataTable.Value(...)
```

```
Set LinkDesc = Description.Create()
```

```
LinkDesc ("Text").Value = strText
```

```
Browser("").Page("").Link(LinkDesc).Click
```

```
DataTable.GetSheet("").SetNextRow
```

Next

```
*****
```

自动化测试	Version:	1.3
文档		2006-5-17

24. QTP 中的步骤生成器

在我们录制脚本过程中,可能会遇到这样一种情况.用 object spy 能识别到某个对象,但录制的时候脚本就是没生成下来.当然如果你对 [QTP](#) 的原理比较熟悉,可以直接把对象加到对象库中,再自己手工写代码.这当然是个比较好的解决方法.但碰到对 QTP 不是很熟, 对 vbscript 的语法比较陌生的朋友来说, 可能需要花点功夫了. 但事实上 QTP 还提供给我们一个方便操作的工具, 叫做步骤生成器. 在 insert 菜单里能够找到, 当然按一下 F7 能直接调出这个工具.

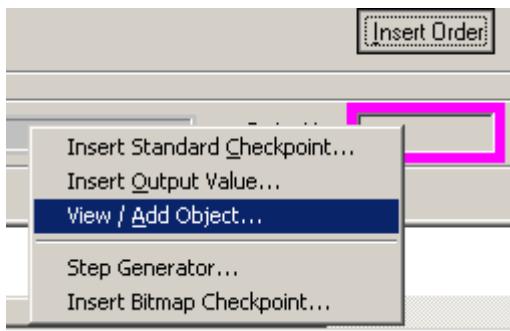
先来介绍一下这个工具.“步骤生成器”对话框有助于快速轻松地添加一些步骤, 这些步骤使用 [测试](#)对象方法、实用程序对象方法和函数调用, 因此用户无需记住语法或者精通高级 VBscript。既可在关键字视图中, 也可在专家视图中使用步骤生成器。

接下来用一个小例子来说明这个步骤生成器的使用.

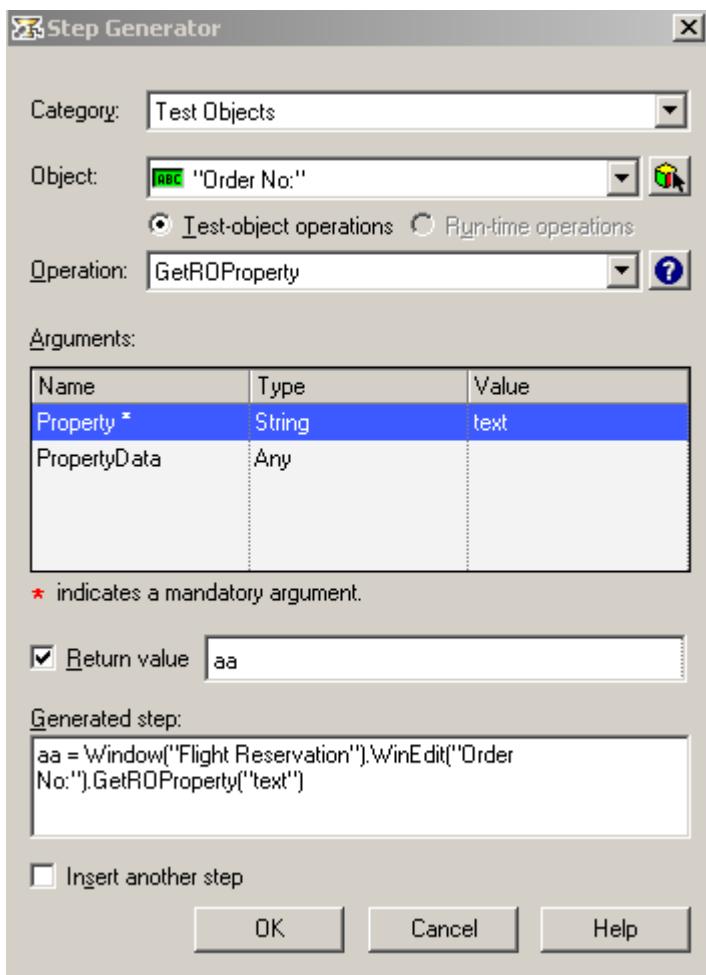
Flight No:	Departure Time:	Arrival Time:	Airline:
2028	08:12 AM	03:23 PM	AA
Name:		Tickets:	2
Class:		Price:	\$224.40
<input type="radio"/> First <input checked="" type="radio"/> Business <input type="radio"/> Economy		Total:	\$448.80
<input type="button" value="Update Order"/> <input type="button" value="Delete Order"/>		<input type="button" value="Insert Order"/>	
<input type="text"/>		Order No:	<input type="text"/>

比如在 QTP 自带的 flight 程序里, 在填好相应信息后, 按下 insert order 会生成一个定单号. 那么我们就用步骤生成器生成脚本来取这个 order no. 首先我们要做的是把这个对象加入到对象库中去. 加入对象的方法, 可以打开对象库来 add 填加. 也可以在 Active screen 里找到这个对象来右击鼠标, 选择 ADD object.如下:

自动化测试	Version: 1.3
文档	2006-5-17



在填出的窗口后，按下 **ok**,然后下一个窗口继续按 **add**，直到把对象真正加入到对象库中去。因为这边有些朋友以为第一次按了 **ok** 就已经填进去了，下面就直接 **close** 掉了，其实不是这样子的，这边是要注意的。对象我们已经准备好后，我们就在合适的地方生成需要的脚本，直接按一下 **F7**，然后打开了步骤生成器的窗口，如下：



自动化测试	Version:	1.3
文档		2006-5-17

在第一个下拉选项 **Category** 我们选择 **test object**. 在第二个下拉选项 **object** 里选择我们要操作的对象，我们这里选择 **order no.** 也可以按旁边的图标，会显示这个脚本中的所有对象让你选择。然后在第三个下拉选项 **operation** 里选择我们要做的操作。这里提供了很多的操作。这里只是取值，所以选择了 **GetRoProperty**.

然后在 **Arguments** 里在 **value** 里填上我们要取的属性名。在 **Return value** 打上勾，设置返回值取好变量名。再按下 **ok** 整句代码就出来了。

```
aa = Window("Flight Reservation").WinEdit("Order No:").GetROProperty("text")
```

到这里整个过程就结束了。大家可以自己试试看。

25. QTP 感想

对我来说，学习 QTP 是一个漫长而有艰苦的过程 。首先我不是 **计算机** 及相关专业毕业的（医学相关）。跳入 **测试** 部时，我正在接受程序员的培训课程。由于自己认为需要，于是开始学习 QTP。

刚开始使用 QTP，就一直对着说明书，不停的“订飞机票”（订飞机票是说明书里的一个例子）。学会了一个步骤就拿到公司产品上玩玩，回忆起来还是挺有趣的。

当我用一些简单的功能开始录制 **脚本** 时，发现保存 ActiveXScreen 的话，生成的脚本很占空间（因为程序会保存每个不同的录制页面），多录一些硬盘空间就满了，而且回放过程会很慢。  但如果不用保存活动页（ActiveXScreen），对脚本的再改造/维护起来就相对困难一些。

于是我开始去了解“关键字”视图里的内容，尝试了解代码。慢慢的，我了解到“关键字”视图显示了整个操作步骤，第 n 个组件相对于程序里一个元素。同时记录了录制过程对该元素的操作和结果。

然后我又开始在论坛找些资料看看，从有点所谓的高级应用中，我发现脚本的维护并不一定要有“活动页”。实际是 QTP 所有对象的识别，都存在脚本的一个对象库里了。QTP 经常出现无法识别对象的问题，可以从这里着手修改。 

为了减少 QTP 脚本占用空间大、录制慢的问题。我查阅了一些资料，可以在设置中进行修改，让脚本中不保存活动控件（ActiveX）或仅保存出错时的录制页面。干脆，我就从此录制页面了。所有的调试都从“关键字”视图和“专家”视图中进行修改。而且关于对象库，QTP 也有个选项，可以设置加载页面上所有的对象，我修改成只保存页面上录制过程使用的对象。  这样，脚本的容量问题就解决了，录制后的脚本会比以前小很多，来了个彻底的瘦身。关于录制速度的问题，和保存“活动页”、动态脚本也有一定的关系，另外可以减小启动的加载项（如：去掉 VB 插件、.net 插件，不需要的就不加进来）。这样的脚本上传到 TestDirector 上，或从 TestDirector 上调用就不会太慢了。

自动化测试	Version:	1.3
文档		2006-5-17

然而真正的问题，棘手的问题就不是上面所述的那么简单了。不过都是有办法解决了的，嘿嘿

以下是我经常遇到的问题：

- 一、无法识别控件。
- 二、错误回放过程未知弹出窗口。
- 三、加载.net 插件后和 TD 的关联问题。
- 四、动态加载元素的识别问题。
- 五、调用外部 dll 的问题。
- 六、随机验证码的问题。

问题一，解决办法有三种：

- 1、更改 QTP 自身对某控件的识别方式，在 **tools——Object Identification** 中。在这里列出了所有 QTP 能识别的控件，以及控件的识别方式。你可以给他添加 X、Y 坐标进行识别。或更明显的，列表中的信息，不按名称识别，而是按 ID 识别。这个修改可以解决一些问题，具体的赶紧动手试试吧
- 2、使用虚拟物件，来定义一个控件，在 **tools——Virtual Object** 中。在这里可以自定义一个控件。例如在 ASP 的程序中，程序出错，在客户端的表现形式大部分是一样的，你可以把整个错误页面当成一个控件来识别（感觉不错）。如果加一个判断，出错后你想做什么就由你自己定了。
- 3、使用低级录制或鼠标录制。用 **Test——LowLevelRecording/AnlogRecording** 吧，用它录制就不需要什么设置了，他会记录你的程序控件相对屏幕的位置。用 LowLevelRecording 还有代码可改，用 AnlogRecording 动作就被封装了（维护性极差）。两者因实际环境更取其长吧

问题二的解决过程：

关于弹出提示的问题，我当时需要情况是这样的。一个信息录入系统，由于数据量很大，查询需要一段时间。QTP 回放时动作比较快，点了保存，程序还没反应过来它就进行了下一步操作。这时的操作就和录制时不一样了，程序给出一个提示，但这个提示是录制过程没有的。弹出框是一般都是 POP 形势（至上）的，导致 QTP 无法继续回放，结果就是回放失败。

解决办法有两个：

- 1、进行判断，当出现这个提示时，点是/否/取消按钮。
- 2、通过 **Tools——Recovery Scenario Manager** 设置默认操作。

我最初就是用的第一种方法。写一个函数判断是否出现这个提示，如果出现就点“取消”然后 wait(2)。每个可能出现弹出框的动作后都调用一次这个函数。虽然可以解决这个问题，但回放的效率就低了，而且需要你预知提示框的信息。

当我知道了第二种方法，显然更科学^_^。它可以对所有预知甚至不知的提示进行指定的操作。

实际上，当程序出现了未预知的提示时，可能就是程序的 BUG，所以使用上述办法解决工具问题时，也要考虑是否会掩盖程序的缺陷。

自动化测试	Version:	1.3
文档		2006-5-17

问题三的解决办法:

用好 QTP 后，会不自觉的和 TD 关联起来。但从 TD 直接启动 QTP 时，程序只会加载 QTP 自带的插件，如果你安装了其它插件（如.net、java、etc.），默认是不加载的。这会导致上传的脚本无法正确执行。解决办法很简单，去 **Test——Setting** 里进行 **Modify** 吧。从本地打开的脚本，这里不能进行 **Modify** 的。所以办法很简单，但如果不知道的话就很难了。当初为这个问题我可是废了八牛三虎之力呢

问题四的解决过程:

当我开始改代码时，定义一个动作，然后可以生成 N 个动作。假设 N 个动作产生了 N 个结果，你要对这结果进行处理时，你会发现这 N 个结果都不能被识别：

网页上有个表格，是往[数据库](#)里加数据的。

两个表格显示在同一个页面上，左边为父表，右边为子表。

点击左表，右表显示其子项目。

结构如下：

A

|—1

|—2

|—3

|—4

B

|—1

|—2

|—3

|—4

.....

思想很清晰：

添加一个父项 A、[选中](#)此父项 A、对其添加子项 1、2、3、4

添加一个父项 B、[选中](#)此父项 B、对其添加子项 1、2、3、4

代码也很简单：

```
dim M      '定义父项数
```

自动化测试		Page 59
-------	--	---------

自动化测试	Version:	1.3
文档		2006-5-17

dim N '定义每个父项包含的子项数

For i=1 to M

Call 添加父项(i)

选中父项(i)

'问题就出在这里



For j=1 to bwfi step 1

Call 添加子项(j)

Next

Next

现在问题出来了，思路应该没有问题（除非这方法真的行不通），循环也是顺着思想来的。

问题是，无法实现选中的父项（最多识别到一个）。

由于此循环可以在录制过程进行，如果不改变变量名称，循环可且只可以成功运行一次。问题是这个名称都是从 DataTable 里获取的。

因为，在运行过程中生成的项目没有加到对象库中，无法被识别。

这个问题最后是从思想上解决的。答案是我做的是[功能测试](#)，为什么不先加父项，检查父项的功能是否正常，然后再去测子项的功能。不去改变名字，因为那没有必要。核心答案“[功能测试、测试功能](#)”。即对测试工具首先需要有正确的认识。

当然，这个问题可以用代码去实现，但那需要有一定的编程功底且耗时，可维护性不一定好。有需要的朋友可以去试一下，然后把你的经验也共享一下。
^_

问题五，是对 QTP 很大的一个扩充。



对于 QTP 调用外部 DLL 的功能，由于我的编程功底不够，没有相关人士配合我，我只能望之垂涎了！

如果能调用外部 DLL 的话，QTP 的功能就可以变得很强大。自己写的程序，自己编一些过程用 QTP 进行测试，我想“后果很严重”。真想有一次给我尝试的机会.....

问题六，解决办法有 4 个：

1、测试的时候，让程序员把这块限制去掉，免去验证这关。

2、让程序员提供一个万能验证码，测试可以绕过这一关。

3、请程序员提供识别的方法，从获取的[图片](#)读出验证数据，再传给 QTP。

自动化测试	Version:	1.3
文档		2006-5-17

4、进行位图检查，将验证码分段进行图像验证。

实际上，验证码的目的就是防止用程序灌水或机器录入信息。所以有点为难我们测试了。

方法 1，如果程序已在发布并有客户使用，危险性是可想而知的。方法 2 虽然可以解决验证这一关，但跳过了输入码与验证码一致性问题。方法 3 就需要程序员配合了，可能就需要调用 DLL 了。方法 4 却将图像分段，把获取的图像和已经的图像进行比对，比对通过取对应的值；这个在数字验证会好做一点，因为最多就四个图像的比对。

关于网上的汉字验证码，那块的测试我就不知道他们是怎么做的了。真想了解一下！



以上就是我对过去 QTP 学习过程的一个总结。供天下各界朋友参观、发言、讨论，也是对我过去的一个写照，可能 N 年后，自己看到会很有感觉呢。
😊。

现在又有项目来了，我 学习 的时间慢慢也少了。新项目里融合了 C++ 程序，QTP 对 C++ 的识别似乎很不理想，也许是需要插件支持吧。过程中我尝试了 Rational 的 Robot，Robot 对 C++ 的识别很好，但 Rational 一套组件内容太多，对汉字的支持似乎也不是很好。用了一段时间我就把它从硬盘中给清掉了……

QTP 的学习就在此停住了，不能应用到项目中，单纯看着说明书的学习，我好像不太在行。😊 也许是真的需要活学活用，边学边用吧。或者我不够书呆子吧……

26. QTP 参数化实例

在<[QTP 参数化实例](#)>一文中，讲到了运用描述性编程来进行参数化。今天我们利用另一种方法来实现类似的参数化。在不更改对象库的基础上，运用 SetTOProperty 函数方法来达到同样的效果。

例：

录制打开百度搜索主页，输入 1 1，点击搜索按钮，进入搜索后的页面，随便打开其中一个连接，然后关闭网页。录制 脚本如下：

```
Browser("百度——全球最大中文搜索引擎").Page("百度——全球最大中文搜索引擎").WebEdit("wd").Set "11"
Browser("百度——全球最大中文搜索引擎").Page("百度——全球最大中文搜索引擎").WebButton("百度搜索").Click
Browser("百度——全球最大中文搜索引擎").Page("百度搜索_11").Link("CCTV.com").Click
Browser("CCTV.com").Page("CCTV.com").Sync
Browser("CCTV.com").Close
```

自动化测试	Version:	1.3
文档		2006-5-17

Browser("百度——全球最大中文搜索引擎").Page("百度搜索_11").Sync

Browser("百度——全球最大中文搜索引擎").Close

现在我想做的是，能**测试**搜索页面上的其他 link 打开是否正确。那么势必要把上面蓝色字体的部分给参数化。

首先我们要在蓝色字体这句代码上面增加一句话，如下：

```
Browser("百度——全球最大中文搜索引擎").Page("百度搜索_11").Link("CCTV.com").SetTOProperty "属性","属性值"
```

```
Browser("百度——全球最大中文搜索引擎").Page("百度搜索_11").Link("CCTV.com").Click
```

SetTOProperty 方法能够在脚本运行中，设置对象的属性值。而 QTP 识别应用程序上的对象，完全是依靠对象库中的对象来识别的。所以只要对象库中的对象属性值变掉了，QTP 操作的对象也不同了。通过对该 **Link** 对象，在对象库中属性及属性值的研究发现，起决定性作用的是该对象 **text** 属性。所以只要更改这个 **text** 属性，就能让 QTP 点击其他 **Link** 了。

这里由于时间的关系，只做两个 link 的测试。继续修改脚本，变成如下：

```
Dim i,b
```

```
Browser("百度——全球最大中文搜索引擎").Page("百度——全球最大中文搜索引擎").WebEdit("wd").Set "11"
```

```
Browser("百度——全球最大中文搜索引擎").Page("百度——全球最大中文搜索引擎").WebButton("百度搜索").Click
```

```
For i=1 to 2
```

```
Select Case i
```

```
Case 1 b="11 弄-软件弄堂"
```

```
Case 2 b="CCTV.com"
```

```
End Select
```

```
Browser("百度——全球最大中文搜索引擎").Page("百度搜索_11").Link("CCTV.com").SetTOProperty "text",b
```

```
Browser("百度——全球最大中文搜索引擎").Page("百度搜索_11").Link("CCTV.com").Click
```

```
Browser("CCTV.com").Page("CCTV.com").Sync
```

```
Browser("CCTV.com").Close
```

```
Next
```

```
Browser("百度——全球最大中文搜索引擎").Page("百度搜索_11").Sync
```

```
Browser("百度——全球最大中文搜索引擎").Close
```

自动化测试	Version:	1.3
文档		2006-5-17

在我们脚本的对象库中是没有"11 弄-软件弄堂", 这个 link 对象的. 让我们看看这样修改后, 能不能通过两次循环, 依次点击打开名为 "11 弄-软件弄堂" 的连接, 然后关闭. 接着继续点击打开名为 "CCTV.com" 的连接, 然后关闭. 结果自然是通过的, 我们通过对 [SetTOProperty](#) 方法的运用, 同样达到了参数化的要求.

如果大家想试试看的话, 可以录制我们 QTP 自带的 windows 程序 fight. 其中这么一个 WinRadioButton 对象. 如下图:



对于这个对象的操作, 录出来的脚本是这样的.

```
Window("Flight Reservation").WinRadioButton("Business").Set
```

粗看这个对象要实现参数化, 不是很容易. 但其实也方便实现. 运用上面的方法, 试试吧!

27. 对象库的管理与合并

对象库的管理与合并

对于使用 [QTP](#) 做为 [自动化测试](#) 工具的测试项目来说, 其对象库的管理在整个自动化测试过程中, 占有非常重要的地位. 特别是对于一个大型的应用系统, 其界面的对象多而杂, 一个统一、有序的对象仓库非常有利于 [脚本](#) 的快速开发, 以及团队成员间的协作。

然而, 在实际使用中, 虽然可以改变对象库中对象节点添加时的位置 (如 Tools->Options->Web->Page/Frame Options), 但是在多人操作过程中, 难免会出现偏离原始设计思路的地方。这个就需要对象库管理员进行修改纠正和修改。

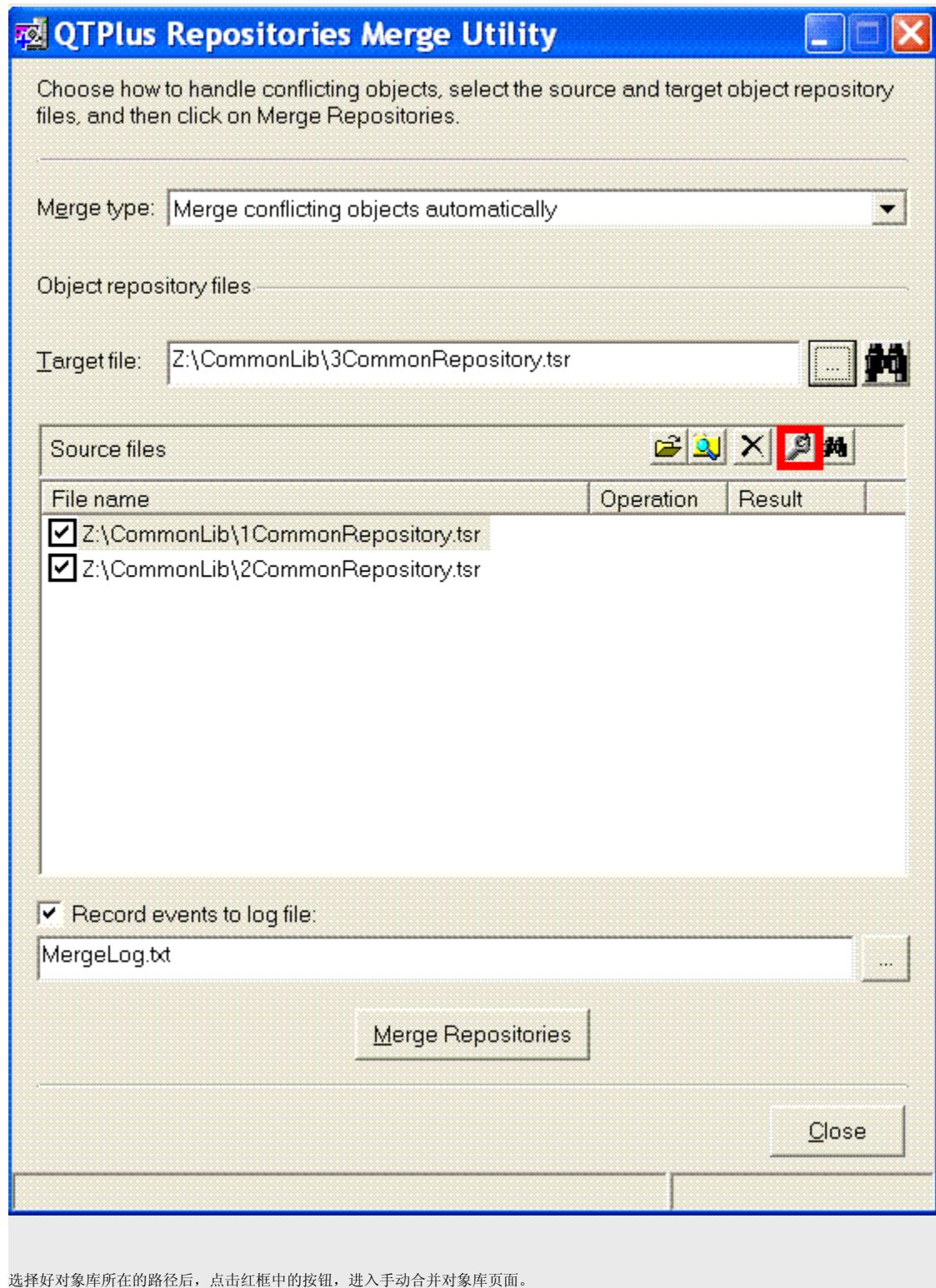
在 Object Repository 中, 可以修改对象节点的名称, 维护对象的属性, 保存对象库, 但并没有移动对象节点的功能。还好 QTP 提供了一个 Reporsitories Merge Utility, 可以实现对象库的合并功能。

下面我们就来尝试使用 Reporsitories Merge Utility 来改变对象节点的位置。

首先, 复制三个源对象库, 一个作为目标对象库, 两个作为源对象库。

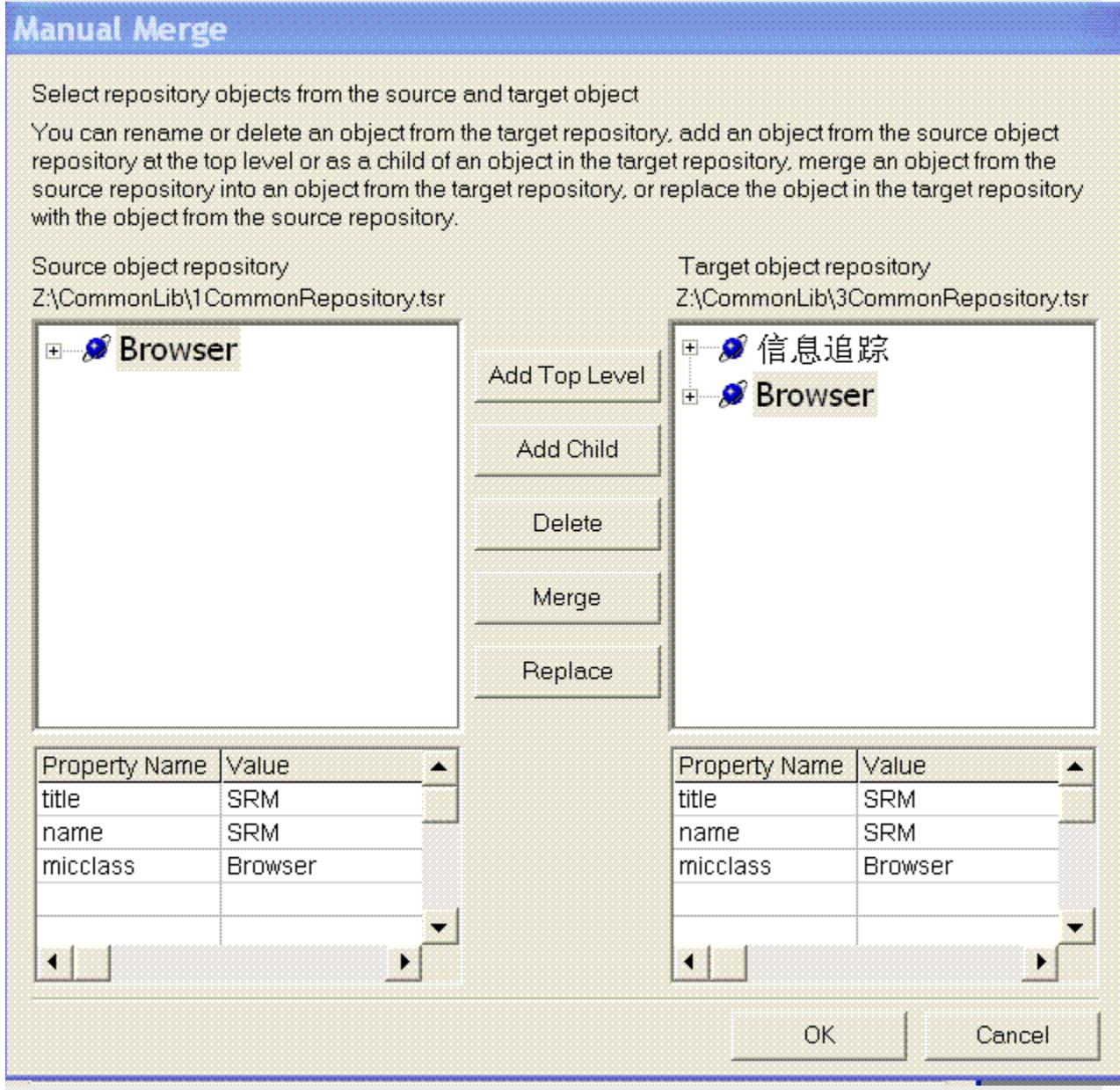
然后, 打开 Reporsitories Merge Utility, 如图

自动化测试	Version:	1.3
文档		2006-5-17



选择好对象库所在的路径后，点击红框中的按钮，进入手动合并对象库页面。

自动化测试	Version:	1.3
文档		2006-5-17



从左边的对象树上选择要移动的节点，目标的节点则默认是根节点。如果要移动到[其他的](#)节点下面，还需要在右边对象树上选择该节点才行。最后是点击按钮 Add top Level 或者 Add child，就将对象节点连同他的子节点一同添加到目标对象树上。也就是实现了我们的目标，把对象节点移位。

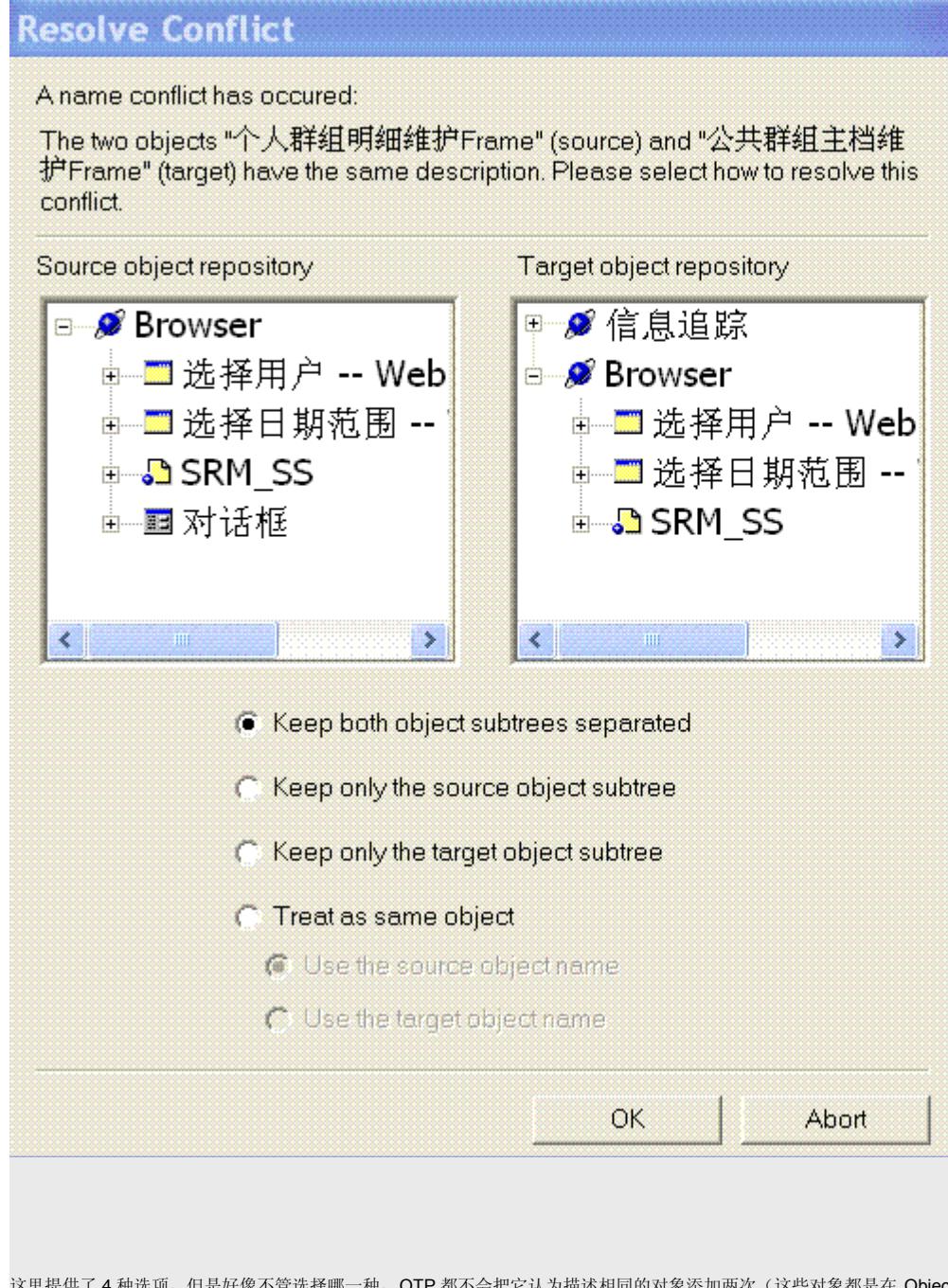
好了，检查一下你需要的所有的对象是否都已经成功添加。

注意一点，当遇到对象的 **description** 完全相同的情况，即使节点的名字不同，QTP 也会当作是相同的节点，不进行多次添加，即使在源对象树上是两个节点，添加到目标对象树上的仍然是一个节点。

自动化测试	Version: 1.3
文档	2006-5-17

对于两边都有的父节点，可以对其进行 Merge 操作，合并子节点。操作同增加节点，不同的是要选中左右两边名字相同的节点，这样 merge 按钮才会出来。

在合并的过程中，如果遇到对象的描述完全相同，会自动跳出 Resolve Conflict 页面，来帮助解决冲突。



这里提供了 4 种选项，但是好像不管选择哪一种，QTP 都不会把它认为描述相同的对象添加两次（这些对象都是在 Object Repository 中 Add 的）。

自动化测试	Version:	1.3
文档		2006-5-17

个人感觉，Repositories Merge Utility 针对于有多个节点相同的两个对象库，进行合并操作的时候不是很好用。我在做测试的时候，有两个节点没有自动添加进来，需要手动添加。而且出现了上述的对象描述相同但名字不同的情况，也是没有办法添加进来的。不过对于其他情况，还是可以做到添加新对象到统一的对象库中，把多人开发的结果集中到一起管理。

28. QTP 中虚拟对象设置

QTP 中虚拟对象设置

文章出处：51testing 博客 作者：xiaonan 发布时间：2007-01-25

当我们在录制脚本中，有时会遇到某个对象回放时不能识别。那么就可以考虑是不是可以做个虚拟对象来代替它。您可以将这些对象定义为虚拟对象，并将它们映射到标准类，例如按钮或复选框。QuickTest 在运行会话过程中模拟虚拟对象上的用户操作。在测试结果中，虚拟对象显示时类似标准类对象。

下面我们还是通过一个例子来具体操作一下。这次选用 LoadRunner 自带的 b/s 架构的 web 例子，来录制脚本。（注：运行这个例子，同时把 web server 先给开起来。）。这次只录制一个点击 sign up（注册）的步骤。

录出脚本如下：

```
Browser("Mercury Tours").Page("Mercury
Tours").Frame("navbar").Image("signup").Click 45,12
```



QTP 把这个 button 对象识别成了 Image（图片）对象，导致最后回放的时候，QTP 不能完成 Click 操作，弹出报错信息。对于这个对象，我马上想到了做个虚拟的 button 对象来代替它。

下面是整个虚拟对象的制作过程：

1. 选择“工具”>“虚拟对象”>“新建虚拟对象”。或者，也可以在虚拟对象管理器中，单击“新建”。将打开虚拟对象向导。单击“下一步”。



2. 选择要将您的虚拟对象映射到的标准类（这里有六种对象类供你选择，分别是 object, button, table, list, check box, radio button）。这里我们要映射一个 button 对象，所

自动化测试	Version:	1.3
文档		2006-5-17

以选择 button. 单击 "下一步 "



3 . 单击 “标记对象” 。选择你要做虚拟对象的区域，拖动鼠标。尽量能覆盖整个你想替代的对象。



4 . 在 “标识对象使用” 框中，选择您希望 QTP 标识和映射虚拟对象的方式。



如果您想要 QuickTest 标识所有出现的虚拟对象，请选择 “Parent only” 。

如果想要 QuickTest 仅标识一次出现的虚拟对象，请选择 “Entire parent hierarchy” 。

这里我们选择默认的 “Parent only” 。

5 . 指定虚拟对象的名称和集合。从集合列表中选择，或通过在 “集合名” 框中输入新名称来新建集合



点击完成，整个虚拟对象就做好了.

虚拟对象做好后，保存在虚拟对象管理器中。我们可以通过虚拟对象管理器来删除或禁用虚拟对象定义。

选择 “工具” > “虚拟对象” > “虚拟对象管理器” 。将打开虚拟对象管理器。

注意：如果希望 QuickTest 在录制时识别虚拟对象，请确保清除 “选项” 对话框的 “常规” 选项卡中的 “录制时禁止识别虚拟对象” 复选框。

接下来我们再来录制一遍对 sign up 的操作，录制代码如下：

```
Browser("Mercury Tours").Page("Mercury
Tours").Frame("navbar").VirtualButton("button").Click
```

在录制过程中值得注意的是，一定要点击到我们做好虚拟对象的范围内，这样 QTP 才能识别到这个虚拟对象。这里我们的虚拟对象识别成功。再次回放，QTP 顺利的执行了脚本。

还有两点要注意的是：

1 . 虚拟对象管理器中显示的虚拟对象集合存储在您的计算机中，而不是随包含虚拟对象步骤的测试或组件存储。这意味着如果您在测试或组件步骤中使用虚拟对象，则仅当在包含正确的虚拟对象定义的计算机中运行时，该对象在运行会话过程中才能被识别。要将您的虚拟对象集合定义复制到另一个计算机，请将您的 <QuickTest 安装文件夹>\dat\VoTemplate 文件夹的

自动化测试	Version:	1.3
文档		2006-5-17

内容（或该文件夹中的单个 .vot 集合文件）复制到目标计算机上的相同文件夹中。

2. 仅当录制和运行测试或组件时，才能使用虚拟对象。您不能在虚拟对象上插入任何类型的检查点，也不能使用“对象探测器”来查看其属性。

接下来你可以继续完成整个 sign up 过程，还有多个类似的对象需要做虚拟对象。你也可以试一下？

29. QTP 删除新建文本文件没有权限的问题

qtp 删除新建文本文件没有权限的问题

代码如下：

```

runfile("d:\a.txt")

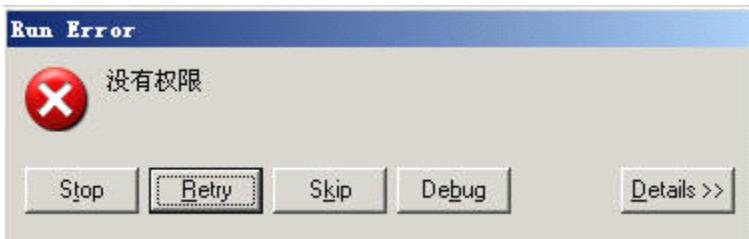
function runfile(url)
    Dim fso,f
    Dim flag
    Dim file1,f1
    set fso=createobject("scripting.filesystemobject")
    flag=fso.fileexists(url)//判断文件存不存在
    If flag=false Then
        reporter.ReportEvent 0,"文件不存在","succeed"
        Set file1=CreateObject("Scripting.FileSystemObject")
        Set f1=file1.CreateTextFile("d:\a.txt",true)
        reporter.ReportEvent 0,"文件创建成功","succeed"
        runfile("d:\a.txt")
    else
        set f=fso.getfile(url)
        fso.deletefile(url)//删除文件
        reporter.ReportEvent 0,"文件删除成功","succeed"
    End If
end function

```

运行这段代码时，文件存在的情况下没有问题，文件不存在新建没有问题，但是删除的时候会弹出没有权限的对话框，高手帮帮忙解决一下

 图片附件: **Snap1.jpg** (2007-2-10 15:53, 8.08 K)

自动化测试	Version:	1.3
文档		2006-5-17



30. 关于 Filter 如何使用的疑问

关于“Filter”如何使用的疑问。

[\[Copy to clipboard\]](#)

```

Dim TestCheck
Dim TestRun (5)

FindText = "SQLException"

TestRun(0) = "java.sql.SQLException: JZ0C0: Connection is already closed. "
TestRun(1) = "<html><head>"
TestRun(2) = "Filter command"
TestRun(3) = "Foxed, Stumped and stuck"
TestRun(4) = "<body>"
TestRun(5) = "Anyone?"

TestCheck = Filter(TestRun, FindText, true)

If UBound(TestCheck) >= 0 Then
    Reporter.ReportEvent 1, "页面状态", "页面" & TestCheck(0) & "中存在 " & FindText & " 无法访问."
else
    Reporter.ReportEvent 3, "页面状态", "正常, " & FindText & " 字符串."
End If

```

我们是不是可以用类似以上的代码，去判断页面是否现实正确呢？

我的页面保存的时候页面中会显示错误信息。

基本都是在第一行就显示了，是否可以取到 TestRun 的数组里面。然后一起判断？

你可以改一下.UBOUND

```

If UBound(TestCheck,1) Then
    Reporter.ReportEvent 1, "页面状态", "页面" & TestCheck(0) & "中存在 " & FindText & " 无法访问."
else
    Reporter.ReportEvent 3, "页面状态", "正常, " & FindText & " 字符串."
End If

```

31. Call Windows API

自动化测试	Version:	1.3
文档		2006-5-17

我自己写的 **dll** 无非就是做个加法,减法输出一下,呵呵,主要是我对 VC++ 不熟悉,拿不出手~ 😊
所以昨天实验了一些 VB 的 API 函数,用的都是系统中的 DLL 文件,比如:Kernel32.dll,user32.dll,GDI32.dll 等等,然后调用里面的一些函数实现一些功能,函数我没有多少,现在发一些我有的给大家分享一下吧~!

FindFirstFile

```
Declare Function FindFirstFile Lib "kernel32" Alias "FindFirstFileA" (ByVal lpFileName As String, lpFindFileData As
```

WIN32_FIND_DATA) As Long

说明:

根据文件名查找文件

返回值:

Long, 如执行成功, 返回一个搜索句柄。如果出错, 返回一个 INVALID_HANDLE_VALUE 常数, 一旦不再需要, 应该用 **FindClose** 函数关闭这个句柄

参数 类型及说明:

lpFileName String, 欲搜索的文件名。可包含通配符, 并可包含一个路径或相对路径名

lpFindFileData WIN32_FIND_DATA, 这个结构用于装载与找到的文件有关的信息。该结构可用于后续的搜索

注解

由这个函数返回的句柄可以作为一个参数用于 **FindNextFile** 函数。这样一来, 就可以方便的枚举出与 lpFileName 参数指定的文件名相符的所有文件

FindWindow

```
Declare Function FindWindow Lib "user32" Alias "FindWindowA" (ByVal lpClassName As String, ByVal lpWindowName
```

As String) As Long

说明:

寻找窗口列表中第一个符合指定条件的顶级窗口 (在 **vb** 里使用: **FindWindow** 最常见的一个用途是获得 ThunderRTMain 类的隐藏窗口的句柄; 该类是所有运行中 **vb** 执行程序的一部分。获得句柄后, 可用 api 函数 **GetWindowText** 取得这个窗口的名称; 该名也是应用程序的标题)

返回值:

Long, 找到窗口的句柄。如未找到相符窗口, 则返回零。会设置 **GetLastError**

参数 类型及说明:

lpClassName String, 指向包含了窗口类名的空中止 (C 语言) 字串的指针; 或设为零, 表示接收任何类

lpWindowName String, 指向包含了窗口文本 (或标签) 的空中止 (C 语言) 字串的指针; 或设为零, 表示接收任何窗口标题

注解

很少要求同时按类与窗口名搜索。为自己不准备参数传递一个零, 最简便的办法是传递 **vbNullString** 常数

SetActiveWindow

```
Declare Function SetActiveWindow Lib "user32" Alias "SetActiveWindow" (ByVal hwnd As Long) As Long
```

说明:

激活指定的窗口

返回值:

Long, 前一个活动窗口的句柄

参数 类型及说明:

hwnd Long, 待激活窗口的句柄

注解

在 **vb** 里使用这个函数要小心, 它不会改变输入焦点, 所以焦点可能设向一个不活动窗口, 最好换用 **SetFocusAPI** 函数来激活窗口。如指定的窗口不从属于当前输入线程, 则没有任何效果

自动化测试	Version:	1.3
文档		2006-5-17

SetWindowText

Declare Function SetWindowText Lib "user32" Alias "SetWindowTextA" (ByVal hwnd As Long, ByVal lpString As String)
As Long

说明:

设置窗口的标题文字或控件的内容 (在 vb 里使用: 针对 vb 窗体, 应使用 caption 或 text 属性)

返回值:

Long, 非零表示成功, 零表示失败。会设置 GetLastError

参数 类型及说明:

hwnd Long, 要设置文字的窗口的句柄

lpString String, 要设到 hwnd 窗口中的文字

ExitProcess

Declare Sub ExitProcess Lib "kernel32" Alias "ExitProcess" (ByVal uExitCode As Long)

说明:

中止一个进程

参数 类型及说明:

uExitCode Long, 指定想中断的那个进程的一个退出代码

GetProcessTimes

Declare Function GetProcessTimes Lib "kernel32" Alias "GetProcessTimes" (ByVal hProcess As Long, lpCreationTime
As FILETIME, lpExitTime As FILETIME, lpKernelTime As FILETIME, lpUserTime As FILETIME) As Long

说明:

获取与一个进程的经过时间有关的信息

返回值:

Long, 非零表示成功, 零表示失败。会设置 GetLastError

参数 类型及说明:

hProcess Long, 一个进程句柄

lpCreationTime FILETIME, 指定一个 FILETIME 结构, 在其中装载进程的创建时间

lpExitTime FILETIME, 指定一个 FILETIME 结构, 在其中装载进程的中止时间

lpKernelTime FILETIME, 指定一个 FILETIME 结构, 在其中装载进程花在内核模式上的总时间

lpUserTime FILETIME, 指定一个 FILETIME 结构, 在其中装载进程花在用户模式上的总时间

适用平台

Windows NT

LoadModule

Declare Function LoadModule Lib "kernel32" Alias "LoadModule" (ByVal lpModuleName As String,
lpParameterBlock As Any) As Long

说明:

载入一个 windows 应用程序, 并在指定的环境中运行

返回值:

Long, 大于 32 表示成功, 请参考 FindExecutable 函数的返回值

参数 类型及说明:

自动化测试	Version:	1.3
文档		2006-5-17

lpModuleName String, 要装载的可执行程序的文件名
 lpParameterBlock Any, 指定一个结构, 用它定义装载新应用程序时使用的参数

32. Global 和 local 区别

global 和 local 的区别

global 是全局的!有几行数据程序就要回放几次!!不能重新设置!!

local 是局部的!有几行数据 action 就要回访几次!!

设置 action 的重复次数的操作: 反击 action, 选择 action call property, 即可进行选择.

下面有几种情况:

1. 当 global 有不止一行的数据, action call property->Run On all Rows

程序每次运行时, action 中的每行都要执行

2. 当 global 有不止一行的数据, action call property->Run one iteration only

而且 global 的行数>action 的行数, 当 action 执行到最后一行后, 不管此时 global 的行数为几, 下次回放时 action 都执行最后一行!

如果 global 的行数<action 的行数, action 就执行不到最后一行

3. action call property->Run from rows to rows, 就结合上面所说的理解吧!!

希望能给大家帮上忙, 帮我顶一下!!第一次回答问题!

<http://bbs.51testing.com/thread-14818-1-1.html>

更正一下

global 的重复次数在 test->run 中设置

其实对 DataTable 的操作可以很灵活.....

由 QTP 自动生成的语句是这样的:

DataTable("Col", dtLocalSheet)

DataTable("Col", dtGlobalSheet)

其中, dtLocalSheet 就是当前 Action 的 DataTable, dtGlobalSheet 就是 Global 的 DataTable, Col 是列名

还有很多写法, 比如

DataTable("Col", "Action2")

就是读取 Action2 的 Col 列。

你完全可以读取另一个 Action 的 DataTable, 而不需要跨 Action 传递变量, 使用 DataTable 比变量更方便, 因为结束后能在 Result 里看到运行时的值

你可以在一个 Action 里读取另外一个 Action 的列, 但是要注意另外一个 Action 的当前行

比如你在 Action1 里读取 Action2 的某列, 如果 Action1 运行到第二行, 你读取的 Action2 还是第一行的数据, 解决办法就是写上这句:

DataTable.GetSheet("Action2").Set.CurrentRow(2)

你也可以用 GetCurrentRow 来获取 Action1 的行, 然后再用 Set.CurrentRow 来保持两个 Action 的当前行一致:

CurrRow = DataTable.GetSheet("Action1").GetCurrentRow

DataTable.GetSheet("Action2").Set.CurrentRow(CurrRow)

还有一种写法:

DataTable(1, "Action2")

这样就是读取 Action2 的第一列, 不管第一列叫什么名字, 都能读

这样就很方便, 比如:

For i = 1 To 10

 MsgBox DataTable(i, "Action2")

Next

自动化测试	Version:	1.3
文档		2006-5-17

这样就能循环读取 Action2 的 1~10 列了

同样，表也可以用数字代替：

DataTable(1, 2)

这样就是读取第二个 Action 表的第一列

33. QTP itself API

这个说法不够准确，应该说 TO 是仓库文件里定义的仓库对象，RO 是被测试软件的实际对象
 QTP 要求先在仓库文件里定义仓库对象，里面存有实际对象的特征属性的值，
 运行的时候，QTP 会根据仓库对象的特征属性描述，寻找到实际对象，然后操作实际对象。

仓库对象 TO 一般在录制/编写脚本时加入仓库文件，它不仅可以在编写时进行修改，
 也可以在运行过程中进行动态修改，以匹配实际对象。

相关的几个函数有：

GetTOProperty():取得仓库对象的某个属性的值

GetTOProperties():取得仓库对象的所有属性的值

SetTOProperty():设置仓库对象的某个属性的值

GetRProperty():取得实际对象的某个属性的值

理解了 TO 的含义，你就可以自由的用 SetTOProperty() 定义 TO，以灵活的操作 RO

比如有个测试任务，窗口上有很多待检查的记录，每条记录右边都有一个 Check 按钮，用来检查各条记录。
 记录个数不定，所以 Check 按钮个数也就不定，只有一个 Edit 显示记录个数。

我们要对每条记录进行检查，也就是要点击每个 Check 按钮。

但是 Check 按钮个数不定，不好录制，而且个数可能也很多（上百个），即使能一一录制，那也很麻烦。

那我有一个好办法，只录制一个按钮对象，它设有两个特征属性 label=OK, index=0
 然后用下面的脚本，就可以完成测试

```
buttonNum = CInt(JavaWindow("Test").JavaEdit("Record Num").GetRProperty("value"))
For buttonIndex = 0 to buttonNum - 1
    JavaWindow("Test").JavaButton("Check").SetTOProperty("index", buttonIndex)
    JavaWindow("Test").JavaButton("Check").Click
Next
```

或者窗口上有 New、Modify、Delete、Check 等好几个按钮，要把这几个按钮一一按过去
 我在对象仓库里只设置一个按钮对象 AnyButton，label 特征属性值填任意值，然后用下面脚本执行测试

```
JavaWindow("Test").JavaButton("AnyButton").SetTOProperty("label", "New")
JavaWindow("Test").JavaButton("AnyButton").Click
```

```
JavaWindow("Test").JavaButton("AnyButton").SetTOProperty("label", "Modify")
JavaWindow("Test").JavaButton("AnyButton").Click
```

自动化测试	Version:	1.3
文档		2006-5-17

```
JavaWindow("Test").JavaButton("AnyButton").SetTOProperty("label", "Delete")
JavaWindow("Test").JavaButton("AnyButton").Click
```

```
JavaWindow("Test").JavaButton("AnyButton").SetTOProperty("label", "Check")
JavaWindow("Test").JavaButton("AnyButton").Click
```

另外，QTP 还支持脚本描述的方法来定义和访问对象，即不需要在仓库里定义，也能访问和操作实际对象

如上面两个任务，可以如下实现

1. 不需要在仓库里定义 Check 按钮对象，直接用下面脚本来实现测试

```
buttonNum = CInt(JavaWindow("Test").JavaEdit("Record Num").GetROProperty("value"))
For buttonIndex = 0 to buttonNum - 1
    JavaWindow("Test").JavaButton("label:=Check", "index:=" + CStr(buttonIndex)).Click
Next
```

2. 不需要在仓库里定义 New、Modify、Delete、Check 按钮对象，直接用下面脚本来实现测试

```
JavaWindow("Test").JavaButton("label:=New").Click
JavaWindow("Test").JavaButton("label:=Modify").Click
JavaWindow("Test").JavaButton("label:=Delete").Click
JavaWindow("Test").JavaButton("label:=Check").Click
```

34. QTP 需探讨的问题

1, 异常处理和场景恢复????

2, installation issues

<http://bbs.51testing.com/thread-49840-1-1.html>

3, 单步运行,调试?

4, 描述性编程

在我的文章中也提到过了,描述性编程能让 QTP 脱离对象库来执行脚本.在某些特殊应用的时候,可能就更方便点.比如:要针对某个对象的参数化等等.而且描述性脚本因为脱离了对象库,所以增加了脚本的可移植性.同时脚本也更具有可读性.当然也给手工写脚本提供了一个平台.

5, 函数 和 过程

qtp 有自带的 function 和 sub 生成器。你可以利用生成器来生成函数或过程。

位置: 开始--程序--qtp---ols---FunctionDefGen.exe

1. executefile ("vb script name here")
2. add function in the "expert view" first and then use function call
3. test-->settings-->resources, add your vbs file

6, 对象识别

<http://bbs.51testing.com/thread-42574-1-1.html>

```
7, strLogin="7777.123"
num=FormatNumber(strLogin,2)
```

自动化测试	Version:	1.3
文档		2006-5-17

```
msgbox num
MyArray = Split(num, ",", -1, 1)
Msg= MyArray(0) & MyArray(1)
msgbox Msg
```

FormatNumber , InStr , right 函数.

```
strLogin="7777.123"
MyPos = Instr(1, strLogin, ".")
msgbox MyPos
LeftString = Left(strLogin, MyPos+2)
msgbox LeftString
```

```
strLogin="7777.123"
MyPos = Instr(1, strLogin, ".")
strHead=left(strLogin,MyPos -1)
strTail=mid(strLogin,MyPos+1,2)
strLogin=strHead & strTail
```

8. 移植

如果没有用到低级录制和虚拟对象，一般不用修改就可以运行
假如是共享对象库的话，需要注意路径的变换。

9 URL 参数化

设想一下在 VBS 中,你要把一个字符串中的某一个单词或者某一段参数化,你应该怎么做呢?所以这是一个 VBS 编程的问题,如果你不是很清楚,那就看看 VBS 的语法吧.在 LR 中右键是可以提供参数化功能,但 QTP 没有这个功能的.
还有就是你想参数化的应该是 liujia 和 123456 吧

```
userStr = liujia
pawStr = 123456
参数化后就是
"http://192.168.0.10/wusercheck?user=" & userStr & "&paw=" & pawStr
```

35. 正则表达式的应用

正则表达式

转一篇正则表达式的文章。

鉴于正则表达式的重要性。特转此文。

正则表达式使用详解（一）

正则表达式使用详解（一）

2000-04-26 吕晓波

如果我们问那些 UNIX 系统的爱好者他们最喜欢什么，答案除了稳定的系统和可以远程启动之外，十有八九的人会提到正则表达式；如果我们再问他们最头痛的是什么，可能除了复杂的进程控制和安装过程之外，还会是正则表达式。

自动化测试	Version:	1.3
文档		2006-5-17

式。那么正则表达式到底是什么？如何才能真正的掌握正则表达式并正确的加以灵活运用？本文将就此展开介绍，希望能够对那些渴望了解和掌握正则表达式的读者有所助益。

入门简介

简单的说，正则表达式是一种可以用于模式匹配和替换的强有力的工具。我们可以在几乎所有的基于 **UNIX** 系统的工具中找到正则表达式的身影，例如，**vi** 编辑器，**Perl** 或 **PHP** 脚本语言，以及 **awk** 或 **sed shell** 程序等。此外，象 **JavaScript** 这种客户端的脚本语言也提供了对正则表达式的支持。由此可见，正则表达式已经超出了某种语言或某个系统的局限，成为人们广为接受的概念和功能。

正则表达式可以让用户通过使用一系列的特殊字符构建匹配模式，然后把匹配模式与数据文件、程序输入以及 **WEB** 页面的表单输入等目标对象进行比较，根据比较对象中是否包含匹配模式，执行相应的程序。

举例来说，正则表达式的一个最为普遍的应用就是用于验证用户在线输入的邮件地址的格式是否正确。如果通过正则表达式验证用户邮件地址的格式正确，用户所填写的表单信息将会被正常处理；反之，如果用户输入的邮件地址与正则表达的模式不匹配，将会弹出提示信息，要求用户重新输入正确的邮件地址。由此可见正则表达式在 **WEB** 应用的逻辑判断中具有举足轻重的作用。

基本语法

在对正则表达式的功能和作用有了初步的了解之后，我们就来具体看一下正则表达式的语法格式。

正则表达式的形式一般如下：

/love/

其中位于“/”定界符之间的部分就是将要在目标对象中进行匹配的模式。用户只要把希望查找匹配对象的模式内容放入“/”定界符之间即可。为了能够使用户更加灵活的定制模式内容，正则表达式提供了专门的“元字符”。所谓元字符就是指那些在正则表达式中具有特殊意义的专用字符，可以用来规定其前导字符（即位于元字符前面的字符）在目标对象中的出现模式。

自动化测试	Version:	1.3
文档		2006-5-17

较为常用的元字符包括：“+”，“*”，以及“?”。其中，“+”元字符规定其前导字符必须在目标对象中连续出现一次或多次，“*”元字符规定其前导字符必须在目标对象中出现零次或连续多次，而“?”元字符规定其前导对象必须在目标对象中连续出现零次或一次。

下面，就让我们来看一下正则表达式元字符的具体应用。

/fo+/
因为上述正则表达式中包含“+”元字符，表示可以与目标对象中的“fool”，“fo”，或者“football”等在字母 f 后面连续出现一个或多个字母 o 的字符串相匹配。

/eg*/
因为上述正则表达式中包含“*”元字符，表示可以与目标对象中的“easy”，“ego”，或者“egg”等在字母 e 后面连续出现零个或多个字母 g 的字符串相匹配。

/Wil?/
因为上述正则表达式中包含“?”元字符，表示可以与目标对象中的“Win”，或者“Wilson”，等在字母 i 后面连续出现零个或一个字母 l 的字符串相匹配。

除了元字符之外，用户还可以精确指定模式在匹配对象中出现的频率。例如，

/jim{2,6}/
上述正则表达式规定字符 m 可以在匹配对象中连续出现 2-6 次，因此，上述正则表达式可以同 jimmy 或 jimmmy 等字符串相匹配。

自动化测试	Version:	1.3
文档		2006-5-17

在对如何使用正则表达式有了初步了解之后，我们来看一下其它几个重要的元字符的使用方式。

\s: 用于匹配单个空格符，包括 tab 键和换行符；

\S: 用于匹配除单个空格符之外的所有字符；

\d: 用于匹配从 0 到 9 的数字；

\w: 用于匹配字母，数字或下划线字符；

\W: 用于匹配所有与\w 不匹配的字符；

. : 用于匹配除换行符之外的所有字符。

(说明：我们可以把\s 和\S 以及\w 和\W 看作互为逆运算)

下面，我们就通过实例看一下如何在正则表达式中使用上述元字符。

/\s+/
上述正则表达式可以用于匹配目标对象中的一个或多个空格字符。

/\d000/
如果我们手中有一份复杂的财务报表，那么我们可以通过上述正则表达式轻而易举的查找到所有总额达千元的款项。

自动化测试	Version:	1.3
文档		2006-5-17

除了我们以上所介绍的元字符之外，正则表达式中还具有另外一种较为独特的专用字符，即定位符。定位符用于规定匹配模式在目标对象中的出现位置。

较为常用的定位符包括：“^”，“\$”，“\b”以及“\B”。其中，“^”定位符规定匹配模式必须出现在目标字符串的开头，“\$”定位符规定匹配模式必须出现在目标对象的结尾，“\b”定位符规定匹配模式必须出现在目标字符串的开头或结尾的两个边界之一，而“\B”定位符则规定匹配对象必须位于目标字符串的开头和结尾两个边界之内，即匹配对象既不能作为目标字符串的开头，也不能作为目标字符串的结尾。同样，我们也可以把“^”和“\$”以及“\b”和“\B”看作是互为逆运算的两组定位符。举例来说：

/^hell/

因为上述正则表达式中包含“^”定位符，所以可以与目标对象中以“hell”，“hello”或“hellhound”开头的字符串相匹配。

/ar\$/

因为上述正则表达式中包含“\$”定位符，所以可以与目标对象中以“car”，“bar”或“ar”结尾的字符串相匹配。

/\bboom/

因为上述正则表达式模式以“\b”定位符开头，所以可以与目标对象中以“bomb”，或“bom”开头的字符串相匹配。

/man\b/

因为上述正则表达式模式以“\b”定位符结尾，所以可以与目标对象中以“human”，“woman”或“man”结尾的字符串相匹配。

为了能够方便用户更加灵活的设定匹配模式，正则表达式允许使用者在匹配模式中指定某一个范围而不局限于具体的字符。例如：

自动化测试	Version:	1.3
文档		2006-5-17

/[A-Z]/

上述正则表达式将会与从 A 到 Z 范围内任何一个大写字母相匹配。

/[a-z]/

上述正则表达式将会与从 a 到 z 范围内任何一个小写字母相匹配。

/[0-9]/

上述正则表达式将会与从 0 到 9 范围内任何一个数字相匹配。

/([a-z][A-Z][0-9])+/

上述正则表达式将会与任何由字母和数字组成的字符串，如 “aB0” 等相匹配。这里需要提醒用户注意的一点就是可以在正则表达式中使用 “()” 把字符串组合在一起。“()” 符号包含的内容必须同时出现在目标对象中。因此，上述正则表达式将无法与诸如 “abc” 等的字符串匹配，因为 “abc” 中的最后一个字符为字母而非数字。

如果我们希望在正则表达式中实现类似编程逻辑中的“或”运算，在多个不同的模式中任选一个进行匹配的话，可以使用管道符 “|”。例如：

/to|too|2/

上述正则表达式将会与目标对象中的 “to”，“too”，或 “2” 相匹配。

正则表达式中还有一个较为常用的运算符，即否定符 “[^]”。与我们前文所介绍的定位符 “^” 不同，否定符 “[^]” 规定目标对象中不能存在模式中所规定的字符串。例如：

自动化测试	Version:	1.3
文档		2006-5-17

/[^A-C]/

上述字符串将会与目标对象中除 A, B, 和 C 之外的任何字符相匹配。一般来说，当“^”出现在 “[]”内时就被视做否定运算符；而当“^”位于“[]”之外，或没有“[]”时，则应当被视做定位符。

最后，当用户需要在正则表达式的模式中加入元字符，并查找其匹配对象时，可以使用转义符“\”。例如：

/Th*/

上述正则表达式将会与目标对象中的“Th*”而非“The”等相匹配。

36. 数据驱动测试：EXCEL

11 excel

本主题由 **walker1020** 于 **2006-9-6 21:58** 移动

自动化测试	Version:	1.3
文档		2006-5-17

jul #1
iet 发表于 2006-8-31 23:18 资料 短消息

74 QuickTest Plus 帮助文档中 EXCEL 相关函数

38

中级站友

中级站友

UI D

42

03 精华

2

积分

26

帖子

8

5

综合技术

指数

26

生活情趣

指教

0

鲜花

0

鸡

?

Dim ExcelApp 'As Excel.Application
Dim excelSheet 'As Excel.worksheet
Dim excelBook 'As Excel.workbook
Dim fso 'As Scripting.FileSystemObject

,

' 函数说明: 创建一个 Excel 应用程序 ExcelApp, 并创建一个新的工作簿 Workbook;
' 参数说明: 无
' 调用方法:
' CreateExcel()
,

Function CreateExcel()
 Dim excelSheet
 Set ExcelApp = CreateObject("Excel.Application")
 ExcelApp.Workbooks.Add
 ExcelApp.Visible = True
 Set CreateExcel = ExcelApp
End Function

,

' 函数说明: 关闭 Excel 应用程序;
' 参数说明:
' (1)ExcelApp:Excel 应用程序名称;
' 调用方法:
' CloseExcel(ExcelApp)

Sub CloseExcel(ExcelApp)
 Set excelSheet = ExcelApp.ActiveSheet
 Set excelBook = ExcelApp.ActiveWorkbook
 Set fso = CreateObject("Scripting.FileSystemObject")
 On Error Resume Next
 fso.CreateFolder "C:\Temp"
 fso.DeleteFile "C:\Temp\ExcelExamples.xls"
 excelBook.SaveAs "C:\Temp\ExcelExamples.xls"
 ExcelApp.Quit
 Set ExcelApp = Nothing
 Set fso = Nothing

自动化测试	Version:	1.3
文档		2006-5-17

蛋
0 Err = 0
On Error GoTo 0
End Sub
阅读权限
1 *****
' 函数说明: 保存工作薄;
0 注册
2 ' 参数说明:
0 ' (1)ExcelApp:Excel 应用程序名称;
4- ' (2)workbookIdentifier:属于 ExcelApp 的工作簿名称;
10 ' (3)path:保存的路径;
2 ' 返回结果:
00 ' (1)保存成功, 返回字符串: OK
4- ' (2)保存失败, 返回字符串: Bad Worksheet Identifier
10 ' 调用方法:
-8 ' ret = SaveWorkbook(ExcelApp, "Book1", "D:\Example1.xls")

离线 附件: [LibExcel.vbs] **LibExcel.vbs** (2006-8-31 23:18, 13.76 K)
该附件被下载次数 86

 (0)  (0) 引用 报告 回复 

[广告] 论坛发贴就是写博客, 51Testing 博客全新登场, 论坛博客同一帐号, 咱的空间咱做主, 拥有自己的个性空间: blog.51testing.com

自动化测试	Version:	1.3
文档		2006-5-17

jul #2

iet 发表于 2006-9-1 00:15 资料 短消息

74

38

中级
站友



UI
D
42

03
精华
2

积分
2

26
帖子
8

5
综合
技术
指
数

2
26

生活
情趣
指
数

0
鲜花
0

鸡

```

*****
' 函数说明: 保存工作薄;
' 参数说明:
'   (1)ExcelApp:Excel 应用程序名称;
'   (2)workbookIdentifier:属于 ExcelApp 的工作薄名称;
'   (3)path:保存的路径;
' 返回结果:
'   (1)保存成功, 返回字符串: OK
'   (2)保存失败, 返回字符串: Bad Worksheet Identifier
' 调用方法:
'   ret = SaveWorkbook(ExcelApp, "Book1", "D:\Example1.xls")
*****
```

```

Function SaveWorkbook(ExcelApp, workbookIdentifier, path) 'As String
    Dim workbook
    On Error Resume Next '启用错误处理程序
    Set workbook = ExcelApp.Workbooks(workbookIdentifier)
    On Error GoTo 0 '禁用错误处理程序

    If Not workbook Is Nothing Then
        If path = "" Or path = workbook.FullName Or path = workbook.Name Then
            workbook.Save
        Else
            Set fso = CreateObject("Scripting.FileSystemObject")

            '判断路径中是否已添加扩展名.xls
            If InStr(path, ".") = 0 Then
                path = path & ".xls"
            End If

            '删除路径下现有同名的文件
            On Error Resume Next
            fso.DeleteFile path
            Set fso = Nothing
            Err = 0
            On Error GoTo 0

            workbook.SaveAs path
        End If
        SaveWorkbook = "OK"
    Else
        SaveWorkbook = "Bad Workbook Identifier"
    End If
End Function
```

```

*****
```

自动化测试	Version:	1.3
文档		2006-5-17

```

蛋
0   *
0   | 函数说明：设置工作表 excelSheet 单元格的值
阅读    | 参数说明：
权限    |   (1)excelSheet:工作表名称;
        |   (2)row:列的序号,第一列为 1;
        |   (3)column:行的序号,第一行为 1;
        |   (4)value:单元格要设置的值;
1   | 返回结果:
0   |   无返回值
注冊    | 调用方法:
2   |     SetCellValue excelSheet1, 1, 2, "test"
00   ****
4-
10  Sub SetCellValue(excelSheet, row, column, value)
-8    On Error Resume Next
状态      excelSheet.Cells(row, column) = value
        On Error GoTo 0
End Sub

离线    'The GetCellValue returns the cell's value according to its row column and sheet
'excelSheet - the Excel Sheet in which the cell exists
'row - the cell's row
'column - the cell's column
'return 0 if the cell could not be found
'

****

' 函数说明：获取工作表 excelSheet 单元格的值
参数说明：
(1)excelSheet:工作表名称;
(2)row:列的序号;
(3)column:行的序号;
返回结果：
(1)单元格存在,返回单元格值;
(2)单元格不存在，返回 0;
调用方法：
  set CellValue = GetCellValue(excelSheet, 1, 2)

****

Function GetCellValue(excelSheet, row, column)
  value = 0
  Err = 0
  On Error Resume Next
  tempValue = excelSheet.Cells(row, column)
  If Err = 0 Then
    value = tempValue
    Err = 0
  End If
  On Error GoTo 0
  GetCellValue = value

```

自动化测试	Version:	1.3
文档		2006-5-17

End Function

```
' *****
' 函数说明: 获取并返回工作表对象
' 参数说明:
'   (1)ExcelApp:Excel 应用程序名称;
'   (2)sheetIdentifier:属于 ExcelApp 的工作表名称;
' 返回结果:
'   (1)成功: 工作表对象 Excel.worksheet
'   (1)失败: Nothing
' 调用方法:
'   Set excelSheet1 = GetSheet(ExcelApp, "Sheet Name")
' *****
```

```
Function GetSheet(ExcelApp, sheetIdentifier)
On Error Resume Next
Set GetSheet = ExcelApp.Worksheets.Item(sheetIdentifier)
On Error GoTo 0
End Function
```

 (0)  (0) 引用 报告 回复 

[广告] 论坛发贴就是写博客, 51Testing 博客全新登场, 论坛博客同一帐号, 咱的空间咱做主, 拥有自己的个性空间: blog.51testing.com

自动化测试	Version:	1.3
文档		2006-5-17

jul
iet
74
38

#3

发表于 2006-9-1 00:15 资料 短消息

中级站友



UI
D
42

03
精华
2

积分
2

26
帖子
8

5
综合技术指
数
2

26

生活情趣指
数
0

鲜花
0

鸡

```
*****
' 函数说明: 添加一张新的工作表
' 参数说明:
'   (1)ExcelApp:Excel 应用程序名称;
'   (2)workbookIdentifier:属于 ExcelApp 的工作薄名称;
'   (2)sheetName:要插入的工作表名称;
' 返回结果:
'   (1)成功: 工作表对象 worksheet
'   (1)失败: Nothing
' 调用方法:
'   InsertNewWorksheet(ExcelApp, workbookIdentifier, "new sheet")
*****
Function InsertNewWorksheet(ExcelApp, workbookIdentifier, sheetName)
    Dim workbook 'As Excel.workbook
    Dim worksheet 'As Excel.worksheet

    '如果指定的工作薄不存在, 将在当前激活状态的工作表中添加工作表
    If workbookIdentifier = "" Then
        Set workbook = ExcelApp.ActiveWorkbook
    Else
        On Error Resume Next
        Err = 0
        Set workbook = ExcelApp.Workbooks(workbookIdentifier)
        If Err <> 0 Then
            Set InsertNewWorksheet = Nothing
            Err = 0
            Exit Function
        End If
        On Error GoTo 0
    End If

    sheetCount = workbook.Sheets.Count '获取工作薄中工作表的数量
    workbook.Sheets.Add , sheetCount '添加工作表
    Set worksheet = workbook.Sheets(sheetCount + 1) '初始化 worksheet 为新添加的工作表对象

    '设置新添加的工作表名称
    If sheetName <> "" Then
        worksheet.Name = sheetName
    End If

    Set InsertNewWorksheet = worksheet
End Function
*****
```

自动化测试	Version:	1.3
文档		2006-5-17

蛋
0 | 函数说明：修改工作表的名称；
 阅读权限
1 | 参数说明：
 0 | (1)ExcelApp:Excel 应用程序名称；
 注册
2 | (2)workbookIdentifier:属于 ExcelApp 的工作薄名称；
 离线
00 | (3)worksheetIdentifier:属于 workbookIdentifier 工作薄的工作表名称；
 4- | (4)sheetName:修改后的工作表名称；
 10 | 返回结果：
 -8 | (1)修改成功，返回字符串：OK
 状态
2 | (2)修改失败，返回字符串：Bad Worksheet Identifier
 调用方法：
 00 | set ret = RenameWorksheet(ExcelApp, "Book1", "Sheet1", "Sheet Name")

 10 | Function RenameWorksheet(ExcelApp, workbookIdentifier, worksheetIdentifier, sheetName)
 -8 | Dim workbook
 状态
2 | Dim worksheet
 离线
00 | On Error Resume Next
 4- | Err = 0
 10 | Set workbook = ExcelApp.Workbooks(workbookIdentifier)
 -8 | If Err <> 0 Then
 状态
2 | RenameWorksheet = "Bad Workbook Identifier"
 离线
00 | Err = 0
 4- | Exit Function
 10 | End If
 2 | Set worksheet = workbook.Sheets(worksheetIdentifier)
 00 | If Err <> 0 Then
 4- | RenameWorksheet = "Bad Worksheet Identifier"
 2 | Err = 0
 00 | Exit Function
 4- | End If
 2 | worksheet.Name = sheetName
 00 | RenameWorksheet = "OK"
 4- | End Function
 2 |
 00 | ****
 10 | | 函数说明：删除工作表；
 -8 | | 参数说明：
 状态
2 | | (1)ExcelApp:Excel 应用程序名称；
 离线
00 | | (2)workbookIdentifier:属于 ExcelApp 的工作薄名称；
 4- | | (3)worksheetIdentifier:属于 workbookIdentifier 工作薄的工作表名称；
 10 | | 返回结果：
 -8 | | (1)删除成功，返回字符串：OK
 状态
2 | | (2)删除失败，返回字符串：Bad Worksheet Identifier
 离线
00 | | 调用方法：
 4- | | set ret = RemoveWorksheet(ExcelApp, "Book1", "Sheet1")
 2 |
 00 | ****
 10 | Function RemoveWorksheet(ExcelApp, workbookIdentifier, worksheetIdentifier)

自动化测试		Page 89
-------	--	---------

自动化测试	Version:	1.3
文档		2006-5-17

```

Dim workbook 'As Excel.workbook
Dim worksheet 'As Excel.worksheet
On Error Resume Next
Err = 0
Set workbook = ExcelApp.Workbooks(workbookIdentifier)
If Err <> 0 Then
    RemoveWorksheet = "Bad Workbook Identifier"
    Exit Function
End If
Set worksheet = workbook.Sheets(worksheetIdentifier)
If Err <> 0 Then
    RemoveWorksheet = "Bad Worksheet Identifier"
    Exit Function
End If
worksheet.Delete
RemoveWorksheet = "OK"
End Function

```

```

*****
' 函数说明: 添加新的工作薄
' 参数说明:
'     (1)ExcelApp:Excel 应用程序名称;
' 返回结果:
'     (1)成功: 工作表对象 NewWorkbook
'     (1)失败: Nothing
' 调用方法:
'     set NewWorkbook = CreateNewWorkbook(ExcelApp)
*****
```

```

Function CreateNewWorkbook(ExcelApp)
    Set NewWorkbook = ExcelApp.Workbooks.Add()
    Set CreateNewWorkbook = NewWorkbook
End Function

```

```

*****
' 函数说明: 打开工作薄
' 参数说明:
'     (1)ExcelApp:Excel 应用程序名称;
'     (2)path:要打开的工作薄路径;
' 返回结果:
'     (1)成功: 工作表对象 NewWorkbook
'     (1)失败: Nothing
' 调用方法:
'     set NewWorkbook = CreateNewWorkbook(ExcelApp)
*****
```

```
Function OpenWorkbook(ExcelApp, path)
```

自动化测试	Version:	1.3
文档		2006-5-17

```

On Error Resume Next
Set NewWorkbook = ExcelApp.Workbooks.Open(path)
Set OpenWorkbook = NewWorkbook
On Error GoTo 0
End Function

'
*****函数说明: 将工作薄设置为当前工作状态
'参数说明:
'(1)ExcelApp:Excel 应用程序名称;
'(2)workbookIdentifier:要设置为当前工作状态的工作薄名称;
'返回结果: 无返回值;
'调用方法:
'    ActivateWorkbook(ExcelApp, workbook1)
'

*****Sub ActivateWorkbook(ExcelApp, workbookIdentifier)
On Error Resume Next
ExcelApp.Workbooks(workbookIdentifier).Activate
On Error GoTo 0
End Sub

'
*****'函数说明: 关闭 Excel 工作薄;
'参数说明:
'(1)ExcelApp:Excel 应用程序名称;
'(2)workbookIdentifier:
'调用方法:
'    CloseWorkbook(ExcelApp, workbookIdentifier)
'

*****Sub CloseWorkbook(ExcelApp, workbookIdentifier)
On Error Resume Next
ExcelApp.Workbooks(workbookIdentifier).Close
On Error GoTo 0
End Sub

'
*****'函数说明: 判断两个工作表对应单元格内容是否相等
'参数说明:
'(1)sheet1:工作表 1 的名称;
'(2)sheet2:工作表 2 的名称;
'(3)startColumn:开始比较的行序号;
'(4)numberOfColumns:要比较的行数;
'(5)startRow:开始比较的列序号;
'(6)numberOfRows:要比较的列数;

```

自动化测试	Version:	1.3
文档		2006-5-17

```

' (7)trimed:是否先除去字符串开始的空格和尾部空格后再进行比较,true 或 flase;
' 返回结果:
' (1)两工作表对应单元格内容相等: true
' (2)两工作表对应单元格内容不相等:flase
' 调用方法:
'     ret = CompareSheets(excelSheet1, excelSheet2, 1, 10, 1, 10, False)
'

*****
Function CompareSheets(sheet1, sheet2, startColumn, numberOfColumns, startRow, numberOfRows, trimed)
    Dim returnVal 'As Boolean
    returnVal = True

    '判断两个工作表是否都存在,任何一个不存在停止判断, 返回 flase
    If sheet1 Is Nothing Or sheet2 Is Nothing Then
        CompareSheets = False
        Exit Function
    End If

    '循环判断两个工作表单元格的值是否相等
    For r = startRow To (startRow + (numberOfRows - 1))
        For c = startColumn To (startColumn + (numberOfColumns - 1))
            Value1 = sheet1.Cells(r, c)
            Value2 = sheet2.Cells(r, c)

            '如果 trimed 为 true,去除单元格内容前面和尾部空格
            If trimed Then
                Value1 = Trim(Value1)
                Value2 = Trim(Value2)
            End If

            '如果单元格内容不一致,函数返回 flase
            If Value1 <> Value2 Then
                Dim cell 'As Excel.Range
                '修改 sheet2 工作表中对应单元格值
                sheet2.Cells(r, c) = "Compare conflict - Value was " & Value2 & ", Expected value is " & Value1 &
                """
                '初始化 cell 为 sheet2 中 r:c 单元格对象
                Set cell = sheet2.Cells(r, c)'
                '将 sheet2 工作表中对应单元格的颜色设置为红色
                cell.Font.Color = vbRed
                returnVal = False
            End If
        Next
    Next
    CompareSheets = returnVal
End Function

```

自动化测试	Version:	1.3
文档	2006-5-17	

37. Action parameters

QTP 输出参数用法(图解)



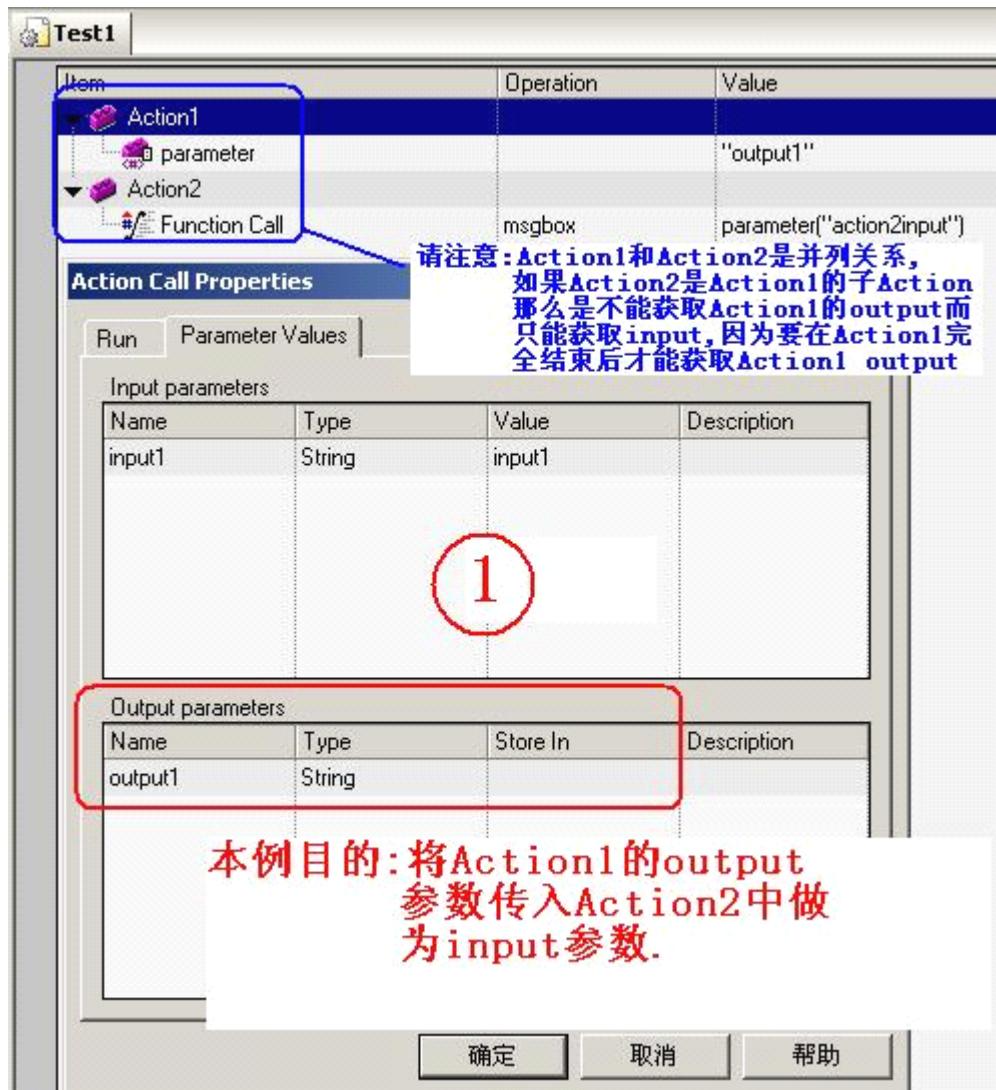
QTP 输出参数用法(图解)

[本帖最后由 梦醒十分 于 2007-1-29 09:08 编辑]

自动化测试	Version:	1.3
文档	2006-5-17	

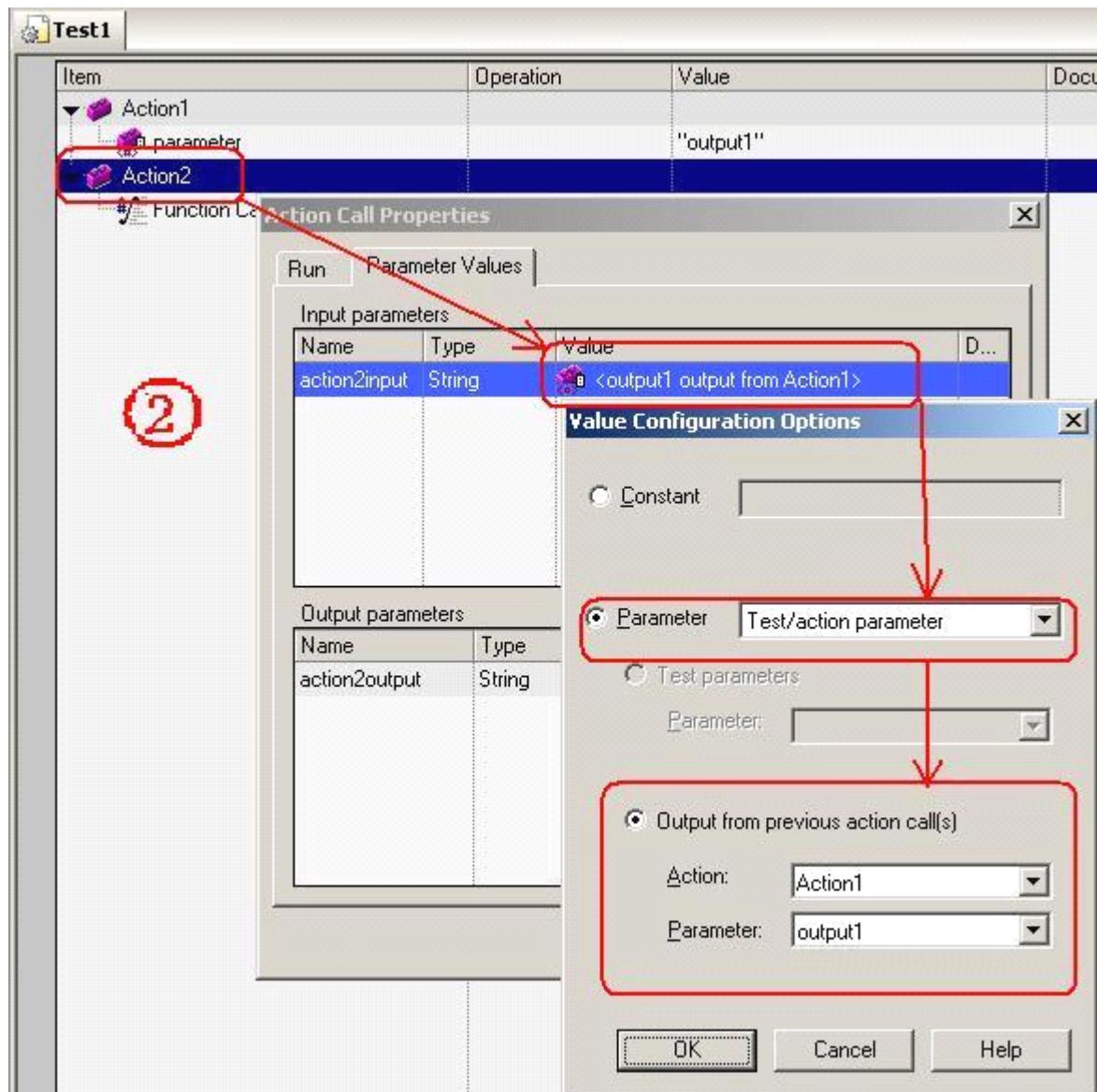
自动化测试	Version:	1.3
文档		2006-5-17

图片附件: [1.jpg](#) (2007-1-29 09:07, 57.44 K)

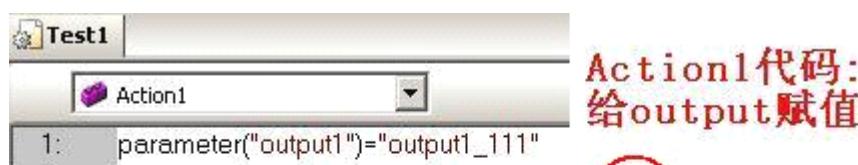


图片附件: [2.jpg](#) (2007-1-29 09:07, 53.11 K)

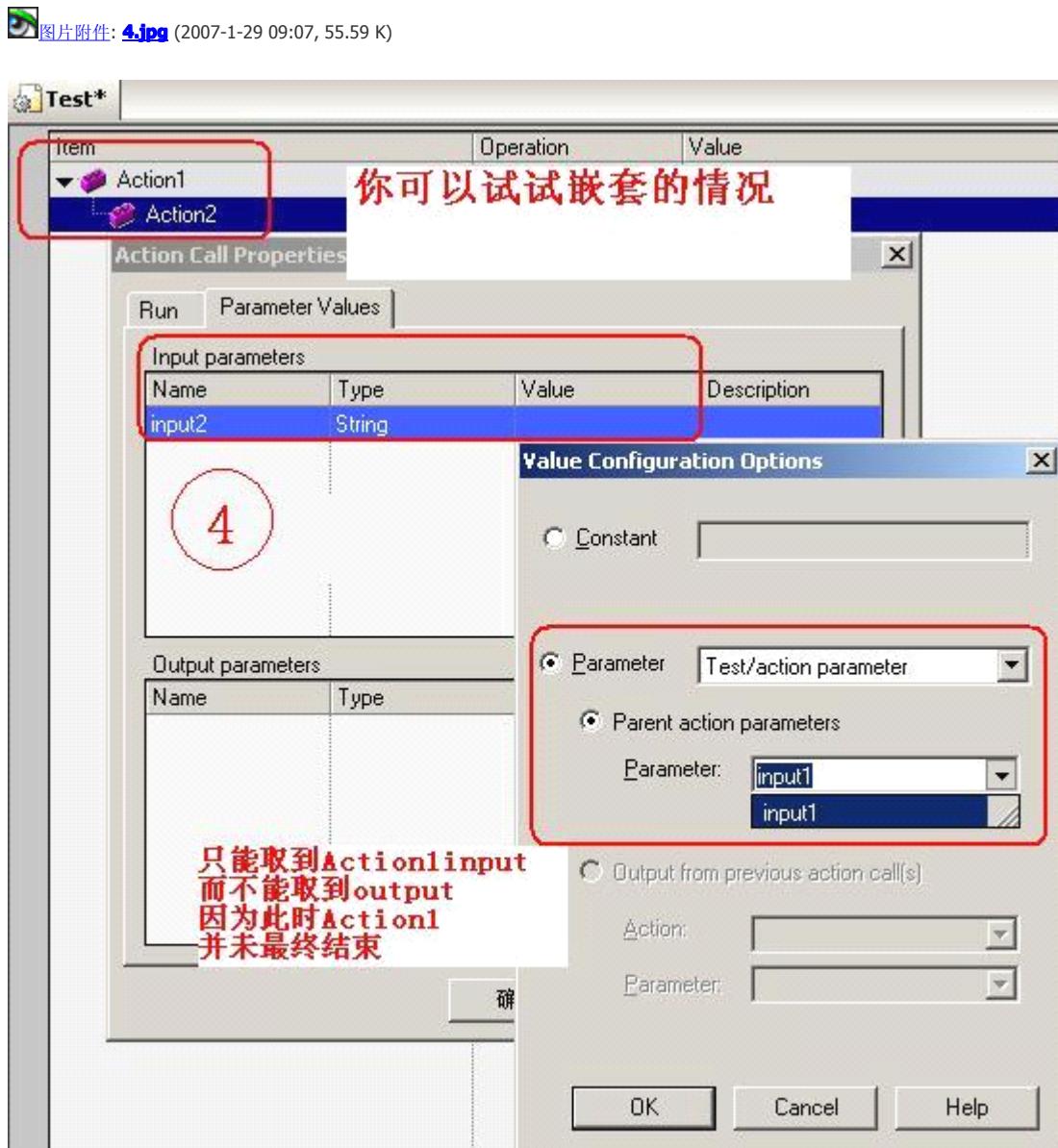
自动化测试	Version: 1.3
文档	2006-5-17



图片附件: 3.jpg (2007-1-29 09:07, 20.61 K)



自动化测试	Version: 1.3
文档	2006-5-17



38. 数据驱动测试: 文件

file operation

自动化测试	Version:	1.3
文档		2006-5-17

```
Dim fso, f, f1, fc, s
Set fso = CreateObject("Scripting.FileSystemObject")
Set f = fso.GetFolder("C:\MyTest")           '目的是删除该文件夹下所有的文件
夹
Set fc = f.SubFolders

For Each f1 in fc                          '逐一获取子文件夹名称并将子文件夹及下属文件删除
    s = f1.name
    Dim ff
    Set ff = CreateObject("Scripting.FileSystemObject")
    ff.DeleteFolder("C:\MyTest\" & f1.name)
    msgbox (s & "已成功删除")
Next
```

上述代码运行通过,不过要注意要先建好文件夹

 每天一点小小的积累,积少成多 

 (0)  (0) 引用 报告 回复 

自动化测试	Version:	1.3
文档	2006-5-17	

Coffey111111

中级站友



#7

发表于 2007-1-29 12:03 资料 短消息



学习~~ 😊



DISCUZ

Have a try until you succeed!! You will meet bright one day~~

(0) (0) 引用 报告 回复

UID 90645

精华 0

积分 141

帖子 121

综合技术指数 141

生活情趣指数 0

鲜花 0

鸡蛋 0

阅读权限 10

注册 2006-12-1

来自 上海

状态 离线

自动化测试	Version:	1.3
文档		2006-5-17

zpf
一般站长


#8
发表于 2007-2-1 16:08 资料 短消息

找到个全的！分享下

UID 85176

精华 0

积分 57

帖子 37

综合技术指数 57

生活情趣指数 0

鲜花 0

鸡蛋 0

阅读权限 10

注册 2006-10-30

状态 离线

创建文件

```
dim fso, f
set fso = server.CreateObject("Scripting.FileSystemObject")
set f = fso.CreateTextFile("C:\test.txt", true) '第二个参数表示目标文件存在时是否覆盖
f.Write("写入内容")
f.WriteLine("写入内容并换行")
f.WriteBlankLines(3) '写入三个空白行（相当于在文本编辑器中按三次回车）
f.Close()
set f = nothing
set fso = nothing
```

打开并读文件

```
dim fso, f
set fso = server.CreateObject("Scripting.FileSystemObject")
set f = fso.OpenTextFile("C:\test.txt", 1, false) '第二个参数 1 表示只读打开，第三个参数表示目标文件不存在时是否创建
f.Skip(3) '将当前位置向后移三个字符
f.SkipLine() '将当前位置移动到下一行的第一个字符，注意：无参数
response.Write f.Read(3) '从当前位置向后读取三个字符，并将当前位置向后移三个字符
response.WriteLine() '从当前位置向后读取直到遇到换行符（不读取换行符），并将当前位置移动到下一行的第一个字符，注意：无参数
response.Write f.ReadAll() '从当前位置向后读取，直到文件结束，并将当前位置移动到文件的最后
if f.atEndOfLine then
    response.Write("一行的结尾！")
end if
if f.atEndOfStream then
    response.Write("文件的结尾！")
end if
f.Close()
set f = nothing
set fso = nothing
```

打开并写文件

```
dim fso, f
set fso = server.CreateObject("Scripting.FileSystemObject")
set f = fso.OpenTextFile("C:\test.txt", 2, false) '第二个参数 2 表示重写，如果是 8 表示追加
f.Write("写入内容")
f.WriteLine("写入内容并换行")
f.WriteBlankLines(3) '写入三个空白行（相当于在文本编辑器中按三次回车）
f.Close()
set f = nothing
set fso = nothing
```

判断文件是否存在

自动化测试	Version:	1.3
文档		2006-5-17

```
dim fso
set fso = server.CreateObject("Scripting.FileSystemObject")
if fso.FileExists("C:\test.txt") then
    response.Write("目标文件存在")
else
    response.Write("目标文件不存在")
end if
set fso = nothing

移动文件
dim fso
set fso = server.CreateObject("Scripting.FileSystemObject")
call fso.MoveFile("C:\test.txt", "D:\test111.txt") '两个参数的文件名部分可以不同
set fso = nothing

复制文件
dim fso
set fso = server.CreateObject("Scripting.FileSystemObject")
call fso.CopyFile("C:\test.txt", "D:\test111.txt") '两个参数的文件名部分可以不同
set fso = nothing

删除文件
dim fso
set fso = server.CreateObject("Scripting.FileSystemObject")
fso.DeleteFile("C:\test.txt")
set fso = nothing

创建文件夹
dim fso
set fso = server.CreateObject("Scripting.FileSystemObject")
fso.CreateFolder("C:\test") '目标文件夹的父文件夹必须存在
set fso = nothing

判断文件夹是否存在
dim fso
set fso = server.CreateObject("Scripting.FileSystemObject")
if fso.FolderExists("C:\Windows") then
    response.Write("目标文件夹存在")
else
    response.Write("目标文件夹不存在")
end if
set fso = nothing

删除文件夹
dim fso
set fso = server.CreateObject("Scripting.FileSystemObject")
fso.DeleteFolder("C:\test") '文件夹不必为空
set fso = nothing
```

 (0)  (0) 引用 报告 回复 

自动化测试	Version:	1.3
文档	2006-5-17	

vbic

51Testing 就业学员



UID 79428

精华 0

积分 51

帖子 31

综合技术指数 50

生活情趣指数 1

鲜花 0

鸡蛋 0

阅读权限 10

注册 2006-9-22

状态 离线

#9

发表于 2007-2-2 16:44 资料 个人空间 短消息



哈哈哈

楼上写这么多啊,

我也来写一个,你好像是只要删除一个非空目录吧

```
'path 为要删除目录路径
sub delfolder(path)
Dim WshShell, oExec
Set WshShell = CreateObject("WScript.Shell")
Set oExec = WshShell.Exec("cmd /c del /s /q "+path)
set oexec = wshshell.exec("cmd /c rd /s /q "+path)
end sub
```

文件夹信息的话

我喜欢用 dir,dir 功能很强大

自动化测试	Version:	1.3
文档		2006-5-17

39. 相对路径 action path

【分享】使用相对路径 call 外部 action

使用相对路径 call 外部 action 有很多好处。当 test 所在路径变化时，只要外部 action 所在的 test 的相对路径没变，且文件名没变，那么路径不管怎么变化，对外部 action 的 call 都有效，否则只要一改变路径，external action 都不可用。

使用相对路径的方法：

1).首先，在 Tools > Options > Folders 里，把存放脚本的路径加进来。比如你有 test1, test2 脚本文件，都放在路径 d:\QTP_Test 下，

那么加入路径 d:\QTP_Test。

2).在 test1 中插入 test2 中的 reusable actionA 的方法：在 Insert call to existing Action 的对话框中，From test 栏不要点击浏览按钮，而是直接输入 test2，然后 Action 下拉框中会自动列出 test2 中可重用的 action，此时选中 actionA，并确定。

3).验证是否是以相对路径的方式插入外部 Action：在刚插入的 actionA 上点击右键，Action Properties，在 Location 栏中显示的应该是 test2\actionA，那么就对了。如果显示的是 d:\QTP_Test\test2\actionA，说明还是以绝对路径的方式插入的，只要 d:\QTP_Test 的路径变了，所有的外部 Action 都不能访问。

比较麻烦的是，原来用绝对路径方式插入的 action，都不能修改，只能删除后重新按相对路径方式插入

40. 两种方法参数化 link

18 link parameter

2 种参数化 LINK 的方法

看到顶置贴，有点启发，其实还可以有另外 2 种方法实现 Link 的参数化

利用 Description 对象，Description 对象有 5 个方法

- Add Method
- Count Property
- Remove Method
- RegularExpression Property
- Value Property

CODE:

```
For intLoop = 1 to N
    strText=DataTable.Value(...)
    Set LinkDesc = Description.Create()
    LinkDesc ("Text").Value = strText
    Browser("").Page("").Link(LinkDesc).Click
```

自动化测试	Version:	1.3
文档		2006-5-17

```
DataTable.GetSheet("").SetNextRow  
Next
```

或者还可以用以下方法，描述性编程，格式 **object("Property:=" & Parameterized_Variant)**

CODE:

```
For intLoop = 1 to N  
strText=DataTable.Value(...)  
Browser("").Page("").Link("text:=" & strText).Click  
DataTable.GetSheet("").SetNextRow  
Next
```

41. WScript

WScript

本主题由 **QA_BAY** 于 **2006-8-4 16:24** 加入精华

自动化测试	Version:	1.3
文档		2006-5-17

skyhen

中级校友



UID 50473

精华 1

积分 147

帖子 108

综合技术指数 137

生活情趣指数 10

鲜花 0

鸡蛋 0

阅读权限 10

注册 2006-4-13

状态 离线

#1

发表于 2006-8-2 16:10 资料 短消息

请教各位哥哥姐姐，关于 WScript 这个的问题

```
Dim WshShell, oExec
Set WshShell = CreateObject("WScript.Shell")
Set oExec    = WshShell.Exec("%comspec% /c dir")

Function ReadAllFromAny(oExec)

If Not oExec.StdOut.AtEndOfStream Then
    ReadAllFromAny = oExec.StdOut.ReadAll
    Exit Function
End If

If Not oExec.StdErr.AtEndOfStream Then
    ReadAllFromAny = oExec.StdErr.ReadAll
    Exit Function
End If

ReadAllFromAny = -1
End Function

Dim allInput, tryCount

allInput = ""
tryCount = 0

Do While True

    Dim input
    input = ReadAllFromAny(oExec)

    If -1 = input Then
        If tryCount > 10 And oExec.Status = 1 Then
            Exit Do
        End If
        tryCount = tryCount + 1
        WScript.Sleep 100
    Else
        allInput = allInput & input
        tryCount = 0
    End If
Loop

If oExec.ExitCode <> 0 Then
    msgbox "Warning: Non-zero exit code"
End If

msgbox allInput
```

自动化测试	Version:	1.3
文档	2006-5-17	

运行到" WScript.Sleep 100 "这句话时，系统提示" 缺少对象: 'WScript' "，请问各位帮小弟看一下，问题错在哪里，小弟先谢谢了。

 (0)  (0) 引用 报告 回复 

[广告]

自动化测试	Version:	1.3
文档		2006-5-17

xiaonan

版主



UID 26649

精华 0

积分 1680

帖子 1610

综合技术指数 160

5

生活情趣指数 75

鲜花 0

鸡蛋 0

阅读权限 100

注册 2005-9-8

来自 上海

状态 离线

#2

发表于 2006-8-2 17:06 资料 个人空间 短消息

把这两句放进循环里面

```
Set WshShell = CreateObject("WScript.Shell")
Set oExec    = WshShell.Exec("%comspec% /c dir")
```

如：

```
Do While True
Set WshShell = CreateObject("WScript.Shell")
Set oExec    = WshShell.Exec("%comspec% /c dir")
```

```
Dim input
input = ReadAllFromAny(oExec)
```

```
If -1 = input Then
    If tryCount > 10 And oExec.Status = 1 Then
        Exit Do
    End If
    tryCount = tryCount + 1
    WScript.Sleep 100
Else
    allInput = allInput & input
    tryCount = 0
End If
Loop
```

你这程序有什么意义？



欢迎光临 xiaonan 的个人空间

DISCUZZ

(0) (0) 引用 报告 回复

[广告] 论坛发贴就是写博客，51Testing 博客全新登场，论坛博客同一帐号，咱的空间咱做主，拥有自己的个性空间：blog.51testing.com

自动化测试	Version: 1.3
文档	2006-5-17

QA_BAY

版主



UID 4665

精华 4

积分 2278

帖子 1786

综合技术指数 2204

生活情趣指数 74

鲜花 0

鸡蛋 0

阅读权限 100

注册 2004-10-13

来自 广州

状态 离线

#3

发表于 2006-8-2 18:01 资料 文集 短消息

QUOTE:

原帖由 *xiaonan* 于 2006-8-2 17:06 发表

把这两句放进循环里面

```
Set WshShell = CreateObject("WScript.Shell")
Set oExec    = WshShell.Exec("%comspec% /c dir")
```

如：

```
Do While True
Set WshShell = CreateObject("WSc ...")
```

把那两句放进循环里面,脚本是可以执行,但没有执行到

WScript.Sleep 100 ,如果执行到了还是会提示“缺少对象: 'WScript'

把楼主的脚本放到 DOS 下运行是没有问题的,但放到 QTP 里面运行就提示有问题,
我觉得有可能是语言的问题,老是提示没有定义,但定义了也是提示不行,执行 QTP HELP 里面的
脚本也是这样.

你可以这样做,既然 DOS 可以执行,那你就用 QTP 执行 DOS 调用脚本,这样是没有问题的.
其它的办法还没有想到!

[本帖最后由 QA_BAY 于 2006-8-2 18:03 编辑]

(0) (0) 引用 报告 回复

[广告] 当您在使用论坛过程中遇到问题时可发送 email:webmaster@51testing.com 或发送论
坛短信至管理员“风在吹”

自动化测试	Version:	1.3
文档	2006-5-17	

skyhen

中级校友



#4

发表于 2006-8-3 11:32 资料 短消息

UID 50473

精华 1

积分 147

帖子 108

综合技术指数 137

生活情趣指数 10

鲜花 0

鸡蛋 0

阅读权限 10

注册 2006-4-13

状态 离线



(0)



(0)

引用

报告

回复



自动化测试	Version:	1.3
文档	2006-5-17	

QA_BAY

版主



#5

发表于 2006-8-3 15:19 资料 文集 短消息

你可以这样做

先把脚本保存起来(保证你的脚本能正确运行),例如命名为 test.vbs
然后在 QTP 写以下脚本就可以正确运行了!

```
Dim oShell
Set oShell =CreateObject ("WSCript.shell")
oShell.run "cmd /K wscript e:\test.vbs"
Set oShell = Nothing
```



UID 4665

精华 4

积分 2278

帖子 1786

综合技术指数 2204

生活情趣指数 74

鲜花 0

鸡蛋 0

阅读权限 100

注册 2004-10-13

来自 广州

状态 离线

自动化测试	Version:	1.3
文档		2006-5-17

42. QTP Plus 的应用

QuickTest Plus 小工具，大作用

象我这样初学 QTP 的朋友刚开始时很可能没有注意到 QuickTest Plus，因为 QTP 安装后默认是没有安装 plus 的，千回百转知道了 plus，大概看了看，发现 plus 虽然都是些辅助性的小工具，但往往会给你的工作带来事半功倍的效果。

一、安装 QuickTest plus

QTP 安装后，在 程序 > QuickTest Professional 下点击 QuickTest Plus，然后按照提示一步步往下安装即可，其中要求输入序列号，输入和 QTP 安装时相同的序列号就可以了（8888-8888888888）。

二、提示和技巧

plus 不仅提供了一些工具，还在它的帮助手册里给出了一些提示和技巧，以及一些实用的 Function。在这里我把一些比较常用的好东东贴出来，其他的就看 plus 的帮助吧。

1、创建 action template。

当希望在每一个新建 action 时都增加一些头部说明，比如作者、创建日期、说明等，用 action template 来实现最简单快捷。

方法：用记事本等文本编辑器，输入如下类似的内容：

```
'Company: xxxx
'Author: xxx
'Product: xxx
'Date: xx
```

然后将文件保存为 ActionTemplate.mst，并放到 QTP 安装目录下的 dat 目录，重启 QTP，新建一个 action 试试，新建的 action 会包含以上信息。

2、关于设置测试报告里只显示 error 的信息。

帮助中说：修改安装目录下 bin\QTReport.ini 文件，增加以下内容：

```
[FilterDialog]
ReportAppDefaultFilter=1 # for error only
ReportAppDefaultFilter=3 # shows all messages (default)
```

但根据我的测试结果，不尽其然：

1) 当 ReportAppDefaultFilter=1 时，如果 Object Repository 中缺少对象，在报告中会在相应的

action 前打叉，但不会提示具体错误，而成功的步骤都有具体信息显示。

2) 用 Reporter.ReportEvent 测试的结果是：

```
ReportAppDefaultFilter=1 时，只显示 micDone 的具体信息；
ReportAppDefaultFilter=2 时，只显示 micFail 的具体信息；
ReportAppDefaultFilter=3 时，只显示 micDone 和 micFail 的具体信息；
ReportAppDefaultFilter=4 时，只显示 micPass 的具体信息；
```

似乎无规律可寻，所以我的结论暂时是：不要设置这个参数，用默认的，显示所有信息，更多的信息有利于分析结果。

3. 启动 IE 的语句：SystemUtil.Run "iexplore.exe", "http://www.mercuryinteractive.com"

4. 关闭 IE 或其他程序的语句：SystemUtil.CloseProcessByName "app.exe"
or SystemUtil.CloseProcessByWndTitle "Some Title"

三、Function Libraries

plus 的帮助中提供了一些常用的 Function，把这些 function copy 到文本编辑器中保存为.vbs 文件，并添加到 Resources 中就可以直接调用了，或直接 copy 到你的 action 中，就可以在当前的 action 中调用。更推荐第一种方法，所有的 action 都可以调用。

1、文件操作相关的 function，如下，望名则可生意：

```
Function CreateFile(sFilename, bOverwrite);
Function OpenFile(sFilename, iomode, create);
Function AppendToFile(sFilename, sLine);
Function WriteToFile(sFilename, sLine);
```

2、Function NormalizeString(OrgStr); （将字符串变成 regular express）

3、GlobalDictionary 的使用，这是另外一种可以共享全局变量的方法，在所有的 action 中，包括 local 和 external action 中都可以访问。

自动化测试		Page 111
-------	--	----------

自动化测试	Version:	1.3
文档		2006-5-17

4. 使用文件系统相关的 function:

```
Function ReadLineFromFile (byref FileRef);
Sub FileDelete ( FilePath);
Function FileCompare (byref FilePath1, byref FilePath2, byref FilePathDiff, ignoreWhiteSpace);
Function CheckFileExists (FilePath)
```

5. web table 相关的 function:

```
Function ItemByKeyColumn(): 根据 table 中某列的值，得到同一行中另一列的对象。（这个功能非常有用。）
Function ObjectsByMicClass(Obj, micClass): 得到 table 中所有的 micClass 类型的对象集合。
```

四、工具

1. Automation Generator Utility

添加一系列动作，然后自动完成。如：启动 QTP，然后执行 test1, test2, test3...，最后关闭 QTP，还可以连接或断开 Quality center。
值得注意的是，每个 test 都可以分别指定 test result 文件，这样可以把所有 test result 指定到同一个目录下，所有 test 执行完成后，用 Test Result Viewer 就可以快速的查看测试报告。

2. External Action Call Modifier Utility

外部 Action 调用修改工具。当删除一个 Reusable action 前，如果没有先删除其他 test 中对它的调用，则打开其他 test 时，会提示找不到某某 action，这种情况下启用该工具，工具会列出 test 下所有调用的外部 action（如果调用的 action 是使用相对路径，则不会列出来），如果外部 action 找不到，会用红色的问号表示，删除它并保存后就 OK 了。
也可以在此处修改外部 action 的来源，修改后原来的 action parameter 仍会保留。

3. Repository Merge Utility

合并多个对象库文件中的对象，如果出现冲突，可以选择忽略，或手工合并，或自动合并。
对象库文件是 Action 目录下的 Resource.mtr 文件。

4. Report Analyzer

Test Result 的另一个查看工具，提供了一些过滤条件，比如只查看 failed step，或只查看 checkpoints，或只查看某个 action。

先就写这么多了，抛砖引玉，希望你能发现你想要的东西。写得不对的，不要客气，请指出来，不胜感激。

[本帖最后由 麦子华华 于 2006-8-22 12:13 编辑]

43. 一个万能的键盘模拟函数

一个万能的键盘模拟函数

'对于一些能识别但不能通过鼠标操作，通过键盘操作可以达到目的的情况很有用

'对你又用一定要顶阿，给点鼓励，以后我会陆续贴些又帮助的代码

'声明

Extern.Declare micVoid, "keybd_event", "user32.dll", "keybd_event", micByte,micByte,micDWord,micULong

'调用

Extern.keybd_event 38,0,0,0

'一些虚拟键盘码

'vk_up 0x26

'vk_down 0x28

'vk_home 0x24

'vk_space 0x20

'vk_back 0x08

'vk_end 0x23

自动化测试	Version:	1.3
文档		2006-5-17

Symbolic constant name Value
 (hexadecimal) Mouse or keyboard equivalent

VK_LBUTTON 01 Left mouse button
 VK_RBUTTON 02 Right mouse button
 VK_CANCEL 03 Control-break processing
 VK_MBUTTON 04 Middle mouse button (three-button mouse)
 VK_XBUTTON1 05 Windows 2000/XP: X1 mouse button
 VK_XBUTTON2 06 Windows 2000/XP: X2 mouse button
 — 07 Undefined
 VK_BACK 08 BACKSPACE key
 VK_TAB 09 TAB key
 — 0A–0B Reserved
 VK_CLEAR 0C CLEAR key
 VK_RETURN 0D ENTER key
 — 0E–0F Undefined
 VK_SHIFT 10 SHIFT key
 VK_CONTROL 11 CTRL key
 VK_MENU 12 ALT key
 VK_PAUSE 13 PAUSE key
 VK_CAPITAL 14 CAPS LOCK key
 VK_KANA 15 IME Kana mode
 VK_HANGUEL 15 IME Hanguel mode (maintained for compatibility; use VK_HANGUL)
 VK_HANGUL 15 IME Hangul mode
 — 16 Undefined
 VK_JUNJA 17 IME Junja mode
 VK_FINAL 18 IME final mode
 VK_HANJA 19 IME Hanja mode
 VK_KANJI 19 IME Kanji mode
 — 1A Undefined
 VK_ESCAPE 1B ESC key
 VK_CONVERT 1C IME convert
 VK_NONCONVERT 1D IME nonconvert
 VK_ACCEPT 1E IME accept
 VK_MODECHANGE 1F IME mode change request
 VK_SPACE 20 SPACEBAR
 VK_PRIOR 21 PAGE UP key
 VK_NEXT 22 PAGE DOWN key
 VK_END 23 END key
 VK_HOME 24 HOME key
 VK_LEFT 25 LEFT ARROW key
 VK_UP 26 UP ARROW key
 VK_RIGHT 27 RIGHT ARROW key
 VK_DOWN 28 DOWN ARROW key
 VK_SELECT 29 SELECT key
 VK_PRINT 2A PRINT key
 VK_EXECUTE 2B EXECUTE key
 VK_SNAPSHOT 2C PRINT SCREEN key
 VK_INSERT 2D INS key
 VK_DELETE 2E DEL key
 VK_HELP 2F HELP key
 30 0 key
 31 1 key

自动化测试	Version:	1.3
文档		2006-5-17

32 2 key
 33 3 key
 34 4 key
 35 5 key
 36 6 key
 37 7 key
 38 8 key
 39 9 key
 — 3A–40 Undefined
 41 A key
 42 B key
 43 C key
 44 D key
 45 E key
 46 F key
 47 G key
 48 H key
 49 I key
 4A J key
 4B K key
 4C L key
 4D M key
 4E N key
 4F O key
 50 P key
 51 Q key
 52 R key
 53 S key
 54 T key
 55 U key
 56 V key
 57 W key
 58 X key
 59 Y key
 5A Z key
 VK_LWIN 5B Left Windows key (Microsoft® Natural® keyboard)
 VK_RWIN 5C Right Windows key (Natural keyboard)
 VK_APPS 5D Applications key (Natural keyboard)
 — 5E Reserved
 VK_SLEEP 5F Computer Sleep key
 VK_NUMPAD0 60 Numeric keypad 0 key
 VK_NUMPAD1 61 Numeric keypad 1 key
 VK_NUMPAD2 62 Numeric keypad 2 key
 VK_NUMPAD3 63 Numeric keypad 3 key
 VK_NUMPAD4 64 Numeric keypad 4 key
 VK_NUMPAD5 65 Numeric keypad 5 key
 VK_NUMPAD6 66 Numeric keypad 6 key
 VK_NUMPAD7 67 Numeric keypad 7 key
 VK_NUMPAD8 68 Numeric keypad 8 key
 VK_NUMPAD9 69 Numeric keypad 9 key
 VK_MULTIPLY 6A Multiply key
 VK_ADD 6B Add key

自动化测试	Version:	1.3
文档		2006-5-17

VK_SEPARATOR 6C Separator key
 VK_SUBTRACT 6D Subtract key
 VK_DECIMAL 6E Decimal key
 VK_DIVIDE 6F Divide key
 VK_F1 70 F1 key
 VK_F2 71 F2 key
 VK_F3 72 F3 key
 VK_F4 73 F4 key
 VK_F5 74 F5 key
 VK_F6 75 F6 key
 VK_F7 76 F7 key
 VK_F8 77 F8 key
 VK_F9 78 F9 key
 VK_F10 79 F10 key
 VK_F11 7A F11 key
 VK_F12 7B F12 key
 VK_F13 7C F13 key
 VK_F14 7D F14 key
 VK_F15 7E F15 key
 VK_F16 7F F16 key
 VK_F17 80H F17 key
 VK_F18 81H F18 key
 VK_F19 82H F19 key
 VK_F20 83H F20 key
 VK_F21 84H F21 key
 VK_F22 85H F22 key
 VK_F23 86H F23 key
 VK_F24 87H F24 key
 — 88–8F Unassigned
 VK_NUMLOCK 90 NUM LOCK key
 VK_SCROLL 91 SCROLL LOCK key
 92–96 OEM specific
 — 97–9F Unassigned
 VK_LSHIFT A0 Left SHIFT key
 VK_RSHIFT A1 Right SHIFT key
 VK_LCONTROL A2 Left CONTROL key
 VK_RCONTROL A3 Right CONTROL key
 VK_LMENU A4 Left MENU key
 VK_RMENU A5 Right MENU key
 VK_BROWSER_BACK A6 Windows 2000/XP: Browser Back key
 VK_BROWSER_FORWARD A7 Windows 2000/XP: Browser Forward key
 VK_BROWSER_REFRESH A8 Windows 2000/XP: Browser Refresh key
 VK_BROWSER_STOP A9 Windows 2000/XP: Browser Stop key
 VK_BROWSER_SEARCH AA Windows 2000/XP: Browser Search key
 VK_BROWSER_FAVORITES AB Windows 2000/XP: Browser Favorites key
 VK_BROWSER_HOME AC Windows 2000/XP: Browser Start and Home key
 VK_VOLUME_MUTE AD Windows 2000/XP: Volume Mute key
 VK_VOLUME_DOWN AE Windows 2000/XP: Volume Down key
 VK_VOLUME_UP AF Windows 2000/XP: Volume Up key
 VK_MEDIA_NEXT_TRACK B0 Windows 2000/XP: Next Track key
 VK_MEDIA_PREV_TRACK B1 Windows 2000/XP: Previous Track key
 VK_MEDIA_STOP B2 Windows 2000/XP: Stop Media key

自动化测试	Version:	1.3
文档		2006-5-17

VK_MEDIA_PLAY_PAUSE B3 Windows 2000/XP: Play/Pause Media key

VK_LAUNCH_MAIL B4 Windows 2000/XP: Start Mail key

VK_LAUNCH_MEDIA_SELECT B5 Windows 2000/XP: Select Media key

VK_LAUNCH_APP1 B6 Windows 2000/XP: Start Application 1 key

VK_LAUNCH_APP2 B7 Windows 2000/XP: Start Application 2 key

— B8-B9 Reserved

VK_OEM_1 BA Used for miscellaneous characters; it can vary by keyboard.
Windows 2000/XP: For the US standard keyboard, the ';' key

VK_OEM_PLUS BB Windows 2000/XP: For any country/region, the '+' key

VK_OEM_COMMMA BC Windows 2000/XP: For any country/region, the ',' key

VK_OEM_MINUS BD Windows 2000/XP: For any country/region, the '-' key

VK_OEM_PERIOD BE Windows 2000/XP: For any country/region, the '.' key

VK_OEM_2 BF Used for miscellaneous characters; it can vary by keyboard.

Windows 2000/XP: For the US standard keyboard, the '?' key

VK_OEM_3 C0 Used for miscellaneous characters; it can vary by keyboard.

Windows 2000/XP: For the US standard keyboard, the '` ~` key

— C1-D7 Reserved

— D8-DA Unassigned

VK_OEM_4 DB Used for miscellaneous characters; it can vary by keyboard.

Windows 2000/XP: For the US standard keyboard, the '[' key

VK_OEM_5 DC Used for miscellaneous characters; it can vary by keyboard.

Windows 2000/XP: For the US standard keyboard, the '\' key

VK_OEM_6 DD Used for miscellaneous characters; it can vary by keyboard.

Windows 2000/XP: For the US standard keyboard, the ']' key

VK_OEM_7 DE Used for miscellaneous characters; it can vary by keyboard.

Windows 2000/XP: For the US standard keyboard, the 'single-quote/double-quote' key

VK_OEM_8 DF Used for miscellaneous characters; it can vary by keyboard.

— E0 Reserved

E1 OEM specific

VK_OEM_102 E2 Windows 2000/XP: Either the angle bracket key or the backslash key on the RT 102-key keyboard

E3-E4 OEM specific

VK_PROCESSKEY E5 Windows 95/98/Me, Windows NT 4.0, Windows 2000/XP: IME PROCESS key

E6 OEM specific

VK_PACKET E7 Windows 2000/XP: Used to pass Unicode characters as if they were keystrokes. The VK_PACKET key is the low word of a 32-bit Virtual Key value used for non-keyboard input methods. For more information, see Remark in KEYBDINPUT, SendInput, WM_KEYDOWN, and WM_KEYUP

— E8 Unassigned

E9-F5 OEM specific

VK_ATTN F6 Attn key

VK_CRSEL F7 CrSel key

VK_EXSEL F8 ExSel key

VK_EREOF F9 Erase EOF key

VK_PLAY FA Play key

VK_ZOOM FB Zoom key

VK_NONAME FC Reserved for future use

自动化测试	Version:	1.3
文档		2006-5-17

VK_PA1 FD PA1 key
VK_OEM_CLEAR FE Clear key

44. XML 参数化

XML 参数化

<http://bbs.51testing.com/thread-13762-1-3.html>

45. Oracle QTP

oracle QTP

试着连了下，应该可以的

```

Dim rs,sq
set conn=createobject("adodb.connection")
set rs=createobject("adodb.recordset")
' Oracle client is required to be installed on your box
conn.open="Driver={Microsoft ODBC for Oracle};Server=SID_HOST.DOMAIN.COM;Uid=mssqa;Pwd=password;"

sql="select * from CLIENT_TABLE"
rs.open sql,conn,3,3

'Services.StartTransaction "TIMER"

do while not rs.eof
values=rs("CLIENT_NAME")
if values<>"State Street Corporation" then
    Reporter.ReportEvent 2, "ClientNameSearched:&" &values, ""
    rs.movenext
else
    Reporter.ReportEvent 0, "ClientNameFound:&" &values, ""
    exit do
end if
Loop

'Services.EndTransaction "TIMER"

rs.close
set rs=nothing
conn.close
set conn=nothing

```

自动化测试	Version:	1.3
文档		2006-5-17

参考资料

1. VBScript Database Tutorial <http://www.electronics.dit.ie/st...base%20Tutorial.pdf>
2. ADO 介绍 <http://www.51windows.net/pages/a...scadogsoverview.htm>
3. 数据库连接字符串 <http://www.connectionstrings.com/>

<http://www.51testing.com/html/27/2111.html>

46. 对象库的管理与合并

Object Repository 的合并

文章出处: 51testing 博客 作者: baoju 发布时间: 2007-02-12

Object Repository 合并的功能是在 qtp9.0 时才出现的, 这个功能对于测试脚本整体开发具有很大的意义.

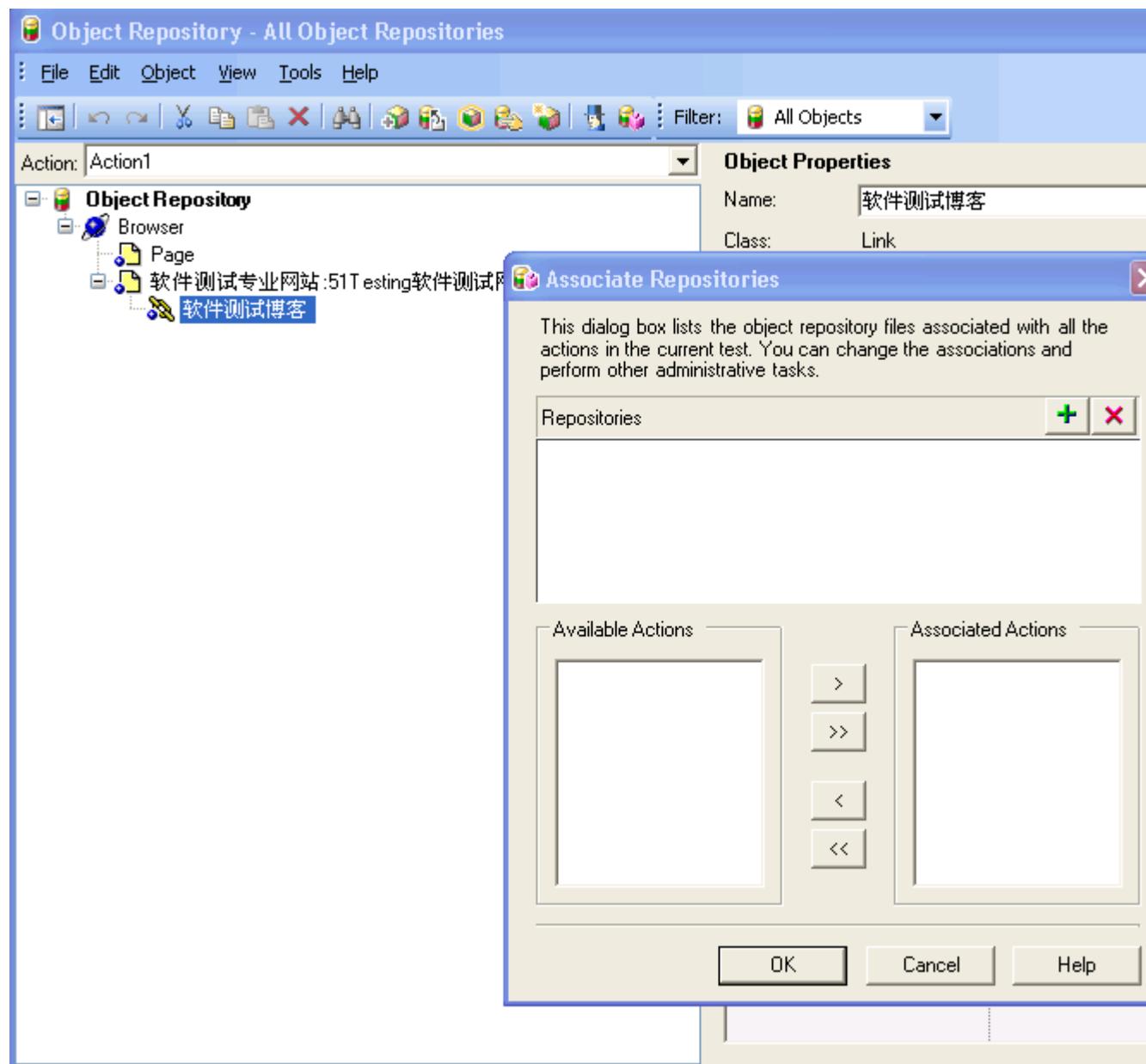
在之前的版本如果要使一些常用的步骤变为函数, 还必须在同一个 object Repository 中添加对象, 有了这个功能之后, 你写完测试脚本, 然后再挑选一些常用的步骤进行函数化, 方便了操作.

下面介绍合并 Object Repository 的三种方法:

1) 从共用对象库合并到本地对象库

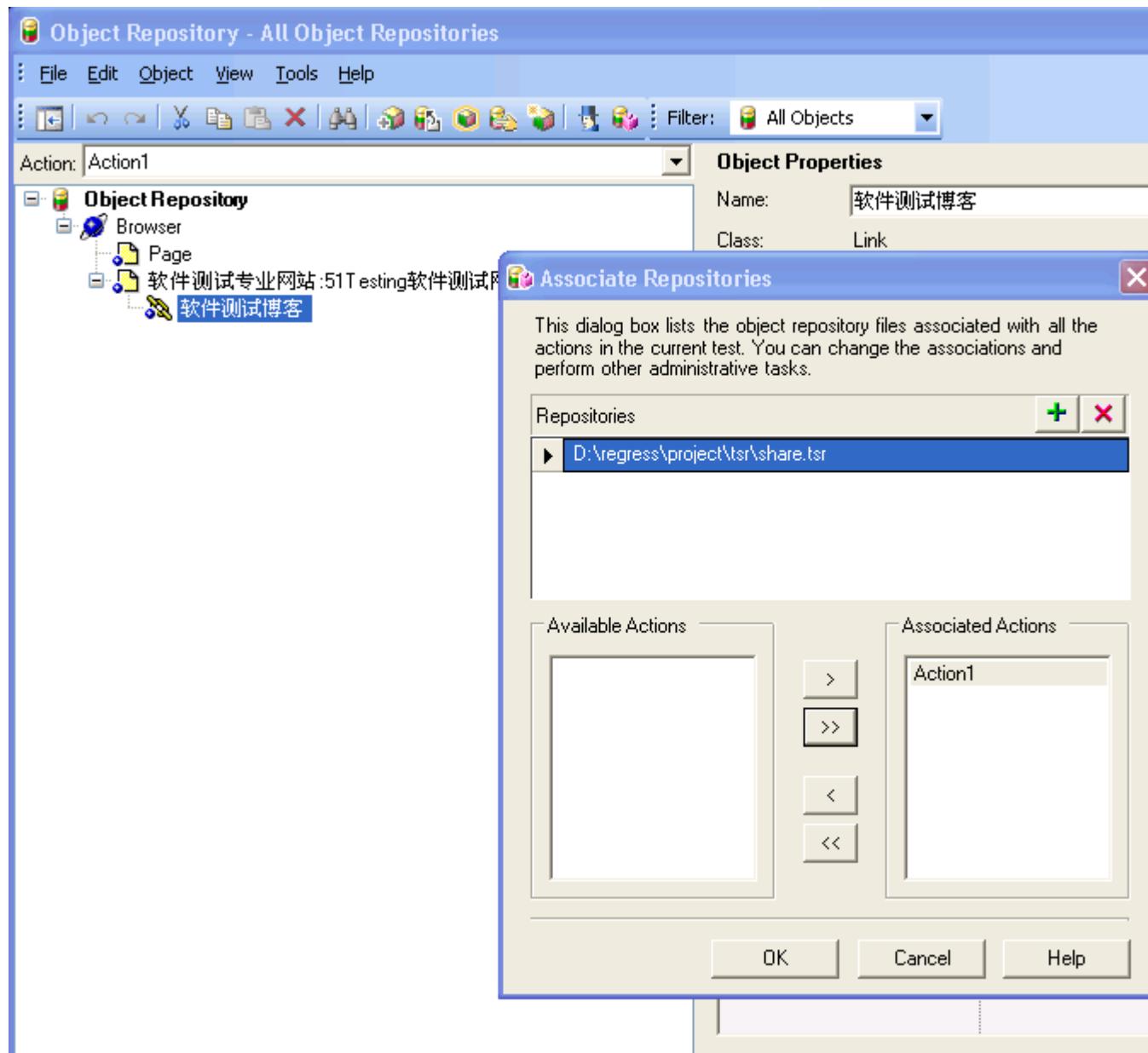
- 首先录制一个脚本
- 打开 Object Repository, 点击 Associate Repositories 按钮, 弹出如图一对话框

自动化测试	Version: 1.3
文档	2006-5-17



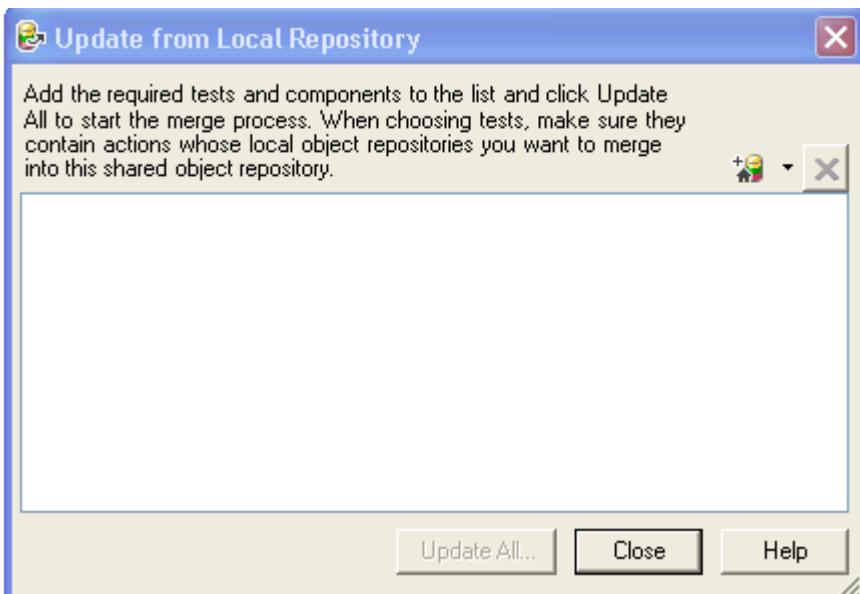
- 点击 "+" 按钮, 添加共享.tsr 文件, 如图 2

自动化测试	Version: 1.3
文档	2006-5-17

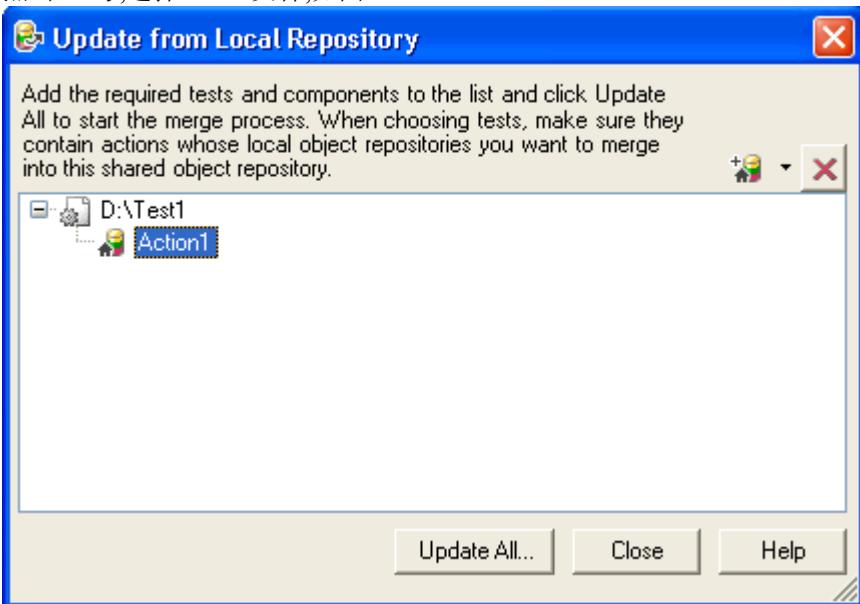


- 点击 ok 按钮,并关闭 Object Repository 对话框,保存该脚本名称为 test1,并点击 New 按钮.
- 打开 Resources-> Object Repository manager...
- 打开 Share.tsr 文件,(在打开的文件对话框中去掉"Open in read-only mode"的复选框,这样打开的文件是可编辑的)
- 点击 tools->Update from local repository...,弹出如图三

自动化测试	Version:	1.3
文档		2006-5-17



- 点击 "+" 号, 选择 "test1" 文件, 如图 4

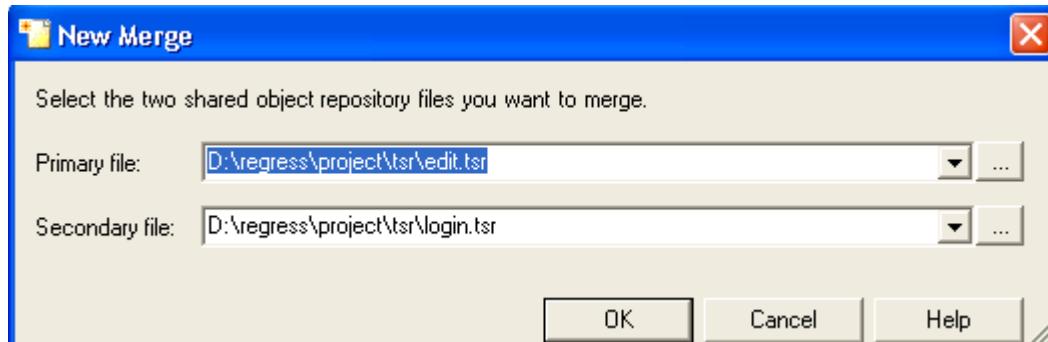


- 点击 Update All... 按钮, 余下的操作就按照提示操作, 你再打开 "test1" 脚本的 tsr 文件, 可以发现这时就把 Share.tsr 文件合上去了

2) 合并两个 tsr 文件

合并两个 tsr 文件比较简单, 打开 resources->Repository Object Manager..., 点击 tool->object repository merge tool 的对话框如图五:

自动化测试	Version:	1.3
文档		2006-5-17



点击 Ok 就合并成功了,期间可能有冲突,这里就不作介绍冲突的解决

3) 在 Share 文件中添加对象

打开 Object Repository manager..., 打开一个要编辑的 tsr 文件, 使成为可编辑状态, 打开要添加对象的页面, 点击 Add Obj 钮, 这时鼠标就会变成手状的模样, 点击要添加的控件, 这时这个控件的属性就添加到了对象库中了.

如果这个对象在另一个对象库中已经存在, 那么也可以打开另外一个对象, 选择那个对象, 点击鼠标右键, 在弹出菜单中点再选择目的对象库的目的节点, 点击粘贴, 这样就把对象复制到另外对象库中去了.

原始链接: http://blog.51testing.com/?34594/action_viewspace_itemid_4166.html

47. change sysdate

```
Dim oShell
Set oShell = CreateObject ("WSCript.shell")
oShell.run "cmd /K date 07-02-02"
Set oShell = Nothing
试一下行不行
```

48. Import Excel

共享一下使用 Import Excel 的体会

以下其实是一个很微不足道的问题,对于非新手没什么意义,但是我共享出来,希望跟我一样的新手看了之后,有所领会,以后少走这方面的弯路

希望大家把实际操作过程中解决的问题共享出来,共同进步^_^

在使用 Import Excel 的过程中,遇到以下问题:明明从外面导进来的 Excel 表格中有多行不同的数据,导入到 QTP 中正确,但是就是只执行第一行的代码,起初感到非常奇怪.代码如下(Action2):

```
datatable.Import("c:\a.xls")
```

自动化测试	Version:	1.3
文档		2006-5-17

```

Window("Flight Reservation").WinMenu("Menu").Select "File;Open Order..."
Window("Flight Reservation").Dialog("Open Order").WinCheckBox("Customer Name").Check CheckPoint("Customer Name")
Window("Flight Reservation").Dialog("Open Order").WinCheckBox("Order No.").Set "ON"
Window("Flight Reservation").Dialog("Open Order").WinCheckBox("Order No.").Check CheckPoint("Order No._4")
Window("Flight Reservation").Dialog("Open Order").WinEdit("Edit").Check CheckPoint("Edit_2")
Window("Flight Reservation").Dialog("Open Order").WinCheckBox("Order No.").Set "ON"
Window("Flight Reservation").Dialog("Open Order").WinEdit("Edit").Set DataTable("OrderNo_p", dtLocalSheet)
DataTable.GetSheet("Copy of Action1_2").SetNextRow
Window("Flight Reservation").Dialog("Open Order").WinButton("OK").Click
Window("Flight Reservation").WinButton("Update Order").Click
Window("Flight Reservation").Activate
该代码运行的结果就是在测试结果中只显示第一行的数据,其他行数据不执行,仔细推敲,把 datatable.Import("c:\a.xls")
放到包含这段代码的 Action 这前,测试结果如预期所要
终于明白了原因:由于对 Action2(以上代码)进行了参数化,并设置了"Run on all rows",进行的是多次执行,这种情况下,把
datatable.Import("c:\a.xls")放到同一个 action 中,则每一次都是从头往 a.xls 中读取初始数据,取的必定是第一行数据(虽然在后面加了 SetNextRow 来读取下一行,但是重新读取表格使得 setNextRow 失去了实际作用),如果放在这个 action 之前,则只读入 a.xls 一次,加上执行过程中有了 SetNextRow 读取下一行,这样就如预期结果了.

```

49. Call Dll

我自己写的 dll 无非就是做个加法,减法输出一下,呵呵,主要是我对 VC++ 不熟悉,拿不出手~ 😊
所以昨天实验了一些 VB 的 API 函数,用的都是系统中的 DLL 文件,比如:Kernel32.dll,user32.dll,GDI32.dll 等等,然后调用里面的一些函数实现一些功能,函数我没有多少,现在发一些我有的给大家分享一下吧~!

FindFirstFile

```
Declare Function FindFirstFile Lib "kernel32" Alias "FindFirstFileA" (ByVal lpFileName As String, lpFindFileData As
```

WIN32_FIND_DATA) As Long

说明:

根据文件名查找文件

返回值:

Long, 如执行成功, 返回一个搜索句柄。如果出错, 返回一个 INVALID_HANDLE_VALUE 常数, 一旦不再需要, 应该用 FindClose 函数关闭这个句柄

参数 类型及说明:

lpFileName String, 欲搜索的文件名。可包含通配符, 并可包含一个路径或相对路径名

lpFindFileData WIN32_FIND_DATA, 这个结构用于装载与找到的文件有关的信息。该结构可用于后续的搜索

注解

由这个函数返回的句柄可以作为一个参数用于 FindNextFile 函数。这样一来, 就可以方便的枚举出与 lpFileName 参数指定的文件名相符的所有文件

FindWindow

```
Declare Function FindWindow Lib "user32" Alias "FindWindowA" (ByVal lpClassName As String, ByVal lpWindowName
```

As String) As Long

说明:

自动化测试	Version:	1.3
文档		2006-5-17

寻找窗口列表中第一个符合指定条件的顶级窗口（在 **vb** 里使用：**FindWindow** 最常见的一个用途是获得 **ThunderRTMain** 类的隐藏窗口的句柄；该类是所有运行中 **vb** 执行程序的一部分。获得句柄后，可用 **api** 函数 **GetWindowText** 取得这个窗口的名称；该名也是应用程序的标题）

返回值：

Long, 找到窗口的句柄。如未找到相符窗口，则返回零。会设置 **GetLastError**

参数 类型及说明：

lpClassName String, 指向包含了窗口类名的空中止（C 语言）字串的指针；或设为零，表示接收任何类

lpWindowName String, 指向包含了窗口文本（或标签）的空中止（C 语言）字串的指针；或设为零，表示接收任何窗口标题

注解

很少要求同时按类与窗口名搜索。为向自己不准备参数传递一个零，最简便的办法是传递 **vbNullString** 常数

SetActiveWindow

Declare Function SetActiveWindow Lib "user32" Alias "SetActiveWindow" (ByVal hwnd As Long) As Long

说明：

激活指定的窗口

返回值：

Long, 前一个活动窗口的句柄

参数 类型及说明：

hwnd Long, 待激活窗口的句柄

注解

在 **vb** 里使用这个函数要小心，它不会改变输入焦点，所以焦点可能设向一个不活动窗口，最好换用 **SetFocusAPI** 函数来激活窗口。如指定的窗口不从属于当前输入线程，则没有任何效果

SetWindowText

Declare Function SetWindowText Lib "user32" Alias "SetWindowTextA" (ByVal hwnd As Long, ByVal lpString As String)

As Long

说明：

设置窗口的标题文字或控件的内容（在 **vb** 里使用：针对 **vb** 窗体，应使用 **caption** 或 **text** 属性）

返回值：

Long, 非零表示成功，零表示失败。会设置 **GetLastError**

参数 类型及说明：

hwnd Long, 要设置文字的窗口的句柄

lpString String, 要设到 **hwnd** 窗口中的文字

ExitProcess

Declare Sub ExitProcess Lib "kernel32" Alias "ExitProcess" (ByVal uExitCode As Long)

说明：

中止一个进程

参数 类型及说明：

uExitCode Long, 指定想中断的那个进程的一个退出代码

GetProcessTimes

Declare Function GetProcessTimes Lib "kernel32" Alias "GetProcessTimes" (ByVal hProcess As Long, lpCreationTime As FILETIME, lpExitTime As FILETIME, lpKernelTime As FILETIME, lpUserTime As FILETIME) As Long

说明：

获取与一个进程的经过时间有关的信息

返回值：

Long, 非零表示成功，零表示失败。会设置 **GetLastError**

参数 类型及说明：

hProcess Long, 一个进程句柄

自动化测试	Version:	1.3
文档		2006-5-17

lpCreationTime FILETIME, 指定一个 FILETIME 结构, 在其中装载进程的创建时间

lpExitTime FILETIME, 指定一个 FILETIME 结构, 在其中装载进程的中止时间

lpKernelTime FILETIME, 指定一个 FILETIME 结构, 在其中装载进程花在内核模式上的总时间

lpUserTime FILETIME, 指定一个 FILETIME 结构, 在其中装载进程花在用户模式上的总时间

适用平台

Windows NT

LoadModule

Declare Function LoadModule Lib "kernel32" Alias "LoadModule" (ByVal lpModuleName As String, lpParameterBlock As Any) As Long

说明:

载入一个 windows 应用程序, 并在指定的环境中运行

返回值:

Long, 大于 32 表示成功, 请参考 FindExecutable 函数的返回值

参数 类型及说明:

lpModuleName String, 要装载的可执行程序的文件名

lpParameterBlock Any, 指定一个结构, 用它定义装载新应用程序时使用的参数

50. Call Function

按照 VBScript 的语法要求, 在 UltraEdit 里面 编写函数的内容, 然后保存为 function.vbs, 并把它保存在你的电脑上, 假如它在 d: 盘根目录下面。你在需要调用此函数脚本的最前面 写上 Execute("d:\function.vbs"), 然后像调用一般的函数那样调用它就可以了。

感谢 walker1020, 我试过了, 可以的, 但是在 QTP8.2 中调用外部定义的函数是使用 ExecuteFile 方法, 例如: ExecuteFile "D:\Function.vbs"。

22 get data

其实对 **DataTable** 的操作可以很灵活.....

由 QTP 自动生成的语句是这样的:

DataTable("Col", dtLocalSheet)

DataTable("Col", dtGlobalSheet)

其中, dtLocalSheet 就是当前 Action 的 DataTable, dtGlobalSheet 就是 Global 的 DataTable, Col 是列名

还有很多写法, 比如

DataTable("Col", "Action2")

就是读取 Action2 的 Col 列。

你完全可以读取另一个 Action 的 DataTable, 而不需要跨 Action 传递变量, 使用 DataTable 比变量更方便, 因为结束后能在 Result 里看到运行时的值

你可以在一个 Action 里读取另外一个 Action 的列, 但是要注意另外一个 Action 的当前行

比如你在 Action1 里读取 Action2 的某列, 如果 Action1 运行到第二行, 你读取的 Action2 还是第一行的数据, 解决办法就是写上这句:

DataTable.GetSheet("Action2").SetCurrentRow(2)

自动化测试	Version:	1.3
文档		2006-5-17

你也可以用 GetCurrentRow 来获取 Action1 的行，然后再用 SetCurrentRow 来保持两个 Action 的当前行一致：

```
CurrRow = DataTable.GetSheet("Action1").GetCurrentRow
DataTable.GetSheet("Action2").SetCurrentRow(CurrRow)
```

还有一种写法：

```
DataTable(1, "Action2")
```

这样就是读取 Action2 的第一列，不管第一列叫什么名字，都能读

这样就很方便，比如：

```
For i = 1 To 10
    MsgBox DataTable(i, "Action2")
```

Next

这样就能循环读取 Action2 的 1~10 列了

同样，表也可以用数字代替：

```
DataTable(1, 2)
```

这样就是读取第二个 Action 表的第一列

51. Global and local 区别

global 和 local 的区别

global 是全局的!有几行数据程序就要回放几次!!不能重新设置!!

local 是局部的!有几行数据 action 就要回访几次!!

设置 action 的重复次数的操作:反击 action,选择 action call property,即可进行选择.

下面有几种情况：

1.当 global 有不止一行的数据,action call property->Run On all Rows

程序每次运行时,action 中的每行都要执行

2.当 global 有不止一行的数据,action call property->Run one iteration only

而且 global 的行数>action 的行数,当 action 执行到最后一行后,不管此时 global 的行数为几,下次回放时 action 都执行最后一行!

如果 global 的行数<action 的行数,action 就执行不到最后一行

3.action call property->Run from rows to rows,就结合上面所说的理解吧!!

希望能给大家帮上忙,帮我顶一下!!第一次回答问题!

52. Shell

我要录制 ALT+F4,为什么执行到 shell.sendkeys "%{F4}",不是关闭窗口,而是出现其他菜单

```
Set shell=Createobject("WScript.Shell")
    shell.sendkeys "%{F4}"
```

自动化测试	Version:	1.3
文档		2006-5-17

53. 对文件系统操作

对系统文件的操作

CODE:

```
11.1 从系统的文件中获取信息及删除文件
get_file_infor("c:\she.mpg")
function get_file_infor(url)
dim fso,f
set fso=createobject("scripting.filesystemobject")
set f=fso.getfile(url)
f.name:f.size:f.type:f.datatime//获取文件信息
fso.deletefile(url)////删除文件
end function

11.2 获取文件夹里所有文件信息
get_folder_infor("c:\kai")
function get_folder_infor(folder)
dim fso,f,f1,n
set fso=createobject("scripting.filesystemobject")
set f=fso.getfolder(folder)
set fc=f.files
for each f1 in fc
select case f1.name
case "kai.mpg","she.mpg","dd.mp3"//检查文件夹里是否含有这些文件
end select
next
end function
```

54. 对象同步

等待某个对象出现方法

CODE:

```
y=.....waitproperty("visible",true,10000)
```

55. 正则表达式应用模板

正则表达式应用模板

CODE:

```
进行日期 YYYY-MM-DD 的格式检查 ：
Function RegExpTest(patrn, strng)
Dim regEx, Match, Matches      ' Create variable.
Set regEx = New RegExp          ' Create a regular expression.
```

自动化测试	Version:	1.3
文档		2006-5-17

```

regEx.Pattern = patrn      ' Set pattern.
regEx.IgnoreCase = True    ' Set case insensitivity.
regEx.Global = True        ' Set global applicability.
Set Matches = regEx.Execute(strng) ' Execute search.
For Each Match in Matches   ' Iterate Matches collection.
    RetStr = RetStr & "Match found at position "
    RetStr = RetStr & Match.FirstIndex & ". Match Value is "
    RetStr = RetStr & Match.Value & vbCrLf
Next
RegExpTest = RetStr
End Function
date_pattern="^(((19|20)([02468][048])|([13579][26]))-02-29))|((20[0-9][0-9])|(19[0-9][0-9]))-(((0[1-9])|(1[0-2]))-((0[1-9])|(1\d)|(2[0-8])))|(((0[13578])|(1[02]))-31)|(((01,3-9)|(1[0-2]))-(29|30))))$"
result_message=RegExpTest(date_pattern, inputbox("请你输入要检查的时间: "))'用其它正则表达式更改此处
Select case result_message
Case ""
    msgbox("你输入的日期格式与标准不匹配")
case else MsgBox(result_message)
end select

```

返回一个字符串在另一字符串中的位置

CODE:

```
instr(string1,string2)
```

56. Wscript.shell 的应用

WScript.Shell的一些应用

CODE:

```

set WshShell =CreateObject("WScript.Shell")
WshShell.SendKeys "{ENTER}"          '模拟键盘进行操作
WshShell.AppActivate "Calculator"    '启动应用程序

```

57. Create log file

Create Log File:

CODE:

```

Dim LOGFile, fso, MyFile

LOGFile="C:\Log.txt"
Set fso = CreateObject("Scripting.FileSystemObject")

```

自动化测试		Page 128
-------	--	----------

自动化测试	Version:	1.3
文档		2006-5-17

```
If fso.FileExists(LOGFile) = False Then
    Set MyFile = fso.CreateTextFile(LOGFile, True)
    MyFile.Close
end if
Set MyFile = fso.OpenTextFile(LOGFile, 8, True)
MyFile.WriteLine("")
MyFile.WriteLine(" " & Cstr(Now) & " -----")
MyFile.WriteLine("LOG Information!")
MyFile.Close
```

补充几点

58. 对象同步

插入同步等待某个对象出现

```
Dim continue
continue = False

While not continue
    wait(1)
    continue = Window("航班预订").Exist
Wend
```

59. 异常控制

异常控制

CODE:

```
If Window("航班预订").Dialog("航班预订").Exist Then
    Set myStatic = Description.Create()
    myStatic("nativeclass").Value="Static"
    Set myObject=Window("航班预订").Dialog("航班预订").ChildObjects(myStatic)
    ErrInfo = myObject(0).GetRoProperty("text")
    Reporter.ReportEvent micFail,"时间输入", ErrInfo
    Window("航班预订").Dialog("航班预订").WinButton("确定").Click
    ExitActionIteration

End If
```

自动化测试	Version:	1.3
文档		2006-5-17

60. Check if the object.exist

校验对象是否存在 object.exist

文章出处: 51testing 博客 作者: qixafeng 发布时间: 2007-03-09

除了用校验点来确认相符的对象是否存在于外, 也可以用 object 的 exist 属性来判断对象的存在与否.

语法: **Object.Exist([Timeout])**

描述:

Argument	Type	Description
<i>Timeout</i>	Number	<p>Optional. The length of time to search for the object before returning a True or False value.</p> <ul style="list-style-type: none"> • If a timeout value is specified, QuickTest waits until it finds the object or until the timeout is reached. • If the value 0 is specified, the method returns the True or False value immediately. • If no value is specified, the value specified in the Test Settings dialog box for the Object Synchronization Timeout is used for tests. For business components, the pre-defined value of 20 seconds is used. <p>For more information on the Object Synchronization Timeout, refer to the <i>QuickTest Professional User's Guide</i>.</p>

实例:

```

Dialog("Login").Activate
Dialog("Login").WinEdit("Agent Name:").Set "training"
Dialog("Login").WinEdit("Agent Name:").Type micTab
if Dialog("Login").WinEdit("Password:").Exist Then
  Reporter.ReportEvent micDone, "存在", "控件存在"
  Dialog("Login").WinEdit("Password:").SetSecure
  "429376f06698f739df8bb1c09eab3d78dee316f7"
else
  Reporter.ReportEvent micDone, "不存在", "控件不存在"
End If

```

原始链接: http://blog.51testing.com/?61184/action_viewspace_itemid_463

自动化测试	Version:	1.3
文档		2006-5-17

[2.html](#)

61. QTP 场景恢复

QTP Recovery Scenario 简介

文章出处: 51testing 博客 作者: 阳阳 发布时间: 2006-12-20

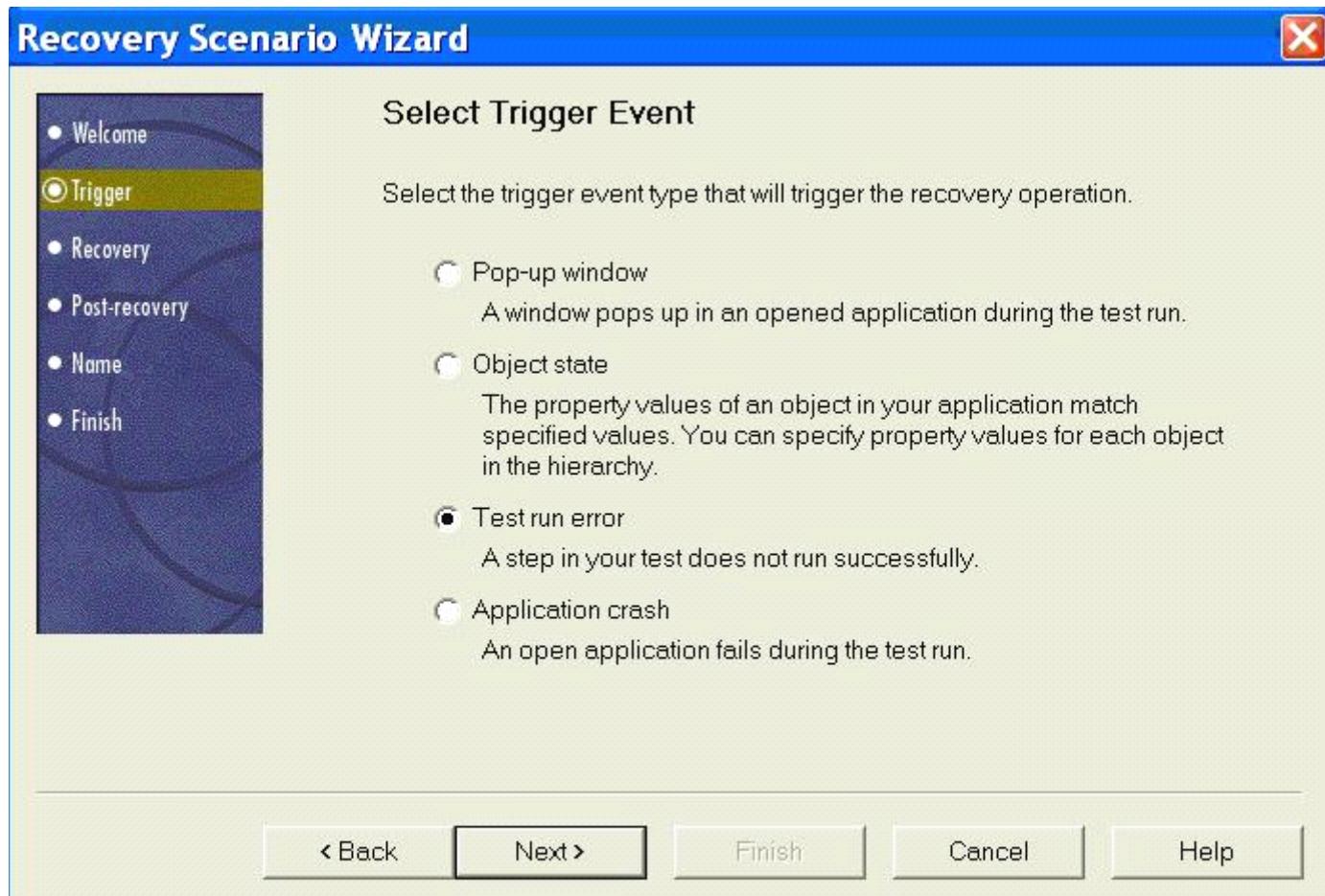
场景恢复可以用于应对测试脚本在运行的过程中出现的异常，在预估可能出现的异常状况下，添加对应的场景恢复，可以使脚本运行的更加通畅。以下是对 QTP Recovery Scenario 的简单介绍，希望对初学者有所帮助。

添加一个新的场景恢复，通过菜单 Tools->Recovery Scenario Manager 进入，主要分为以下四个步骤。

[步骤一: Trigger]

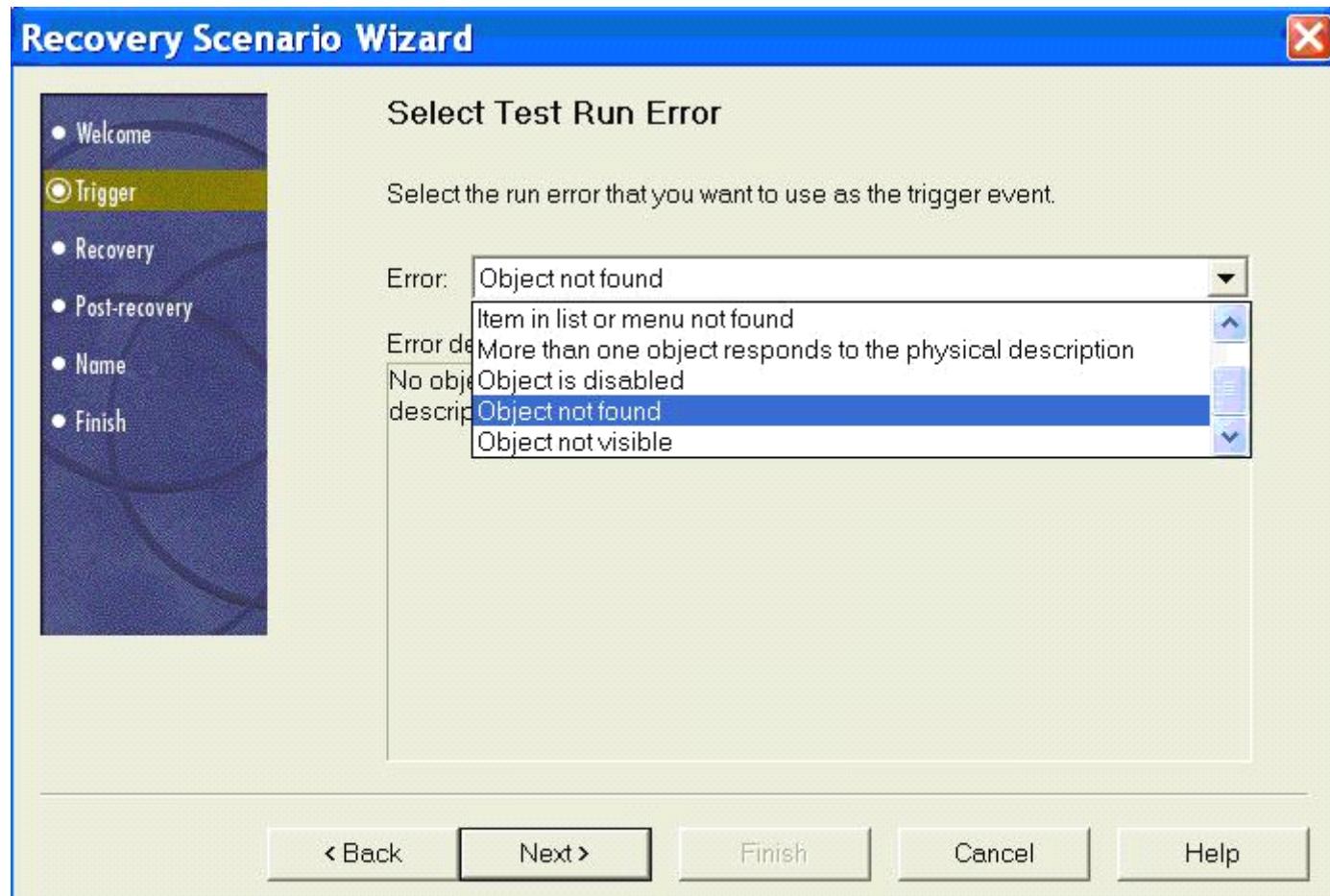
场景恢复机制提供了四种类型的触发事件，分别用来识别：弹出对话框、对象的特殊属性值、运行错误、应用程序失败。

自动化测试	Version:	1.3
文档		2006-5-17



可以根据具体的需求来添加各个类型的恢复场景，每种类型的选项可以在添加向导中选择，如下图。

自动化测试	Version: 1.3
文档	2006-5-17

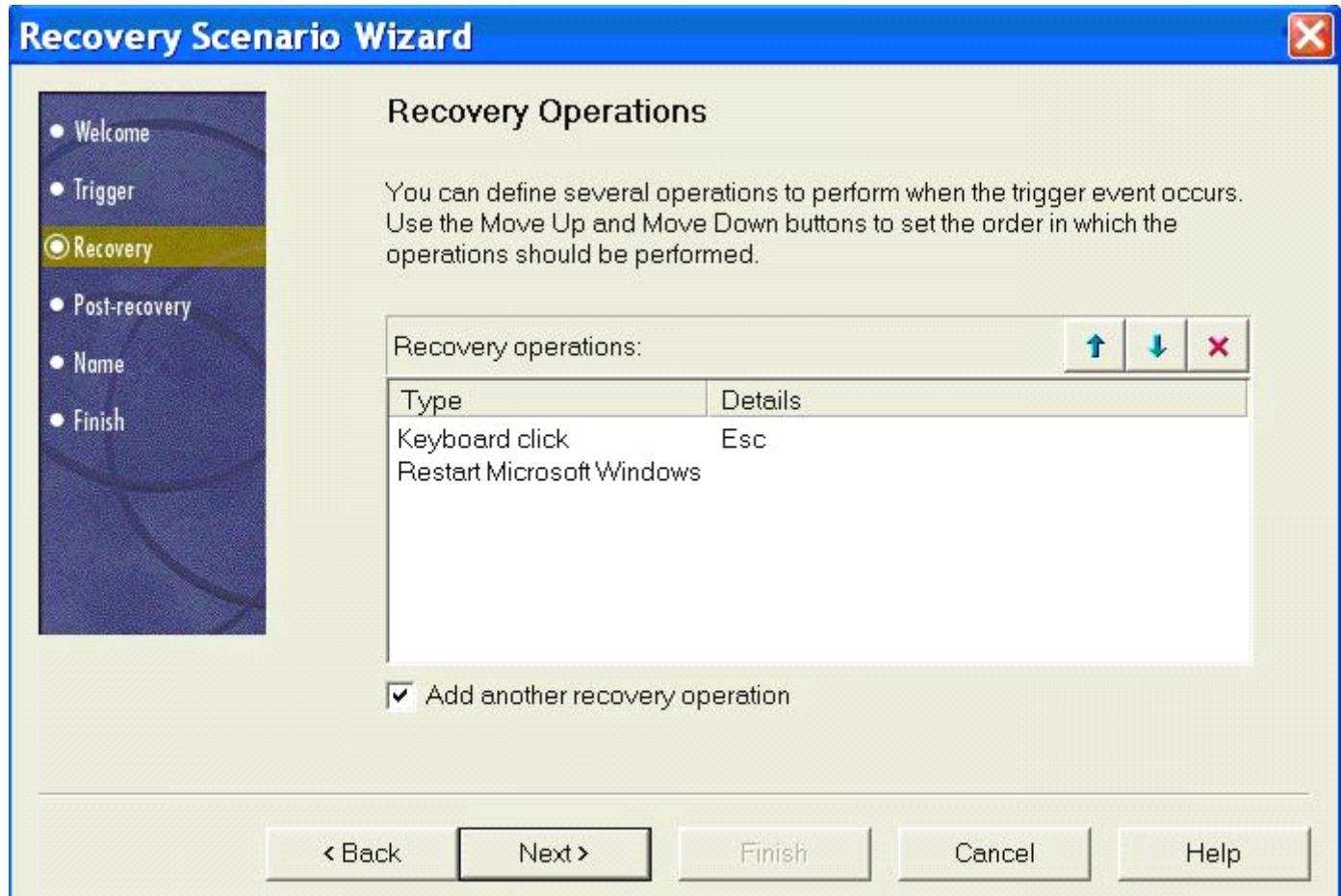


[步骤二: Recovery]

恢复的操作可以是自定义按钮操作，函数调用，关闭应用程序进程，重起机器等等，几乎涵盖了所有 QTP 的正常操作，按照向导进行设置，操作很方便。也可以添加多个恢复操作，且调整执行顺序，注意：重起系统总是排在最后一个。

添加操作完成，把 Add another recovery operation 前的按钮去掉，才可以进入下一步。

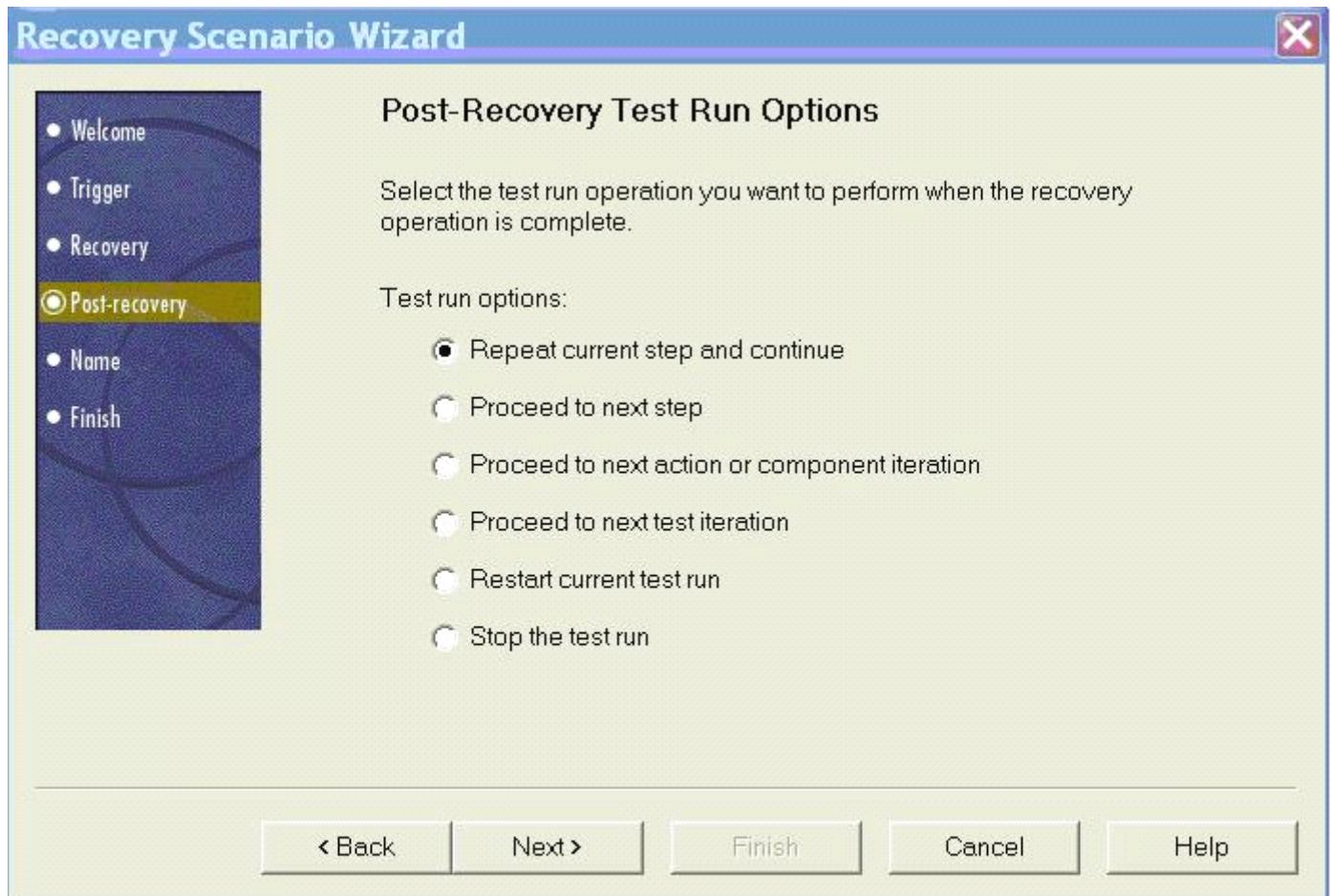
自动化测试	Version: 1.3
文档	2006-5-17



[步骤三: Post-recovery]

在 Post-recovery 中，又分了 6 个选项：

自动化测试	Version:	1.3
文档		2006-5-17



Repeat current step and continue: 重复当前步骤然后继续向下

Proceed to next step: 处理下一步

Proceed to next action or component iteration: 处理下一个 Action, 或者组件的下一个循环

Proceed to next test iteration: 处理该测试的下一个循环

Restart current test run: 重新启动当前的测试

Stop the test run: 终止测试运行

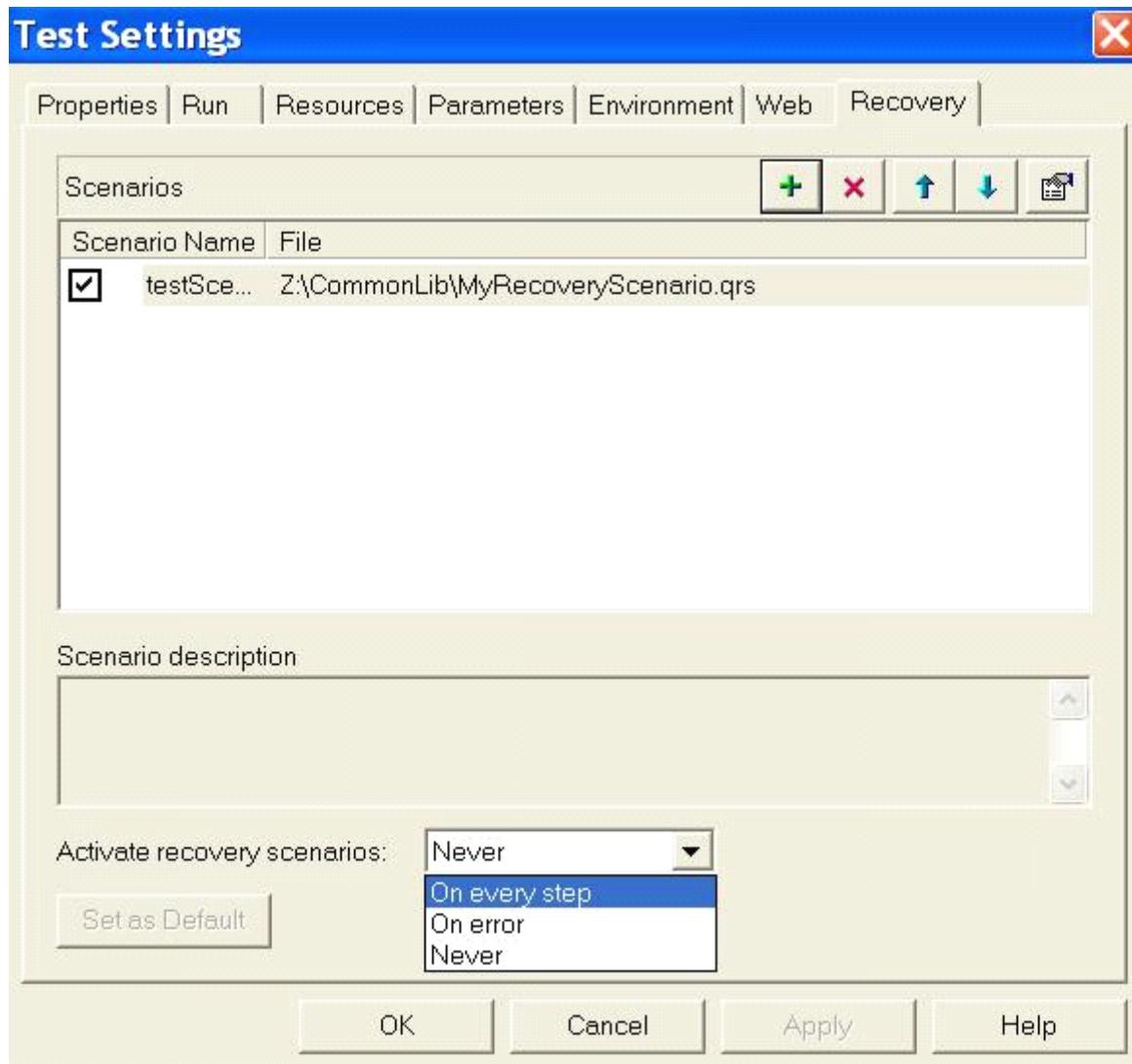
这些操作的差别在具体实践中体会吧, 这里就不赘述了。

[步骤四: Name]

一切都设置好以后, 就给你的场景恢复起个唯一的名字吧。也可以添加一些描述方便维护工作。

开始使用场景恢复, 在 Test->Settings->Recovery 中添加设置好的恢复场景, 然后选择激活方式 (On every step/On error/Never)。

自动化测试	Version:	1.3
文档		2006-5-17



On every step, 只要出现恢复的场景，就执行场景恢复中的动作，然后继续。

On error, 在出现错误的时候，才查找是否符合待恢复的场景，如果是则执行恢复操作。

Never, 无论如何，都不运行场景恢复机制。

提供两个执行场景恢复的小例子：

例 1：

[配置]

1, 在 Recovery Scenario Properties 中, Post-Recovery Operation 选择 Repeat current step and continue

2, 激活方式为 On every step

[现象]

只要出现恢复的场景，就执行场景的动作，并且重复执行出现该场景的那条语句，然后继续。适用于重复尝试必将成功的场景，否则死循环。

例 2：

自动化测试	Version:	1.3
文档		2006-5-17

[配置]

- 1, 在 Recovery Scenario Properties 中, Post-Recovery Operation 选择 Repeat current step and continue
- 2, 激活方式为 On error

[现象]

出现错误后, 进入场景恢复机制, 就执行场景的动作, (并没有重复执行出现该场景的那条语句, 不知道为什么) 直接执行下一条语句。

原始链接:http://blog.51testing.com/?8181/action_viewspace_itemid_1589.html

62. QTP 步骤生成器

QTP 中的步骤生成器

文章出处: 51testing 博客 作者: xiaonan 发布时间: 2006-12-26

在我们录制脚本过程中, 可能会遇到这样一种情况. 用 object spy 能识别到某个对象, 但录制的时候脚本就是没生成下来. 当然如果你对 QTP 的原理比较熟悉, 可以直接把对象加到对象库中, 再自己手工写代码. 这当然是个比较好的解决方法. 但碰到对 QTP 不是很熟, 对 vbscript 的语法比较陌生的朋友来说, 可能需要花点功夫了. 但事实上 QTP 还提供给我们一个方便操作的工具, 叫做步骤生成器. 在 insert 菜单里能够找到, 当然按一下 F 能直接调出这个工具.

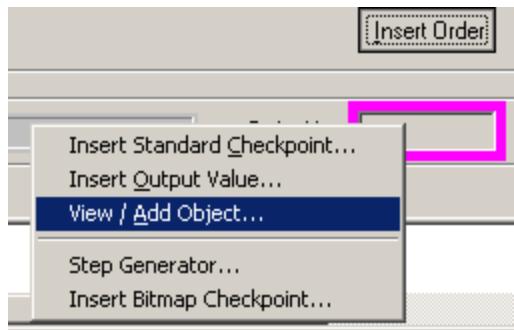
先来介绍一下这个工具. “步骤生成器”对话框有助于快速轻松地添加一些步骤, 这些步骤使用测试对象方法、实用程序对象方法和函数调用, 因此用户无需记住语法或者精通高级 VBscript. 既可在关键字视图中, 也可在专家视图中使用步骤生成器.

接下来用一个小例子来说明这个步骤生成器的使用.

The screenshot shows the 'Step Generator' dialog box from QTP. It contains fields for Flight No (20288), Departure Time (08:12 AM), Arrival Time (03:23 PM), Airline (AA), Name (51testing), Class (Business selected), Tickets (2), Price (\$224.40), and Total (\$448.80). At the bottom are buttons for Update Order, Delete Order, and Insert Order. A text input field at the bottom is highlighted with a pink rectangle. The title bar of the dialog box says 'Step Generator'.

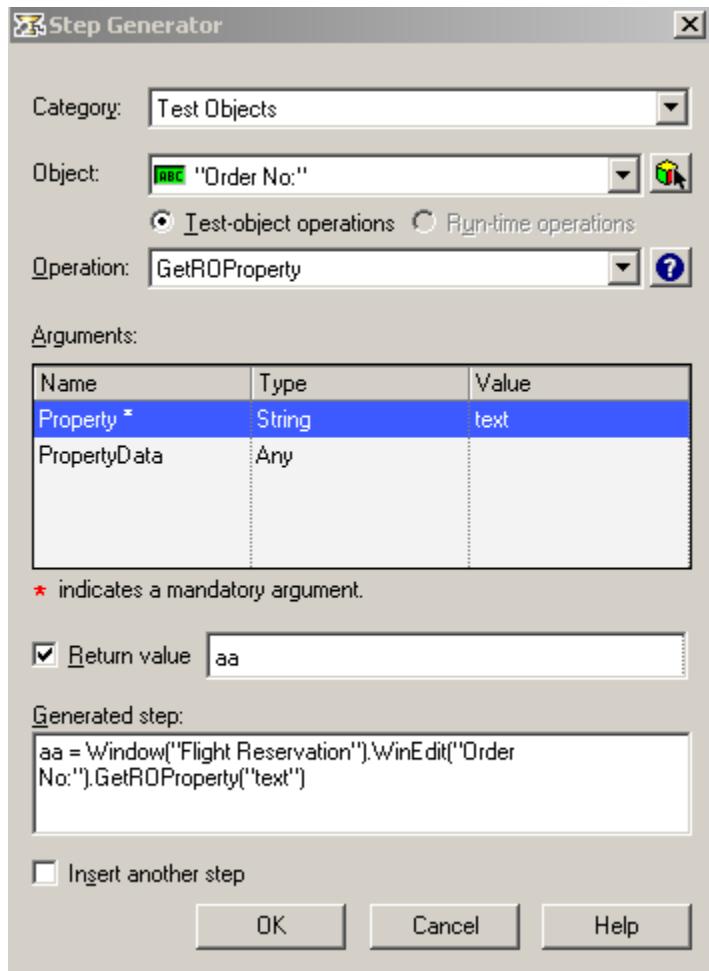
自动化测试	Version:	1.3
文档		2006-5-17

比如在 QTP 自带的 fight 程序里, 在填好相应信息后, 按下 insert order 会生成一个订单号. 那么我们就用步骤生成器生成脚本来取这个 order no. 首先我们要做的是把这个对象加入到对象库中去. 加入对象的方法, 可以打开对象库来 add 填加. 也可以在 Active screen 里找到这个对象来右击鼠标, 选择 ADD object. 如下:



在填出的窗口后, 按下 ok, 然后下一个窗口继续按 add, 直到把对象真正加入到对象库中去. 因为这边有些朋友以为第一次按了 ok 就已经填进去了, 下面就直接 close 掉了, 其实不是这样子的, 这边是要注意的. 对象我们已经准备好后, 我们就在合适的地方生成需要的脚本. 直接按一下 F7. 然后打开的了步骤生成器的窗口. 如下:

自动化测试	Version: 1.3
文档	2006-5-17



在第一个下拉选项 Category 我们选择 test object. 在第二个下拉选项 object 里选择我们要操作的对象，我们这里选择 order no. 也可以按旁边的图标，会显示这个脚本中的所有对象让你选择。然后在第三个下拉选项 operation 里选择我们要做的操作。这里提供了很多的操作。这里只是取值，所以选择了 GetRoProperty.

然后在 Arguments 里在 value 里填上我们要取的属性名。在 Return value 打上勾，设置返回值取好变量名。再按下 ok 整句代码就出来了。

```
aa = Window("Flight Reservation").WinEdit("Order No:").GetROProperty("text")
```

到这里整个过程就结束了。大家可以自己试试看。

原始链

接:http://blog.51testing.com/?26649/action_viewspace_itemid_1808.html

自动化测试	Version:	1.3
文档		2006-5-17

63. IFRAME issues

QuickTestPro 处理带有 IFRAME 的问题

文章出处: 51tesing 博客 作者: pcl 发布时间: 2006-07-19

问题原文

“我的 web 页最下面一块是引入 iframe 的版权页面. 导致我动态读取不到 iframe 外的其它链接。

我的动态读取语句如下：

```
set oDesc = Description.Create()
oDesc("Class Name").value="Link"
set objColl = Browser("供应商").Page("供应商").ChildObjects(oDesc)
NumberofLists =ObjectColl.Count
-----页面上本来有十几个 link 的, 但因为页面下面有 iframe 的原因, 跟踪以上语句发现读取的是 iframe 里的
link, 始终读取不到 iframe 外的 link , 请问如何解决?
”
```

这道问题核心是取得所有的页面链接然后进行处理, 但是由于其中包含了 iframe 导致利用集合无法得到页面中的所有链接 (link) 对象。为了解决这个问题, 我自己模拟了一个环境, 写了带有 iframe 的页面, 然后利用 qtp 进行处理。

html 代码如下：

1. 带有 iframe 的 html 代码

```
<html>
<body>
<p>这是一个测试</p>

<p>
<a href="com">com</a>
</p>

<p>
<a href="nethttp://edu.51testing.net">net</a>
</p>

<iframe src="包含的页面路径\url.html" width=100% name=comment frameborder=0
3onload="document.all['comment'].style.height=comment.document.body.scrollHeight"></iframe>

</body>
</html>
```

2. iframe 包含的页面文件 Html 代码 - url.html

自动化测试	Version:	1.3
文档		2006-5-17

```

<html>
<body>

<p>
<a href="sohuhttp://www.sohu.com">sohu</a>
</p>

<p>
<a href="hao123http://www.hao123.com">hao123</a>
</p>

</body>
</html>

```

qtp 代码如下：枚举所有的页面 link

```

Dim objPage
Dim objTag
Dim intTotalLink
Dim intCtr
Dim strLink
Dim strHref

Dim oDesc
Dim NumberOfiframe
' 枚举网页中除了带有 iframe 里所有外部的链接
' #####
Set objPage = Browser("Browser").Page("Page").Object
Set objTag = objPage.GetElementsByTagName("A")
intTotalLink = objTag.Length - 1

For intCtr = 0 to intTotalLink
    strLink = objTag(intCtr).InnerText
    strHref = objTag(intCtr).Href
    msgbox strLink
    msgbox strHref
Next
' #####
' 枚举 iframe 内的所有链接
Set oDesc = Description.Create()
oDesc("html tag").Value = "iframe"
Set iframe = Browser("Browser").Page("Page").ChildObjects(oDesc)
NumberOfiframe = iframe.Count()
For i = 0 To NumberOfiframe - 1
    Set objPage = iframe(i).Object
    Set objTag = objPage.GetElementsByTagName("A")
    intTotalLink = objTag.Length - 1

```

自动化测试	Version:	1.3
文档		2006-5-17

```

For intCtr = 0 to intTotalLink
    strLink = objTag(intCtr).InnerText
    strHref = objTag(intCtr).Href
    msgbox strLink
    msgbox strHref
    Next
Next

Set oDesc=nothing
' #####

```

以上代码在中 Window2000ad sp4+qtp8.2 上调试通过。

备注：

解决这个问题的方法很多，在 51testing cpe 工具使用培训的课程中就提到了解决这个问题的技术，我还写了一套代码同样解决这个问题。

原稿出处：

本文初衷为解决论坛中提到的问题，链接：

<http://bbs.51testing.com/viewthread.php?tid=35723&extra=page%3D1>

问题原文

“我的 web 页最下面一块是引入 iframe 的版权页面. 导致我动态读取不到 iframe 外的其它链接。

我的动态读取语句如下：

```

set oDesc = Description.Create()
oDesc("Class Name").value="Link"
set objColl = Browser("供应商").Page("供应商").ChildObjects(oDesc)
NumberofLists = ObjectColl.Count()
-----页面上本来有十几个 link 的，但因为页面下面有 iframe 的原因，跟踪以上语句发现读取的是 iframe 里的
link，始终读取不到 iframe 外的 link，请问如何解决？
"
```

这道问题核心是取得所有的页面链接然后进行处理，但是由于其中包含了 iframe 导致利用集合无法得到页面中的所有链接（link）对象。为了解决这个问题，我自己模拟了一个环境，写了带有 iframe 的页面，然后利用 qtp 进行处理。

html 代码如下：

1. 带有 iframe 的 html 代码

```

<html>
<body>
<p>这是一个测试</p>

```

自动化测试	Version:	1.3
文档		2006-5-17

```

<p>
<a href="com">com</a>
</p>

<p>
<a href="nethttp://edu.51testing.net">net</a>
</p>

<iframe src="包含的页面路径\url.html" width=100% name=comment frameborder=0
3onload="document.all['comment'].style.height=comment.document.body.scrollHeight"></iframe>

</body>
</html>

```

2. iframe 包含的页面文件 Html 代码 - url.html

```

<html>
<body>

<p>
<a href="sohuhttp://www.sohu.com">sohu</a>
</p>

<p>
<a href="hao123http://www.hao123.com">hao123</a>
</p>

</body>
</html>

```

qtp 代码如下：枚举所有的页面 link

```

Dim objPage
Dim objTag
Dim intTotalLink
Dim intCtr
Dim strLink
Dim strHref

Dim oDesc
Dim NumberOfiframe
' 枚举网页中除了带有 iframe 里所有外部的链接
' #####
Set objPage = Browser("Browser").Page("Page").Object
Set objTag = objPage.GetElementsByTagName("A")
intTotalLink = objTag.Length - 1

For intCtr = 0 to intTotalLink
  strLink = objTag(intCtr).InnerText

```

自动化测试	Version:	1.3
文档		2006-5-17

```

strHref = objTag(intCtr).Href
msgbox strLink
msgbox strHref
Next
' #####
' 枚举 iframe 内的所有链接
Set oDesc = Description.Create()
oDesc("html tag").Value = "iframe"
Set iframe = Browser("Browser").Page("Page").ChildObjects(oDesc)
NumberOfiframe = iframe.Count()
For i = 0 To NumberOfiframe - 1
    Set objPage = iframe(i).Object
    Set objTag = objPage.GetElementsByTagName("A")
    intTotalLink = objTag.Length - 1

    For intCtr = 0 to intTotalLink
        strLink = objTag(intCtr).InnerText
        strHref = objTag(intCtr).Href
        msgbox strLink
        msgbox strHref
    Next
Next

Set oDesc=nothing
' #####

```

以上代码在中 Window2000ad sp4+qtp8.2 上调试通过。

备注：

解决这个问题的方法很多，在 51testing cpe 工具使用培训的课程中就提到了解决这个问题的技术，我还写了一套代码同样解决这个问题。

64. Iframe issues 续

QuickTestPro 处理带有 IFRAME 的问题(续)

文章出处：51testing 博客 作者：pcl 发布时间：2006-07-19

今天一上 blog 就看到有人转我的文章！发了这篇文章很久了，但是一直都没人问我另外一个问题，就是我解决这个问题所用到的另外一套代码是什么，很遗憾，所以今天看到其他人转这篇文章的时候，我把另外一套代码也贴出来，共享给大家！

1. 首先建立 HTML 文件，具体页面文件是什么样子请看《QuickTestPro 处理带有 IFRAME 的

自动化测试	Version:	1.3
文档		2006-5-17

问题》一文

2. 利用 vb, 建立 vb 工程文件, 代码如下 (工程需要引入 MSHTML, 在 CPE 课程中 QTP 自动化测试工具实现原理一章我介绍过这个对象, 还有加入另外一个 activex 空间 SHDocVwCtl)

```

Dim webdoc As HTMLDocument

Private Sub cmdfindlink_Click()

Dim ElementCollection As IHTMLElementCollection
Dim x
Dim kx As HTMLDocument
Dim xAcollection As IHTMLElementCollection
,
' 提取超链接
'' 定义元素集
Dim Acollection As IHTMLElementCollection

Set webdoc = wbBrowser.Document
Set ElementCollection = webdoc.All

'' 设置元素集为所有的 Anchor 元素, 用 tags 方法引用元素
Set Acollection = webdoc.All.tags("a")
'' 把 Anchor 元素转换为相应的字符串表示, 用 items 方法引用元素
For i = 0 To Acollection.length - 1
    1stUrl.AddItem Acollection.Item(i).toString
Next
' length 属性返回元素集中元素的个数

For i = 0 To webdoc.frames.length - 1
    Set x = webdoc.frames.Item(i)
    Set kx = x.Document

    Set xAcollection = kx.All.tags("a")

    For j = 0 To xAcollection.length - 1
        1stUrl.AddItem xAcollection.Item(j).toString
    Next

    Set xAcollection = Nothing
Next

```

自动化测试	Version:	1.3
文档		2006-5-17

End Sub

```
Private Sub cmdOpenURL_Click(Index As Integer)
    wbBrowser.Navigate "C:\Documents and Settings\Administrator\桌面\针对某些人的问题
    \test.html"
End Sub
```

实现代码如上。

65. QTP 解析 PDF

如何用 QTP 解析 PDF

文章出处: 51testing 博客 作者: 发布时间: 2006-07-04

今天有人问如何解析 PDF，下边是其中的对话。

Gao. Sheng 说:

你好 qtp 可以解释 pdf 文件吗

[pc1@LoadRunner/QTP/QC SP 认证报名](#) 说:

解释? 还是解析

Gao. Sheng 说:

解析

[pc1@LoadRunner/QTP/QC SP 认证报名](#) 说:

qtp 是功能自动化测试工具，它是基于 gui 元素进行测试的！它不能识别 pdf 里面的文字，如果你要做到解析，那么需要 pdf 的 sdk 作为支撑，开发相应的 dll，然后提供给 qtp 调用就可以了

其实在网上你就可以找到相应的 pdf 解析库，下边这个网站就是

<http://www.pdflib.com/cn/tet.html> 作解析的，可以利用他们写好的东西开发 dll，在你的脚本中调用

66. QTP 常见问题解答一

QTP 常见问题解答一

文章出处: 51testing 博客转 作者: 发布时间: 2007-03-15

- 运行错误: 不能识别对象 (Cannot identify object)

自动化测试		Page 146
-------	--	----------

自动化测试	Version:	1.3
文档		2006-5-17

出现了这个错误时，就表示当前被测程序的窗口中没有符合条件的对象。

出现这种错误，可能的原因有以下两种：

>>> 对象此时在程序中没有显示或不存在。

>>> 在程序中存在该对象，但是它的描述在后来发生了改变，导致与对象仓库中存储的信息不匹配，从而 QTP 无法识别对象。

如果你正遇到“Cannot identify object”的问题，解决方法见 1.1。

1.1 如何去识别对象描述 (Identifying Test Object Description Problems)

如果你能在被测程序中看到对象，但仍出现了“Cannot Identify Object”错误信息，这就表示仓库中对象的描述与程序中对象的描述一定存在着差异。

解决对象描述问题的第一步是找到差异，最简单的方法是将仓库中对象的属性值与程序中对象的属性值进行比较。（仓库中对象的属性值可在 Object Repository 窗口中查看，程序中对象的属性值可用 Object Spy 功能查看。）

比较仓库中对象的属性值与程序中对象的属性值的操作见 1.3。

在完成比较后，是否发现存在差异？

如果是，请参考 1.4；

如果否，请参考 1.5。

1.2 处理对象丢失问题 (Solving Missing Object Problems)

在运行脚本时，QTP 尝试进行某种操作，而该操作的对象在程序中却没有出现，出现这种情况的原因有以下几种：

>>> 对象不再存在。对象已经被从程序中删除。

解决方法：见 1.2.1。

>>> 对象还没有装载。

解决方法：见 1.2.2。

>>> 当前的程序页面（或窗口）不正确，不是对象所在的页面（或窗口）。

解决方法：见 1.2.3。

>>> 前一个步骤没有正确执行。

解决方法：见 1.2.3。

1.2.1 解决对象不再存在的问题

如果对象已经不在被测程序中存在，则应在脚本中修改或删除相关步骤。

1.2.2 解决对象装载超时的问题

如果对象丢失的原因是因为没有足够的时间装载，尝试以下解决方案：

a. 对于 Web 对象，增加 Browser Navigation Timeout (File>Setting>Web 页签) 时间。

b. 在包括该对象的步骤前使用 Wait 语句，让 QTP 在执行该步骤之前等待一段时间。

1.2.3 检查前面步骤的执行情况

如果是因为打开了错误的页面（或窗口）导致对象丢失问题，或因为前一步骤执行错误导致对象丢失问题，则请按以下方法检查原因：

a. 如果怀疑在完成脚本后，被测程序又发生了改变，则检查对象的继承关系以及对象描述。参考 1.1。

b. 如果怀疑脚本错误，则检查脚本的每一个步骤。你可能是遗失了某个步骤，也可能是使用了不正确的参数或方法。

自动化测试	Version:	1.3
文档		2006-5-17

67. QTP 常见问题解答二

QTP 常见问题解答二

文章出处: 51testing 博客转 作者: 发布时间: 2007-03-15

1.3 测试对象的属性值与 Run-time 对象的属性值的比较 (Comparing Test Object and Run-Time Object Property Values)

根据以下的步骤比较测试对象与 Run-time 对象的属性值:

- 1) 进入 Object Repository 窗口 (Resources>Object Repository)，选择对象。
- 2) 用笔记下对象的 class 以及它的各个属性及属性值。
- 3) 打开被测程序，并打开包含被测对象的页面或窗口。
- 4) 在 QTP 中选择菜单 Tools>Object Spy 或点击 Object Spy 按钮，打开 Object Spy 对话框。
- 5) 选择“Test Object Properties”选项。
- 6) 点击右上角的按钮（带有手图标的），这时 QTP 窗口以及 Object Spy 对话框都被最小化。
- 7) 在程序页面（或窗口）中点击目标对象，恢复 Object Spy 对话框，并在对话框中显示对象及其父对象（以树的形式显示），并在 Properties 页签中显示当前对象的属性及属性值。
- 8) 这时 Object Spy 对话框的 Properties 页签中显示 Run-time 对象的所有属性，在此查看对象的 class，以及它的属性及属性值，并将它与第 2 步中记下的内容进行比较。

1.4 解决对象描述存在的问题 (Solving Object Description Problems)

如果发现仓库中对象的属性值与程序中对象的属性值存在不同，你应该判断这个不同是个别对象的问题，还是其它同类对象也存在相同的问题。

是所有（或多个）同类对象都存在问题吗？

如果是，则参考 1.6

如果否，则参考 1.7

1.5 关于父对象描述的识别问题 (Identifying Parent Object Description Problems)

对象的识别还与它所继承的父对象有关。

如果你能在被测程序中看到对象，而且程序中对象的属性值与仓库中对象的属性值也是一致的，但仍然遭遇到了“Cannot identify object”错误，则这个错误可能与它的父对象有关（如仓库中父对象的属性值与程序中父对象的属性值不一致），也可能是仓库中的对象与程序中对象的继承关系不相同导致的。

要判断是继承关系中的哪个父对象出现了问题，请尝试下面的方法之一：

>>>重新录制对象，比较新旧对象的父对象。

欲了解如何完成比较，参考 1.8。

>>>对继承关系中的每个父对象，分别插入一个 Exist 语句，并运行该部分

自动化测试	Version:	1.3
文档		2006-5-17

脚本。

欲了解如何创建 Exist 语句，参考 1.9。

注：你也可以使用 Object Repository 窗口中的 Highlight in Application 功能，在被测程序中定位对象。

一旦找到了存在问题的父对象，接下来尝试以下方法：

>>>修复存在问题父对象的描述。方法见 1.4。

>>>在专家视图的模式下，查找所有继承存在问题的实例并进行修正。

举例：1.8.1。

1.6 解决某类对象的识别问题 (Solving Object Identification Problems for a Test Object Class)

如果你发现某类对象的对象描述对于被测程序来说不是很合理，或你预期到对象描述中的某属性值是经常变化的，你可以在 Object Identification 对话框中改变该类的识别属性的设置，或定义该类的 Smart Identification 设置并启用 Smart Identification 机制，这样 QTP 就可以唯一识别对象了。

1.7 解决单个对象的描述问题 (Solving Individual Test Object Description Problems)

选择下列方法之一来解决对象的描述问题：

>>>如果被测程序中对象描述发生了改变，并且你也清楚改变的内容，并且该改变是永久性或长期性的，你可以直接手工修改仓库中对象描述中的相关属性值。

>>>如果被测程序中的对象的属性值依赖于前面的步骤或其它对象，则将该属性值参数化，这样就可以使用其它步骤的输出值为属性值。

>>>如果属性值的组成部分中，部分是固定的，部分是动态改变的，则可以将属性值设计为正则表达式。

>>>如果属性值是遵循某种规则变化的，或者是不可预期的，则从对象描述中移除该属性，并向对象描述中添加一个或多个属性以便于 QTP 进行对象识别。

>>>If you can only access the information on the property values during the run session, you can create and use functions that use programmatic descriptions to identify the object using property values retrieved earlier in the run session.

For more information on programmatic descriptions, see Using Programmatic Descriptions.

>>>如果在录制过程中对象是唯一的，但现在程序中出现了两个或多个描述相同的对象，但是它们在页面（或窗口）中的位置不相同，则应在 Object Properties 或 Object Repository 窗口中，向对象描述中添加一个 ordinal identifier (index 或 location)。

1.8 重新录制对象，以判断父对象描述是否存在问题 (Re-recording an Object to Identify Parent Object Description Problems)

根据下面的指引，来判断父对象的问题。

通过重新录制对象，来判断它的父对象是否存在问题：

1) 打开浏览器或程序，来到包含被测对象的页面或窗口。

2) 在关键字视图模式，选择最后一个组件 (component)，或在专家视图模式，将光标放在最后一个脚本步骤的下面。

3) 点击 Record 按钮，或选择 Automation>Record。

自动化测试		Page 149
-------	--	----------

自动化测试	Version:	1.3
文档		2006-5-17

- 4) 点击（或操作）页面或窗口中的目标对象。
- 5) 点击 Stop 按钮，或选择 Automation>Stop，完成步骤添加。
- 6) 右击新添加的步骤，并选择 Object Properties 右键菜单。
- 7) 在 Object Properties 对话框中，点击 Repository，打开 Object Repository 窗口，并选中了新对象。注意记下该对象的继承关系。
- 8) 在 Object Repository 窗口中找到旧的对象（即存在问题的对象），将它的继承关系与新对象的继承关系进行比较。

通过比较，可以检查到新旧对象是父对象的描述存在不同，还是它们的继承关系根本就不同。

- a. 如果是父对象的描述问题，则修改父对象的描述。参见 1.4.
- b. 如果是对象的继承关系不正确（即具有不同的父对象）引起的问题，则在脚本中找到所有使用该继承关系的实例，将它们替换为正确的继承关系。

举例：参见 1.8.1。

1.8.1 举例

假设有一个带有 Frame 的网站，你录制了一个操作针对 Image 对象“Poster”的操作步骤。在 Object Repository 窗口，你可以 Image 对象的继承关系如下所示：

```

MyCompany (Browser)
  MyCompany (Page)
    Main(Frame)
      Poster(image)

```

当你运行脚本时，网页看起来没有什么不同，但是识别 Image 对象时却出现了“Cannot identify object”的错误提示。当你重新录制操作步骤，然后在 Object Repository 窗口中查看新的 Image 对象时，发现该对象的继承关系如下所示：

```

MyCompany (Browser)
  MyCompany (Page)
    Poster(image)

```

从中可以看出，Frame 已经从 Web Page 中移除，所以尽管 Image 对象“Poster”的描述没有发生任何改变，但它的继承关系已经改变了。

在 Object Repository 窗口，你可以看到新旧 Image 对象“Poster”是两个完全不同的对象：旧的位于已经被移除的 Frame 对象之下，而新的则直接位于 Page “Mycompany” 对象之下。

修复这个问题，可以在脚本步骤中将旧的对象替换为新的对象。

为了保持 Object Repository 窗口的整洁，你应删除那些已不再存在的对象（如本例中的 Frame）。

1.9 创建 Exist 语句，以判断父对象的描述是否存在问题

QTP 在录制发生在某个对象上的操作时，也会同时去了解对象以及它的继承关系。因此，如果在运行过程中，QTP 不能识别继承关系中的任何一个父对象，都会导致目标对象的识别失败。

使用带有 Exist 方法的 MsgBox 语句，可以方便的检查出是继承关系中的哪个父对象出现了问题。

假设在运行下面脚本时，你遇到了“Cannot identify Object”错误信息（但是可以确定对象仓库中 Link 对象的描述与当前程序中对象的描述是完全一致的）：

自动化测试	Version:	1.3
文档		2006-5-17

Browser("Yahoo!").Page("Yahoo!").Link("Arts & Humanities").Click
 你可以在该语句前面插入以下语句:

```
Msgbox Browser("Yahoo!").Exist
Msgbox Browser("Yahoo!").Page("Yahoo!").Exist
Msgbox Browser("Yahoo!").Page("Yahoo!").Link("Arts &
Humanities").Exist
```

然后你从头开始运行脚本，发现第 1 行语句运



但是，当 QTP 运行到第 2 步时，就弹出信息框，内容为 “False”，表示不能找到 Page object。这就证明 Page 对象存在问题。

确定了出现问题的父对象以后，修复该问题。参见 1.4。

68. QTP 常见问题解答三

QTP 常见问题解答三

文章出处: 51testing 博客转 作者: 发布时间: 2007-03-15

2. 运行错误: 对象不唯一 (The object is not unique)

在被测页面或窗口中的同一父对象下，找到多个与仓库中对象描述相符的 Run-tim 对象。

请确定是只有个别对象存在此问题，还是此类对象的所有对象（或部分对象）存在此问题。

此类对象的所有对象（或部分对象）都存在此问题吗？

如果是，参考 2.1；

如果否，参考 2.2。

2.1 为某对象类设置能唯一识别对象的描述规则 (Configuring Unique Test Object Descriptions for a Test Object Class)

如果你发现某类对象的对象描述都不足以唯一识别对象，你可以在 Object Identification 对话框中，改变该类的识别属性的设置，或定义该类的 Smart Identification 设置并启用 Smart Identification 机制，这样 QTP 就可以唯一识别对象了。

2.2 设置通唯一识别对象的对象描述 (Creating a Unique Test Object Description for an Object)

查看当前对象的描述，并尝试在程序中找到其它具有相同描述的对象并查看它们的描述。

>>> 如果你找到了可以将它们区别开来的一个或多个属性，则修改仓库中对象的相关属性（如向描述中添加属性等），以使 QTP 在运行时可以唯一识别它。

自动化测试	Version:	1.3
文档		2006-5-17

>>>如果它们的识别属性完全相同，则在对象描述中为对象添加一个 ordinal identifier (index 或 location)。

3. 运行错误: 找不到父对象 (Parent not found)

识别对象时，会先识别它的父对象是否正确。如果遇到“Parent not found”的错误信息，请尝试以下解决方法：

>>>修复出现问题的父对象的描述。参见 1.4。

>>>在脚本中找到继承关系不正确的实例，并修改为正确的继承关系。例如 1.8.1。

69. QTP 常见问题解答四

QTP 常见问题解答四

文章出处: 51testing 博客转 作者: 发布时间: 2007-03-15

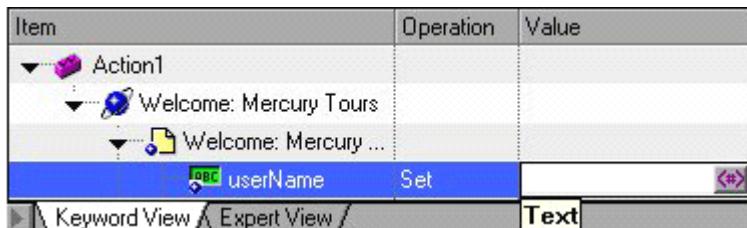
4. 运行错误: 无效的参数 (Invalid arguments)

一个或多个方法的参数是无效的。

参数无效可能是参数类型错误，也可能是参数个数不正确。

根据以下建议，找到方法的正确语法：

>>>在 Keyword 模式，如果方法包括参数，则在 Value 栏会显示该方法的参数提示。



>>>在 Expert 模式，如果方法包括参数，则当你插入方法然后按下 Shift+Ctrl+Space 键时，QTP 会显示该方法的参数的提示信息。（注：只有在 Editor Options 对话框的 General 页签中打开了 Statement completion 功能时才可以看到参数的提示信息。）



5. 运行错误: 不能识别 item (Cannot identify item)

QTP 不能识别方法参数中指定的列表或树对象的 item 时，请确认指定的 item 是否存在于对象之中。

>>>如果 item 不存在，则删除相关脚本步骤，或修改 item。

>>>如果 item 存在，但被测程序中它的名字发生了改变，则修改 item 的名称，或在脚本中使用 item 的 index 而不是名字（如“#4”）。注意，不同的对象或方法，它们使用 index 的语法可能不同。

6. 运行错误: 对象的 item 不唯一 (The object's item is not unique)

在列表或树对象中，可能有多个名字相同的 item。

自动化测试	Version:	1.3
文档		2006-5-17

在这种情况下，可以指定 item 使用 index (例如 “#4”)。

7. 运行错误：运行失败 (Test run failed)

测试运行失败，还有可能是其它未知错误造成的。检查出现错误时的脚本步骤。

如果错误出现在执行检查点或输出值时，请查阅关于检查点与输出值的帮助。

如果错误出现在使用外部文件时，如**数据库**表、数据库、环境变量文件、或关联仓库文件等，则检查是否可以正常获取并使用文件，或这些文件的存储格式是否正确。

如果错误出现在完成某个脚本动作时，请参考 QuickTest Object Model Reference 帮助，以确认方法或函数的使用是否正确。

如果错误出现在执行一个标准的 VBscript 语句时，参考 Microsoft VBscript Reference 以获得帮助。

70. QTP 设计开发脚本

QTP 脚本还可以这样写

文章出处：51testing 博客 作者：阳阳 发布时间：2006-12-22

一提到自动化测试工具，给人印象最深的就是录制和回放了。录制测试人员的操作过程，再回放出来并多次运行，达到使用机器代替人来执行测试的效果。

QTP 采用的关键字驱动与专家视图的设计思想，不但可以自动生成代码，使初学者快速入门，更为高级测试人员提供了便利的编程界面，使其可以高效的完成自动化测试脚本，不用完全依赖于录制和回放过程。

下面就我个人的使用经验，简单谈谈如何不依赖于录制和回放来完成 QTP 脚本的。

操作步骤很简点，主要分为三步：

第一步，维护对象库，添加必需的对象以及配置对象的属性。

第二步，在专家视图中，编写测试代码。

第三步，运行并调试代码。

举个简单的例子。

以在 51testing 的网站上搜索关键字 QTP 为例。

首先，打开 QTP，点击工具栏上的 Object Repository 按钮，打开对象库维护界面。
如图一（图片暂时上传不了）

点击 Add Objects 按钮，添加站内搜索的文本框以及后面的查询按钮对象，如图二

然后，在专家视图中，手动输入如下代码：

```
Browser("软件测试专业网站:51Testing 软件测试网").Page("软件测试专业网站:51Testing 软件测试网").WebEdit("keyword").Set "QTP"
Browser("软件测试专业网站:51Testing 软件测试网").Page("软件测试专业网站:51Testing 软件测试网").WebEdit("content").Set "QTP"
```

自动化测试	Version:	1.3
文档		2006-5-17

Testing 软件测试网").Image("GO").Click

由于录制的过程中，待测试的页面是已经打开的，所以为了保证这个测试的前提条件，在前面添加打开 IE 的语句。

```
SystemUtil.Run "iexplore.exe", ""
```

这样，使用 QTP 的自动化测试脚本就完成了，运行看看是否通过？结果是一次性通过，而且没有其他需要修改的地方了。

还可以在此基础上增加关闭所有 IE 的函数，增加验证点等等使脚本更加健壮和完整，这里就不赘述了。

如果使用录制和回放的方法来开发脚本，可能存在以下几个问题：

一个是自动生成的代码有一些是没有用的，有时候反而会影响代码的质量；

一个是通过录制生成的代码，健壮性都不好，如果要使脚本在无人值守的情况下顺畅地运行，还是需要手动修改的。比如在对象找不到的地方增加判断对象是否存在，再继续下一步操作等等；

再有，就是录制的过程中，会使你的对象库中多出来一些脚本正常运行以外的多余的对象，给维护对象库也增加了工作量。

比较这两种方法，总体感觉直接添加对象并编写代码的方式，可以弥补录制回放过程的不足，节省了脚本的开发时间，减少了对象库的维护工作量，大大提高了自动化测试脚本的开发效率。

当然，能够这样做是有前提的。

第一，必须对待测试的系统非常熟悉。即开始写自动化脚本之前就能从大脑中反映出每一个步骤中需要的对象及操作。

第二，对 QTP 的工作原理以及工具本身非常熟悉。能够有办法解决脚本运行时出现的任何问题。

第三，有良好的编程基础尤其是针对 VBscript 语言的。如果对于代码的结构、可维护性都把握很好的话，那么效果就会更加明显了。

补充一点，使用这种方法，你的对象库的设计一定要到位，保证在写代码的时候，很容易的找到你要操作的对象，否则就得不偿失了。

原始连接：

http://blog.51testing.com/?8181/action_viewspace_itemid_1692.html

71. 设置共享对象库

QTP 中设置共享对象库

文章出处：51testing 博客 作者：xiaonan 发布时间：2006-12-31

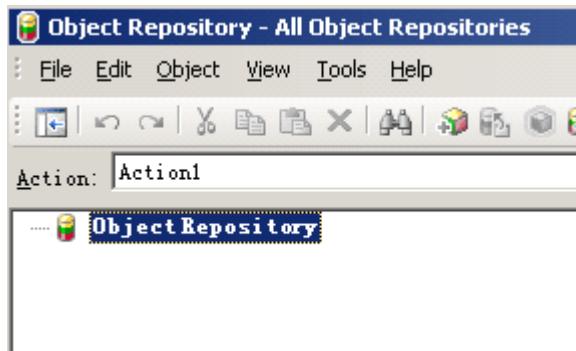
自动化测试	Version:	1.3
文档		2006-5-17

很多时候我们总是觉得管理 QTP 的脚本比较烦.因为除了要对代码的管理之外,还要保证 QTP 对象库的完整. 每一个用例脚本的生成, 同时又都会生成一个对象库. 那么是不是可以做一个公共的对象库来给各个用例脚本调用呢? 接下来, 我们就来实现共享对象库的应用.

第一步: 把需要加到共享对象库中的各个用例脚本的对象库, 分别导出成. tsr 文件.

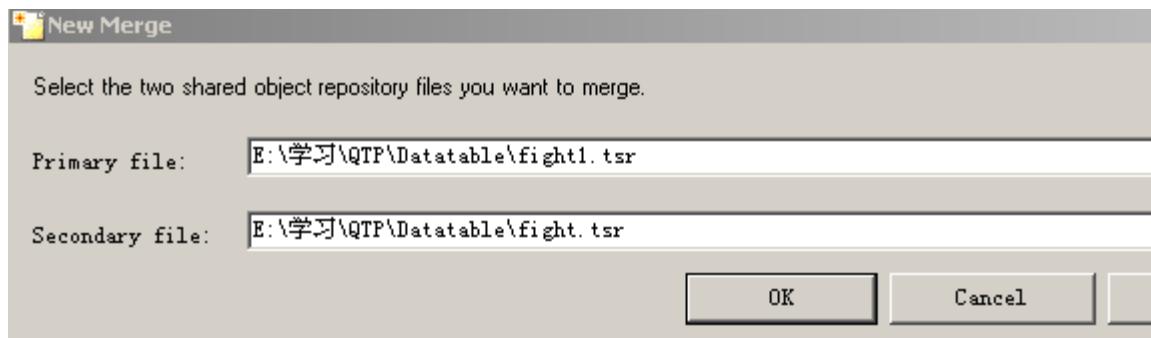
操作方法:先用 QTP 打开已经录制完毕的脚本后, 选择 Resources—> Object Repository

然后 file-->export....取名导出文件. 如下图:



第二步: 把需要加入到共享对象库中的各个用例脚本的对象库, 合并对象及对象属性, 形成一个大的共享对象库.

操作方法:打开 Resources—>Object Repository Manager. 找到 Tools 菜单, 然后选择打开 Object Repository merge tool. 选择两个要合并的对象库文件, 进行对象合并. 一次只能合并两个. 所以如果这里你要合并多个对象库文件的话, 先合并两个后, 保存成一个新的. tsr 文件. 然后再重复上面的操作, 选择这个新的. tsr 文件和另一个准备好的对象库合并. 如下图:

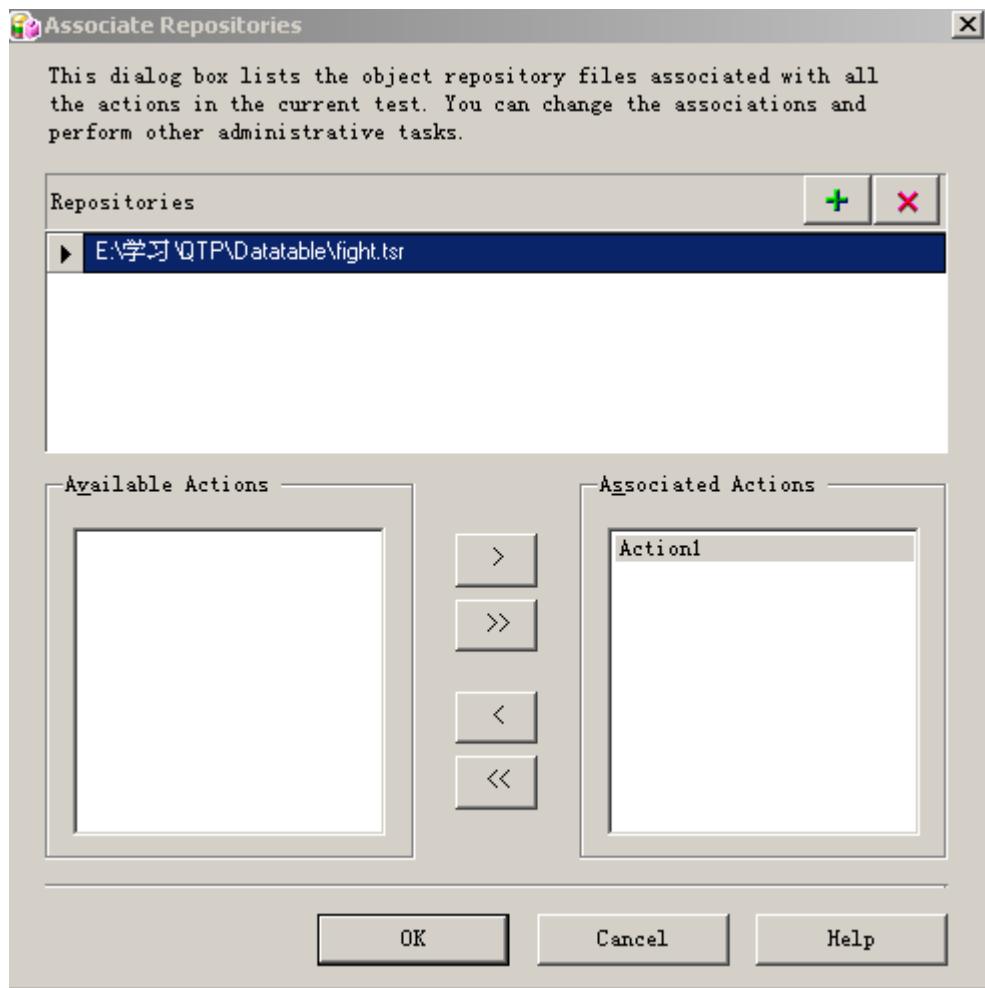


合并的时候, 你会发现, 在两个对象库文件中, 相同的对象合并成一个, 不同的对象, 全部被完整增加进去. 然后形成一个大的对象库. 现在只要保存这个共享对象库. 如果以后有新增的对象, 可以重复上述操作.

第三步: 调用上面保存好的共享对象库. 给新的脚本使用.

操作步骤: Resources—>Associate Repository. 选择上面保存好的共享对象库的. tsr 文件. 加入到 Associate Repository 中. 如下图:

自动化测试	Version:	1.3
文档		2006-5-17



完成这一步后，其实整个共享对象库的操作已经完成。你在对象库中可以看到以后的对象。在录制新脚本的时候，如果对象已经存在，就不会再被记录，只有在这个对象库中没有的对象才会被记录进去。回放脚本，对象识别成功。

(注：此操作只适合与装有 QTP9.0 的版本。对于 8.2 版本不适合)

附：8.2 版本中的共享对象库操作

QTP8.2 的共享方法

1. 选择“测试”>“设置”>“资源”选项卡。
2. 在“对象库类型”区域，“共享”。

请指定要用作对象库文件的共享对象库文件。要指定文件，请输入对象库文件名，或单击浏览按钮并从“打开”对话框中选择资源文件。要新建共享对象库文件，请在“共享”框中输入新文件名。

对象库文件的默认文件扩展名是 .tsr，但是文件可能使用任何扩展名。在“打开”对话框中浏览现有对象库文件时，在“文件类型”框中选择“所有文件”。

3. 注：请在新测试开始录制前进行设置，已有测试的对象库模式不能进行修改。

此文来源于 51testing 博客，转载请注明出处为 51testing 博客

原始链接：

http://blog.51testing.com/?26649/action_viewspace_itemid_2278.html

自动化测试	Version:	1.3
文档		2006-5-17

72. 检查网页上文字显示的颜色

如何使用 qtp 检查网页中显示的文字颜色为指定的颜色

文章出处: 转载 作者: 发布时间: 2006-12-29

这里我们使用 DOM(Document Object Model) 编程技术来完成这一功能的检查。下面举个例子来说明一下:

以 Mercury 自带的定票网站为例, 在用户注册页面中, 用户名、密码和确认密码三项的文字应为红色显示, 下面的代码验证了“密码”为红色。

```
Set Elements = Browser("Register: Mercury Tours").Object.Document.all.tags("TD")
For each Element in Elements
  If Element.InnerText= "Password:" Then
    strhtml=Element.Innerhtml
    n=instr(strhtml,"color=read")
    If n<>0 Then
      Reporter.ReportEvent 0, Element.InnerText, "Text color is red!"
    End If
  End If
Next
```

其中 Element.Innerhtml 的值为**Password:**, 可以看见 color=red 就是定义了文字的颜色为红色, 所以我们在之后的代码中只要验证 Element.Innerhtml 中存不存在 color=red 就可以了。

不是所有的 html 代码都是如上编写的, 有的时候会把 style 单独存放在一个 css 文件中, 然后在网页中调用, 如Password:, 所以我们要根据实际取到的 Element.Innerhtml 值来编写后面的程序, 这里就要验证 Element.Innerhtml 中存不存在 class=redcode, 即代码第五行改为 n=instr(strhtml,"class=redcode")。

所以大家要根据代码的实际情况来编程验证, 这里只提供了一种验证的方法。

我也找到另外一种, 紧供参考:

```
How to know color of an object
' Example:
Dim color
'Grey. Use "#000000" for black
color = "#999"
'Get the DOM object for the text
Set TxtObj =
Browser("myBrowser").Page("myPage").WebElement("myWebElement").Object
retVal = FontColorCheckPoint ( TxtObj, color )
```

```
Function FontColorCheckPoint ( obj, vExpColour )
Dim vActualcolor, sReportStepName, sErrMsg

' get object's color
vActualcolor = obj.currentStyle.color
sReportStepName = "Custom Checkpoint ""Font Color"""
```

自动化测试	Version:	1.3
文档		2006-5-17

```

sMsg = "The expected color " & vExpColour & " was found"
'sErrMsg = "The expected color " & vExpColour & " was NOT found.
Found " & vActualcolor & " instead."
sErrMsg = "The expected color " & vExpColour & " for the object " &
Chr(34) & obj.innertext & Chr(34) & " was NOT found. Found " &
vActualcolor & " instead."
If vActualcolor <> vExpColour Then
Reporter.ReportEvent 1, sReportStepName, sErrMsg
FontColorCheckPoint = False
Else
Reporter.ReportEvent 0, sReportStepName, sMsg
FontColorCheckPoint = True
End If

```

73. QTP 录制如何识别编辑框

用 QTP 录制时如何识别编辑框

文章出处: 51testing 论坛 作者: Lola1123 发布时间: 2007-03-16

录制过程中, 如何对 Budget 的字段输入信息进行录制, 因为其它字段的输入是通过下拉框或链接的得到的;

录制产生的脚本如下:

```

Browser("TenantDef").Window("DialogFrame -- Web
Page").Page("DialogFrame").Frame("Frame").Image("HyperLink").Click
Browser("TenantDef").Window("DialogFrame -- Web
Page").Page("DialogFrame").Frame("Frame").WebElement("WebElement").Click
Browser("TenantDef").Window("DialogFrame -- Web
Page").Window("DialogFrame -- Web
Page").Page("DialogFrame").Frame("Frame").WebList("lstOC").Select "BSH-
HA"
Browser("TenantDef").Window("DialogFrame -- Web
Page").Window("DialogFrame -- Web
Page").Page("DialogFrame").Frame("Frame").WebButton("OK").Click

```

**Browser("TenantDef").Window("DialogFrame -- Web
Page").Page("DialogFrame").Frame("Frame").WebEdit("WebEdit").Set
"1000"** 这是 budget 字段录制的脚本, 但是不能运行; 如何解决

```

Browser("TenantDef").Window("DialogFrame -- Web
Page").Page("DialogFrame").Frame("Frame").WebButton("OK").Click

```

解决方法

自动化测试	Version:	1.3
文档		2006-5-17

尝试用 virtual object 来解决，失败；
 尝试在 object repository 来解决，录制结果如上边红色标记行代码，运行脚本仍失败；
 尝试在 low level recording 中解决，亦失败；
 后来联想到 Winrunner 中使用 Analog recording 的模式来录制签名的方法，于是在录制此操作时切换成 Analog recording 的模式，结果让人兴奋，成功了！！！
 与你分享！！

原始链接： http://blog.51testing.com/html/63/102063_itemid_4974.html

74. How to search the string from the html source

Description [How to search for a string from HTML source. If the search string is a web URL then this article also allows us

Solution [Let's assume that the string from HMTL source that we are searching for is as follows. Now I want to just sea

```
 QuickTest Professional 下点击 QuickTest Plus，然后按照提示一步步往下安装即可，其中要求输入序列号，输入和 QTP 安装时相同的序列号就可以了（8888-8888888888）。

#### 二、提示和技巧

plus 不仅提供了一些工具，还在它的帮助手册里给出了一些提示和技巧，以及一些实用的 Function。在这里我把一些比较常用的好东东贴出来，其他的就看 plus 的帮助吧。

##### 1、创建 action template.

当希望在每一个新建 action 时都增加一些头部说明，比如作者、创建日期、说明等，用 action template 来实现最简单快捷。

方法：用记事本等文本编辑器，输入如下类似的内容：

```
'Company: xxxx
'Author: xxx
'Product: xxx
'Date: xx
```

然后将文件保存为 ActionTemplate.mst，并放到 QTP 安装目录下的 dat 目录，重启 QTP，新建一个 action 试试，新建的 action 会包含以上信息。

##### 2、关于设置测试报告里只显示 error 的信息。

帮助中说：修改安装目录下 bin\QTReport.ini 文件，增加以下内容：

```
[FilterDialog]
ReportAppDefaultFilter=1 # for error only
ReportAppDefaultFilter=3 # shows all messages (default)
```

但根据我的测试结果，不尽其然：

1) 当 ReportAppDefaultFilter=1 时，如果 Object Repository 中缺少对象，在报告中会在相应的 action 前打叉，但不会提示具体错误，而成功的步骤都有具体信息显示。

2) 用 Reporter.ReportEvent 测试的结果是：

```
ReportAppDefaultFilter=1 时，只显示 micDone 的具体信息;
ReportAppDefaultFilter=2 时，只显示 micFail 的具体信息;
ReportAppDefaultFilter=3 时，只显示 micDone 和 micFail 的具体信息;
ReportAppDefaultFilter=4 时，只显示 micPass 的具体信息;
```

似乎无规律可寻，所以我的结论暂时是：不要设置这个参数，用默认的，显示所有信息，更多的信息有利于分析结果。

##### 3. 启动 IE 的语句：SystemUtil.Run "iexplore.exe", "http://www.mercuryinteractive.com"

##### 4. 关闭 IE 或其他程序的语句：SystemUtil.CloseProcessByName "app.exe" or SystemUtil.CloseProcessByWndTitle "Some Title"

#### 三、Function Libraries

plus 的帮助中提供了一些常用的 Function，把这些 function copy 到文本编辑器中保存为.vbs 文件，并添加到 Resources 中就可以直接调用了，或直接 copy 到你的 action 中，就可以在当前的 action 中调用。更推荐第一种方法，所有的 action 都可以调用。

##### 1、文件操作相关的 function，如下，望名则可生意：

```
Function CreateFile(sFilename, bOverwrite);
Function OpenFile(sFilename, iomode, create);
Function AppendToFile(sFilename, sLine);
Function WriteToFile(sFilename, sLine);
```

##### 2、Function NormalizeString(OrgStr); （将字符串变成 regular express）

##### 3、GlobalDictionary 的使用，这是另外一种可以共享全局变量的方法，在所有的 action 中，包括 local 和 external action 中都可以访问。

##### 4、使用文件系统相关的 function:

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

```

Function ReadLineFromFile (byref FileRef);
Sub FileDelete (FilePath);
Function FileCompare (byref FilePath1, byref FilePath2, byref FilePathDiff, ignoreWhiteSpace);
Function CheckFileExists (FilePath)

```

#### 5. web table 相关的 function:

Function ItemByKeyColumn(): 根据 table 中某列的值，得到同一行中另一列的对象。（这个功能非常有用。）  
Function ObjectsByMicClass(Obj, micClass): 得到 table 中所有的 micClass 类型的对象集合。

### 四、工具

#### 1、Automation Generator Utility

添加一系列动作，然后自动完成。如：启动 QTP，然后执行 test1, test2, test3..., 最后关闭 QTP，还可以连接或断开 Quality center。  
值得注意的是，每个 test 都可以分别指定 test result 文件，这样可以把所有 test result 指定到同一个目录下，所有 test 执行完成后，用 Test Result Viewer 就可以快速的查看测试报告。

#### 2、External Action Call Modifier Utility

外部 Action 调用修改工具。当删除一个 Reusable action 前，如果没有先删除其他 test 中对它的调用，则打开其他 test 时，会提示找不到某某 action，这种情况下启用该工具，工具会列出 test 下所有调用的外部 action（如果调用的 action 是使用相对路径，则不会列出来），如果外部 action 找不到，会用红色的问号表示，删除它并保存后就 OK 了。  
也可以在此处修改外部 action 的来源，修改后原来的 action parameter 仍会保留。

#### 3、Repository Merge Utility

合并多个对象库文件中的对象，如果出现冲突，可以选择忽略，或手工合并，或自动合并。  
对象库文件是 Action 目录下的 Resource.mtr 文件。

#### 4. Report Analyzer

Test Result 的另一个查看工具，提供了一些过滤条件，比如只查看 failed step，或只查看 checkpoints，或只查看某个 action。

先就写这么多了，抛砖引玉，希望你能发现你想要的东西。写得不对的，不要客气，请指出来，不胜感激。

## 78. Schedule the QTP tasks

### 用 Schedule 定时运行 QTP 脚本

文章出处: 51testing 博客 作者: 风过无息 发布时间: 2007-03-26

创建一个.vbs，在控制面板的 Schedule 里面定义自己的 task 就好

vbs 的脚本如下：

```

Dim qtApp 'As QuickTest.Application ' Declare the Application
object variable
Dim qtTest 'As QuickTest.Test ' Declare a Test object variable
Dim qtResultsOpt 'As QuickTest.RunResultsOptions ' Declare a Run
Results Options object variable
Set qtApp = CreateObject("QuickTest.Application") ' Create the
Application object
qtApp.Launch ' Start QuickTest
qtApp.Visible = True ' Make the QuickTest application visible
' Set QuickTest run options
qtApp.Options.Run.ImageCaptureForTestResults = "OnError"
qtApp.Options.Run.RunMode = "Fast"

```

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

```

qtApp.Open "C:\ Your Test Path Here", True ' Open the test in
read-only mode
' set run settings for the test
Set qtTest = qtApp.Test
qtTest.Settings.Run.OnError = "NextStep" ' Instruct QuickTest to
perform next step when error occurs
Set qtResultsOpt = CreateObject("QuickTest.RunResultsOptions") '
Create the Run Results Options object
qtResultsOpt.ResultsLocation = "C:\Your Results Path Here" ' Set
the results location
qtTest.Run ' Run the test
'qtTest.Close ' Close the test
Set qtResultsOpt = Nothing ' Release the Run Results Options
object
Set qtTest = Nothing ' Release the Test object
Set qtApp = Nothing ' Release the Application object

```

## 79. General the radom code

产生背景：经常会有一些输入域的[测试](#)，少的有 255，多的更可怕，在准备数据的时候会比较头疼，所以写了个函数和大家分享。

```

Function makestring(inputlength)

If IsNumeric(inputlength) Then

For I = 1 To inputlength

'you may add a random function here

A = Array("a","b","c")

Randomize

x=RandomNumber (0,2)

B = A(x)

makestring =makestring +B

Next

msgbox ("output the string:"&makestring)

else

```

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

```
msgbox ("error format:&inputlength)
```

```
End If
```

```
End Function
```

```
Call makestring("8")
```

## 80. ChildObjects 和 webTable 的使用

2007-03-12 12:37:39 / 个人分类: [QTP](#)

```
dim maxCodeRow,maxCode,temp,webTableIndex
```

```
Dim obj
```

```
maxCode=0
```

```
Set oDesc = Description.Create()
```

```
oDesc("micclass").Value = "WebTable"
```

```
'获得 Browser("test").Page("test").Frame("mainFrame")下所有 webTable
```

```
Set Lists =Browser("test").Page("test").Frame("mainFrame").ChildObjects(oDesc)
```

```
'获得 webTable 的总数
```

```
NumberOfLists = Lists.Count()
```

```
'对 text ="XXX"的 webTable 进行操作
```

```
For listIndex = 0 To NumberOfLists - 1
```

```
Set obj=Lists(listIndex)
```

```
text=obj.GetCellData(1,1)
```

```
if(text ="XXX") then
```

```
num=obj.RowCount'获得这个 webTable 的行数
```

```
For webTableIndex=2 to num'找出自己想要操作的行号, 在这里取得编码为最大的行号
```

```
temp=obj.GetCellData(webTableIndex,1)
```

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

```

if(temp > maxCode) then

maxCode=temp

maxCodeRow=webTableIndex

end if

Next

obj.childItem(maxCodeRow,9,"Link",0).Click'点击该行的"删除"连接

Browser("test").Dialog("Microsoft Internet Explorer").WinButton("确定").Click

exit for

end if

Next

```

## 81. 怎样在 QTP 中打开右键菜单

2007-03-02 17:16:04 / 天气: 晴朗 / 心情: 高兴 / 个人分类: [QTP](#)

如果要打开右键菜单，在 [QTP](#) 中可以使用以下函数：

[\*\*Click\*\*](#) [x], [y], [BUTTON]

在 QTP 的帮助中，对 X、Y、BUTTON 分别做如下定义：

X: Optional. A Long value. The x-coordinate of the click.

Y: Optional. A Long value. The y-coordinate of the click.

Tip: You can enter micNoCoordinate for the x and y argument values if you want to enter a value for the button argument without specifying x- and y- coordinates for the click.

BUTTON: Optional. A pre-defined constant or number. See the Constants table, below. Indicates the mouse button that performs the click. Default = micLeftBtn

这是在 WEB 方式下的定义，在 C/S 应用程序中定义略有不同：

X: Optional. An Integer value. The x-coordinate of the click. Note that the specified coordinates are relative to the upper left corner of the object.

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

Default = micNoCoordinate (center of the object).

Y: Optional. An Integer value. The y-coordinate of the click. Note that the specified coordinates are relative to the upper left corner of the object.

Default = micNoCoordinate (center of the object).

BUTTON: Optional. A pre-defined constant or number. See the Constants table, below. The mouse button used to click the object. Default =

micLeftBtn (0).

大家都知道点击鼠标有三类操作，分别是左击、右击、中间击。对应这里的 BUTTON 就有以下几种应用：

micLeftBtn or 0 - The left mouse button.

micRightBtn or 1 - The right mouse button.

micMiddleBtn or 2 - The middle mouse button.

Example 1:

```
Window("Notepad").WinEditor("Edit").Click 4, 4, micRightBtn
```

```
Browser("Google").Page("Google").Link("Advertising Programs").Click 5,4, micRightBtn
```

Example 2:

\*这是从 <http://www.kernzhang.com/viewtopic.php?t=15> 网上摘抄的一个例子，楼主 **kernzhang** 说如果用了 WEB ADD IN，则需要修改 ReplayType。

但是我试了直接用 Browser("Google").Page("Google").Link("Advertising Programs").Click 5,4, micRightBtn 语句也能成功回放，不知道 **kernzhang** 为何做如此说，是否有相关的配置，我现在还不知道，以后再慢慢研究。这个例子的下一段是点击右键菜单中第四个菜单操作的脚本 \*

```
cur_replay_type = Setting.WebPackage("ReplayType")
' Change replay type to mouse replay..
Setting.WebPackage("ReplayType") = 2
Browser("Google").Page("Google").Link("Advertising Programs").Click 5,4, micRightBtn
Setting.WebPackage("ReplayType") = cur_replay_type
' Select the fourth menu item
idx=4
```

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

```

Set WshShell = CreateObject("Wscript.Shell")
For i = 1 To idx
 WshShell.sendKeys "{DOWN}"
Next
WshShell.sendKeys "{ENTER}"
Set WshShell = nothing

```

在 WEB 方式下，虽然直接写脚本可以进行回放，但是用录制却不能把点击右键的操作录制下来，不知道为何？

## 82. **CaptureBitmap** 方法

### **CaptureBitmap** 方法

**Object** 的 CaptureBitmap 方法可以将对象图片拷贝到物理路径中存档。具体语法如下：

**object.CaptureBitmap FullFileName, [OverrideExisting]**

描述：

**object**: A test object of type WinEdit.

**FullFileName**: Required. A String value.

**OverrideExisting**: Optional. A Boolean value. 0 or 1(True or False) i.e to over the present file or not.

**OverrideExisting**: 0 或者 true 是指如果该文件已经存在，则覆盖。

1 或者 false 如果该文件已经存在，则出错，告知文件已经存在。

实例：Window("Flight Reservation").CaptureBitmap "e:\f.bmp",true

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

### 83. 校验对象是否存在 **object.exist**

2007-03-07 11:38:36 / 天气: 晴朗 / 心情: 高兴 / 个人分类: [QTP](#)

除了用校验点来确认相符的对象是否存在外,也可以用 **object** 的 **exist** 属性来判断对象的存在与否.

语法:**Object.Exist([Timeout])**

描述:

| Argument       | Type   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|----------------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>Timeout</i> | Number | <p><b>Optional.</b> The length of time to search for the object before returning a True or False value.</p> <ul style="list-style-type: none"> <li>• If a timeout value is specified, QuickTest waits until it finds the object or until the timeout is reached.</li> <li>• If the value <b>0</b> is specified, the method returns the True or False value immediately.</li> <li>• If no value is specified, the value specified in the Test Settings dialog box for the <b>Object Synchronization Timeout</b> is used for tests. For business components, the pre-defined value of 20 seconds is used.</li> </ul> <p>For more information on the <b>Object Synchronization Timeout</b>, refer to the <i>QuickTest Professional User's Guide</i>.</p> |

实例:

```

Dialog("Login").Activate

Dialog("Login").WinEdit("Agent Name:").Set "training"

Dialog("Login").WinEdit("Agent Name:").Type micTab

if Dialog("Login").WinEdit("Password:").Exist Then

 Reporter.ReportEvent micDone,"存在","控件存在"

 Dialog("Login").WinEdit("Password:").SetSecure "429376f06698f739df8bb1c09eab3d78dee316f7"

else

 Reporter.ReportEvent micDone,"不存在","控件不存在"

End If

```

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

## 84. QTP 中&和\_的用法。

2007-02-25 09:11:58 / 个人分类: [QTP](#)

*ForWriting = 2*

```
Set fso = CreateObject("scripting.FileSystemObject")
Set f = fso.OpenTextFile("c:\testfile.html", ForWriting, True)
html="HTML"
f.Write "<HTML>" &_
"<HEAD>" &_
"<TITLE>一个简单的 HTML 示例</TITLE>" &_
"</HEAD>" &_
"<BODY>" &_
"<CENTER>" &_
"<H3>欢迎光临" &_
"的主页<H3>" &_
"
" &_
"<HR>" &_
" " &_
"这是我第一次做主页，无论怎么样，我都会努力做好！" &_
""&"</CENTER>"&"</BODY>" &_
" "&"</HTML>"
wait(5)
```

f.Close

这是一段 [QTP 脚本](#)，其中使用到了&和\_。

& 负责连接字符串，当然也可以隐性把整数，浮点数等其他数据类型转换为字符串

\_ 续行连接，如某句代码一行显示过长，我们可以把代码换行写，但句末要加上"\_"符号。当语句本执行时，编译器看到了这句句末有"\_"时，就会知道语句还没

结束，只不过换行了，跳到下面继续读。

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

## 85. 选取 webList 中某一项

2007-03-12 15:31:14 / 个人分类: [QTP](#)

```
groupCodeLenth= Browser("test").Page("test").Frame("mainFrame").WebList("mainFrame").Object.length
```

Randomize

```
groupCodeNum=int(groupCodeLenth*Rnd)
```

```
msgBox"groupCodeNum= "&groupCodeNum
```

```
if(groupCodeNum>=0) then
```

```
Browser("test").Page("test").Frame("mainFrame").WebList("sgrouptitlecode").Select "#& groupCodeNum
```

```
end if
```

## 86. **object.SetCaretPos** 方法

2007-03-12 18:23:29 / 天气: 晴朗 / 心情: 平静 / 个人分类: [QTP](#)

如果想把光标放在输入框的某个指定位置,可以用 **object.SetCaretPos** 方法来实现.

**object.SetCaretPos Col**

如果是 combox,则为 **object.SetCaretPos Line, Col**

例:

```
Dialog("SingleLineEdit").WinEdit("Read only").SetCaretPos 0
```

```
Dialog("SingleLineEdit").WinEdit("Read only").Type "a example"
```

```
Dialog("SingleLineEdit").WinEdit("Read only").SetCaretPos 1
```

```
Dialog("SingleLineEdit").WinEdit("Read only").Type "n"
```

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

## 87. Similar to watir sample using description programming

描述性语言的简单运用：以下代码是在 GOOGLE 上查询关键字“软件测试”，可以将代码拷贝到 QTP 中直接运行。

```
SystemUtil.Run "C:\Program Files\InternetExplorer\IEXPLORE.EXE", "http://www.google.com"
```

```
Set browser_google = Description.Create()
Set page_google = Description.Create()
Set text_google = Description.Create()
Set button_google = Description.Create()

browser_google("name").Value="Google"
page_google("url").value="http://www.google.com/intl/zh-CN"

text_google("name").Value = "q"
text_google("type").Value = "text"
text_google("html tag").Value = "INPUT"

button_google("name").Value = "Google 搜索"
button_google("type").Value = "submit"
button_google("html tag").Value = "INPUT"

Set wb = Browser(browser_google).Page(page_google)

Fn_AboutYou()
Fn_Check()

Public function Fn_AboutYou
wb.WebEdit(text_google).Set "软件测试"
End Function

Public function Fn_Check
wb.WebButton(button_google).Click
End Function
```

## 88. QTP 使用 outlook 发邮件

2007-03-20 17:04:09 / 个人分类：[QTP](#)

因为需要把一些错误信息发给相关人员，让工具更自动化。

```
Dim objOutlook
```

```
Dim objOutlookMsg
```

```
Dim olMailItem
```

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

```

' Create the Outlook object and the new mail object.

Set objOutlook = CreateObject("Outlook.Application")

Set objOutlookMsg = objOutlook.CreateItem(olMailItem)

Set mapi = objOutlook.GetNamespace("MAPI")

'Define mail recipients

objOutlookMsg.To = "my@email.com"

objOutlookMsg.CC = "my@email.com"

'objOutlookMsg.BCC = "my@email.com"

'Body of the message

objOutlookMsg.Subject = "QTP Test Mail"

objOutlookMsg.Body = "This is a test mail"

'Display the email

objOutlookMsg.Display

'Send the message

objOutlookMsg.Send

'Release the objects

objOutlook.quit

Set objOutlook = Nothing

Set mapi = Nothing

```

## 89. QTP 对象识别和操作对象原理

### QTP 识别和操作对象的原理

文章出处: 51testing 论坛 作者: yabest 发布时间: 2007-03-28

QTP 为用户提供了两种操作对象的接口，一种就是对象的封装接口，另一种是对象的自身接口。

对象的自身接口是对象控件本身的接口，只要做过软件开发，使用过控件的人应该很清楚。

对象的封装接口是 QTP 为对象封装的另一层接口，它是 QTP 通过调用

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

对象的自身接口来实现的。

两种接口的脚本书写格式的差别在于：

自身接口需要在对象名后面加 object 再加属性名或方法名，  
封装接口就不用在对象名后面加 object。

比如操作 JavaEdit 对象，通过 QTP 封装的封装接口，脚本如下：

设置 JavaEdit 的内容：

```
JavaDialog("Add NE").JavaEdit("NE Name").Set "NE1"
```

读取 JavaEdit 的内容：

```
msgbox JavaDialog("Add NE").JavaEdit("NE
Name").GetROProperty("value")
```

如果通过 JavaEdit 的自身接口，脚本如下：

设置 JavaEdit 的内容：

```
JavaDialog("Add NE").JavaEdit("NE Name").object.setText("NE1")
```

读取 JavaEdit 的内容：

```
Msgbox JavaDialog("Add NE").JavaEdit("NE
Name").object.getText()
```

QTP 执行 JavaEdit().Set 语句时，是通过执行

JavaEdit().object.setText() 来实现的。

QTP 执行 JavaEdit().GetROProperty("value")，是通过执行

JavaEdit().object.getText() 来实现的。

JavaEdit 对象的封装接口 Set() 和 GetROProperty("value")，是 QTP 封装 JavaEdit 对象的自身接口 setText() 和 getText() 而得来的。

对象的封装接口是 QTP 使用的缺省接口，我们录制出来的脚本都是使用封装接口，大家用的也都是封装接口。

但是封装接口不如自身接口丰富，因为 QTP 只是封装了部分常用的自身接口嘛。

所以我们在需要时，可以绕过封装接口，直接调用对象的自身接口。

不过有些自身接口不够稳定，在实践中偶尔会出现问题，但是概率很少。

封装接口有相应功能的话，就尽量用封装接口吧！

理解了封装接口和自身接口的原理，我们就可以更加灵活的操作对象了。

但是我们怎么知道对象都有哪些封装接口和自身接口呢？

其实很简单，用对象查看器（Object Spy）查看对象，在查看窗口里有列出这些接口，包括属性和方法。

窗口中间有选择栏让你选择 Run-time Object 或者 Test Object，

当你选择 Run-time Object 时，它显示的就是对象的自身接口（自身的属性和方法）

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

当你选择 Test Object 时，它显示的就是对象的封装接口（封装的属性和方法）

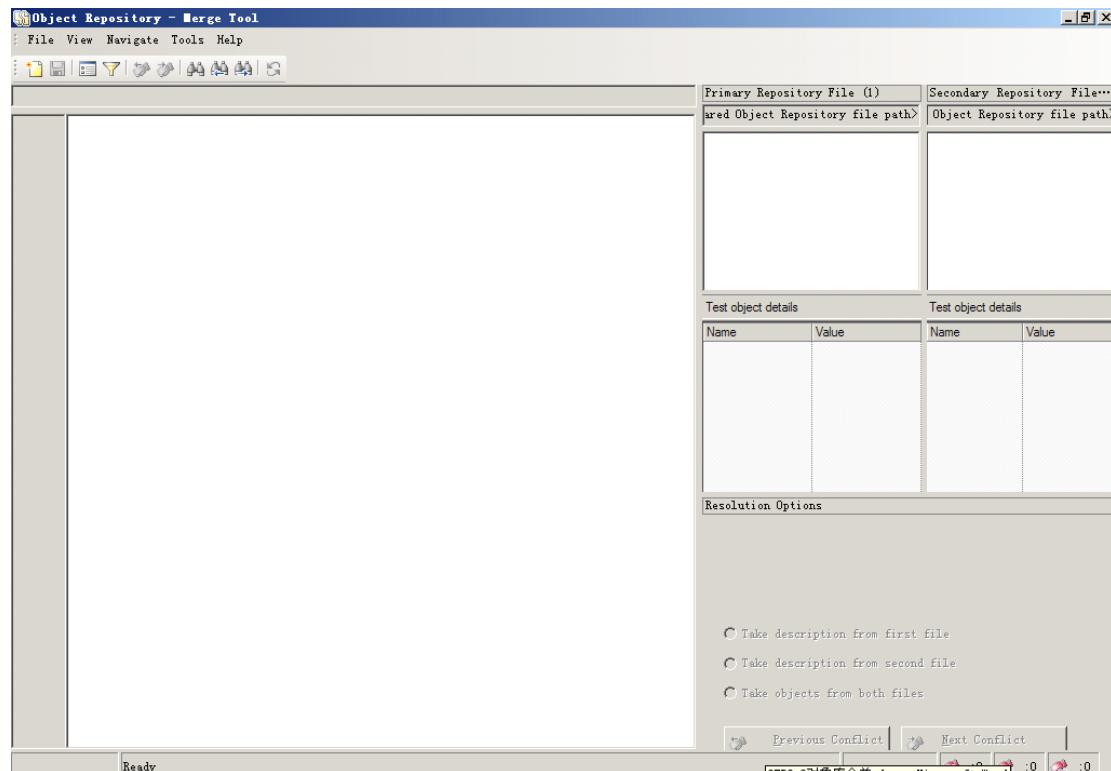
明白了这些，你还等什么呢？快拿起对象查看器，看看对象都有哪些封装接口和自身接口，肆意的操作它，玩弄它吧！

比如执行

```
JavaDialog("Add NE").JavaEdit("NE
Name").object.setVisible(false)
```

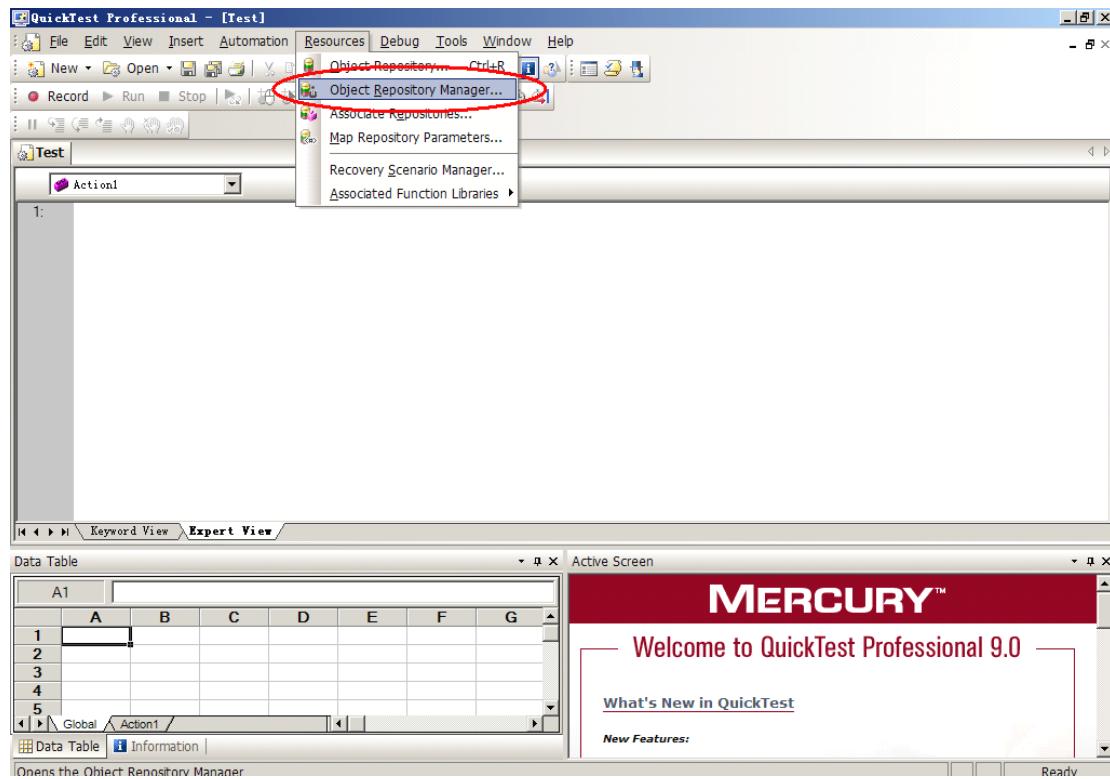
哈哈，你的 JavaEdit 对象就当场消失不见了！！！

## 90. QTP9.0 中的对象库合并是通过对象合并工具 Object Repository Merge Tool 实现的

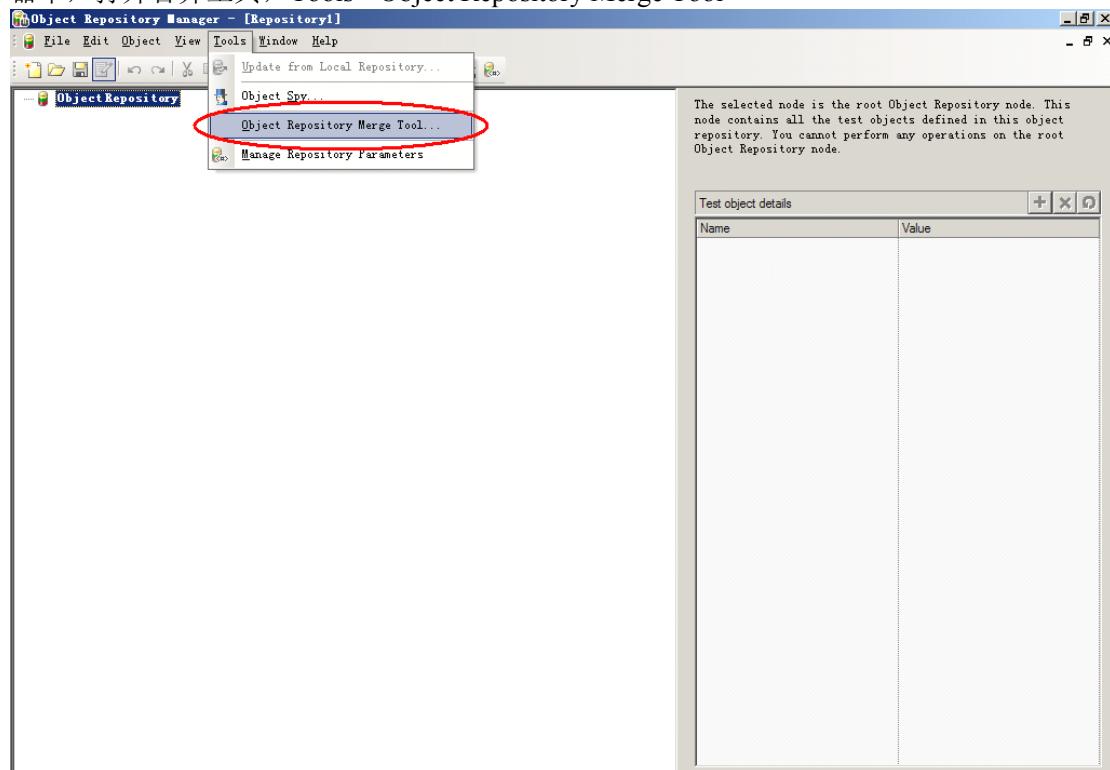


在 QTP9.0 中打开对象管理器，Resources->Object Repository Manager

|       |              |
|-------|--------------|
| 自动化测试 | Version: 1.3 |
| 文档    | 2006-5-17    |



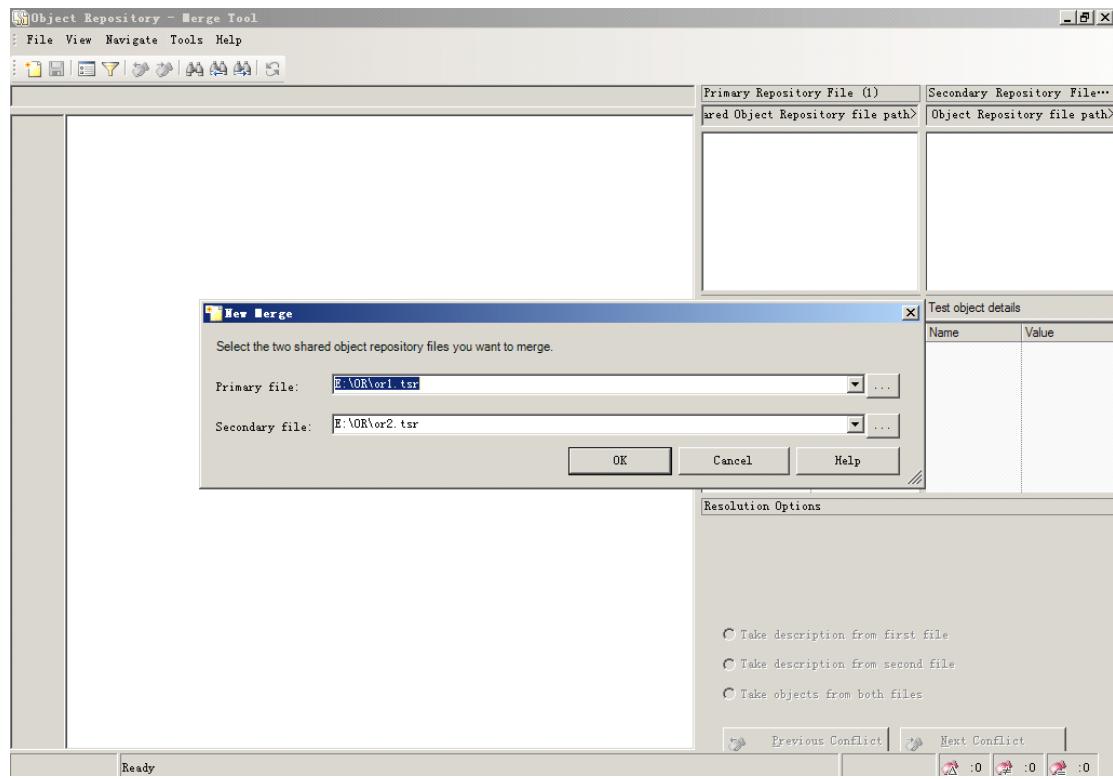
在对象管理器中，打开合并工具，Tools->Object Repository Merge Tool



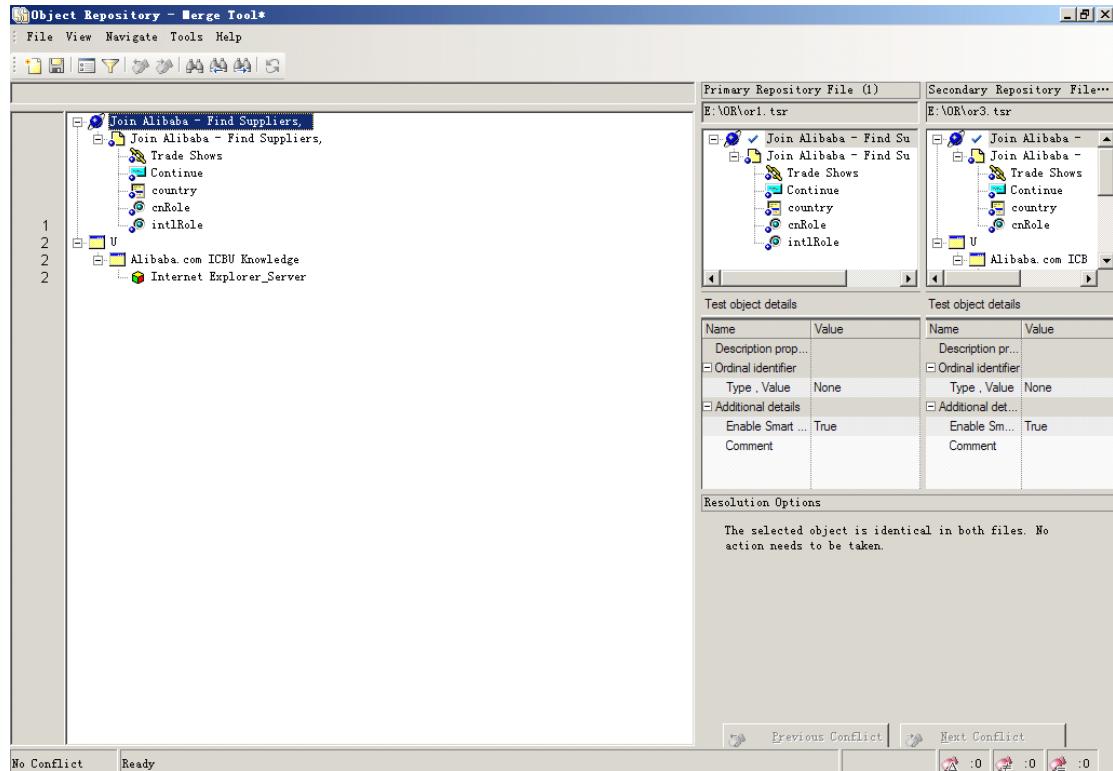
主、从对象库文件点击 ok 开始合并

选择合并的

|       |              |
|-------|--------------|
| 自动化测试 | Version: 1.3 |
| 文档    | 2006-5-17    |

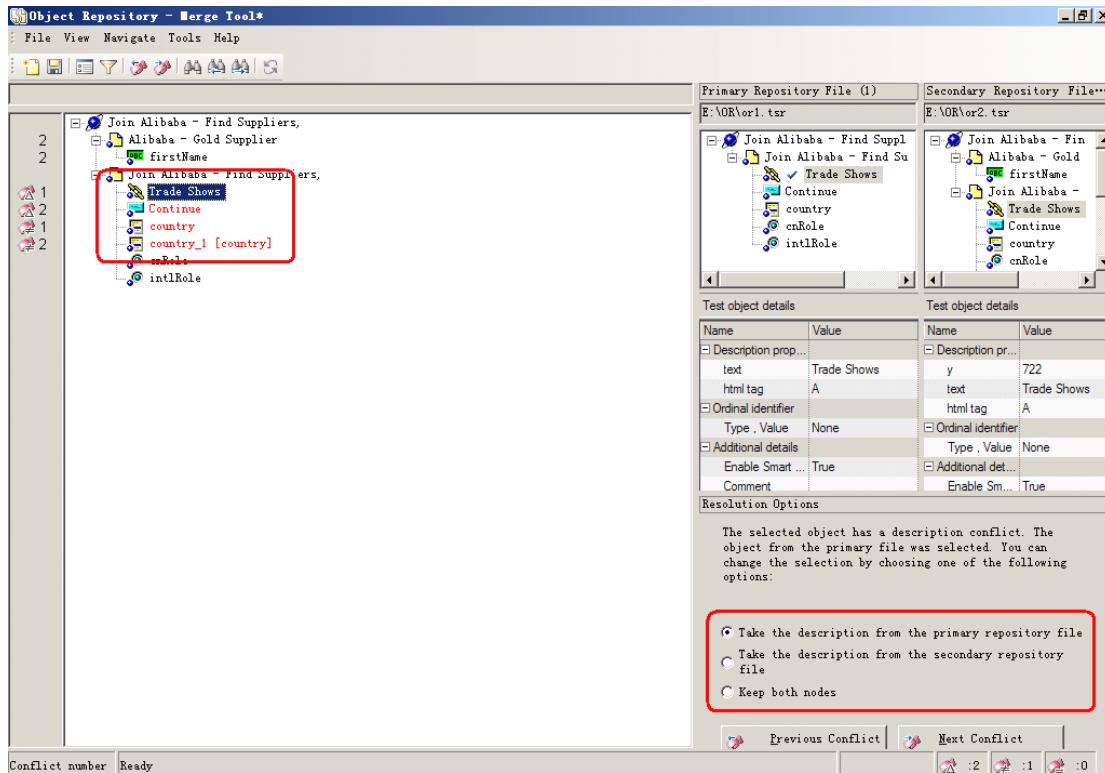


如果没有发生冲突，则可以顺利合并，合并后的对象库为主、从库的并集



如果发生冲突，则需要进行冲突处理后才能完成合并

|       |              |
|-------|--------------|
| 自动化测试 | Version: 1.3 |
| 文档    | 2006-5-17    |



发生冲突的原因是同名对象的属性不同（属性项不同或属性值不同）

QTP 提供的对象冲突解决的方法有三个：

1. 合并后保留发生冲突对象在主对象库中的对象属性
2. 合并后保留发生冲突对象在从对象库中的对象属性
3. 合并后发生冲突对象在主、从对象库中的对象属性均保留，产生两个对象名：主对象库对象名不变，从对象库对象名加“\_1”

举例：对象库表示法：

对象库 a；有对象 A、B；对象 A 有属性 A1、A2，对象 B 有属性 B1、B2；

属性值 A1=“jack”，A2=“male”，B1=“lucy”，B2=“female”。表示为

a{A(A1=“jack”,A2=“male”),B(B1=“lucy”,B2=“female”)}设存在 a, b, c 三个对象库：

a{A(A1=“jack”,A2=“male”),B(B1=“lucy”,B2=“female”)}

b{A(A1=“jack”,A2=“male”),C(C1=“lily”,C2=“female”)}

c{A(A1=“jack”,A2=“male”),B(B1=“kite”,B2=“girl”), C(C1=“lily”,C2=“female”)}对象库

a（主）与 b（从）合并：对象库合并无冲突。

合并后得到对象库 ab 为

ab{A(A1=“jack”,A2=“male”),B(B1=“lucy”,B2=“female”), C(C1=“lily”,C2=“female”)}对象库 a（主）与 c（从）合并：对象库合并后发生冲突。

冲突的对象为对象 B，属性 B1 值不同，对象库 c 中对象 B 多一个属性项 B3，按照 QTP 提供的冲突解决方法进行处理：

方法 1：合并后得到对象库 ac 为

ac{A(A1=“jack”,A2=“male”),B(B1=“lucy”,B2=“female”), C(C1=“lily”,C2=“female”)}

方法 2：合并后得到对象库 ac 为

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

ac{A(A1="jack",A2="male"),B(B1="kite",B2="female",B3="girl"), C(C1="lily",C2="female")}

方法 3：合并后得到对象库 ac 为

ac{A(A1="jack",A2="male"), B(B1="lucy",B2="female"),B\_1(B1="kite",B2="female",B3="girl"), C(C1="lily",C2="female")}

## 91. QTP 中的日期格式

### qtp 中的日期格式

文章出处: 51testing 博客 作者: 魔方十字星 发布时间: 2007-04-03

前几天的时候, 用 QTP 别一个 AX 日期控件对象的时候

不知道是对象的关系还是 QTP 的关系不可以使用 SET 方法设置? 只可以用 SETdate 来设置,

而且格式只有 d-M-yyyy, 如"7-Mar-2007"

而日期只有 2007-3-7 yyyy-mm-d,

先是有 FormatDateTime, cdate(vbs 没 format) 来转换, 我用的是简体中文系统, 所以无果

看来只好自己做, 如果自己做的话就要拼字符串, 日-短英月-长年, 日和年可以用 day 和 year 取出来

月的话就要写 select 来做吗?, 我发现短英月就是英文月份的前 3 个字符, 这样的话, 我就只要取出一个英文月份名就可以了,

如何取出英文月份名呢, 我想了, 日期格式是和区域设置有关, 幸好 vbs 中有 GetLocale 和 SetLocale, 我就方便了, 这样我先设置美国英文"en-gb", 得到

"07 March 2007"(dd MM yyyy), 看来月份是有了, 不过这个在中间, 不太方便

我就是试了试别的英语国家, 终于在加拿大"en-ca"中有了"March 07, 2007"

这样我就只要取左 3 个字符了.

```
Function MyCdate(oDate)
currentLocale = GetLocale
original = SetLocale("en-ca")

mon=left(cstr(FormatDateTime(oDate, 1)), 3)
original = SetLocale(currentLocale)
MyCdate=cstr(day(odate))+"-"+mon+"-"+cstr(year(odate))

End Function
```

此文来源于 51testing 博客, 转载请注明出处

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

原始链接:

[http://blog.51testing.com/html/28/79428\\_itemid\\_6007.html](http://blog.51testing.com/html/28/79428_itemid_6007.html)

|       |           |     |
|-------|-----------|-----|
| 自动化测试 | Version:  | 1.3 |
| 文档    | 2006-5-17 |     |

## 92. 获取内存信息

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

.....

'函数说明：获取并返回进程内存使用情况  
'参数说明：  
'     (1)T1,运行次数  
'     (2)T2,间隔时间  
'     (3)Processname,进程名称  
'     (4)sfilename,Excel 绝对路径文件名  
'前提条件：  
'     Excel 绝对路径文件名，必须事先创建成功  
' 返回结果：  
'     无  
' 调用方法：  
'     MemoryCollection(T1,T2,Processname)

.....

```
Function CreateExcel()
 Set ExcelApp = CreateObject("Excel.Application")
 ExcelApp.Workbooks.Add
 ExcelApp.Visible = True
 Set CreateExcel = ExcelApp
End Function
```

```
Function PutValueInFile (sfilePath, isheet, irow, icolumn, value)
 Set ExcelObj = CreateObject("Excel.Application")
 ExcelObj.Workbooks.Open sfilePath
 Set NewSheet = ExcelObj.Sheets.Item(isheet)
 NewSheet.Cells(irow,icolumn) = value
 ExcelObj.ActiveWorkbook.Save
 ExcelObj.Application.Quit
 Set ExcelObj = Nothing
End Function
```

```
Function GetValueInFile (sfilePath, isheet, irow, icolumn)
 Set ExcelObj = CreateObject("Excel.Application")
 ExcelObj.Workbooks.Open sfilePath
 Set NewSheet = ExcelObj.Sheets.Item(isheet)
 value = NewSheet.Cells(irow,icolumn)
 ExcelObj.ActiveWorkbook.Save
 ExcelObj.Application.Quit
 Set ExcelObj = Nothing
 GetValueInFile = value
End Function
```

```
Function MemoryCollection(T1,T2,Processname,sfilename)
 Set ExcelObj = CreateObject("Excel.Application")
```

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

```

ExcelObj.Workbooks.Open sfilename
Set NewSheet = ExcelObj.Sheets.Item(1)
Set oShell = CreateObject ("WSCript.shell")
oShell.run "Taskmgr.exe"
Dialog("regexpwndtitle:=Windows 任务管理器").Minimize
For i=1 to T1
 Memory = Dialog("regexpwndtitle:=Windows 任务管理器
").WinListView("regexpwndclass:=SysListView32").GetSubItem(ProcessName,"内存使用")
 value = date&"-&Time&" : "&ProcessName&" 内存使用情况 :"&Memory
 NewSheet.Cells(i+1,1) = value
 wait T2
Next
Set oShell = Nothing
ExcelObj.ActiveWorkbook.Save
ExcelObj.Application.Quit
Set ExcelObj = Nothing
Dialog("regexpwndtitle:=Windows 任务管理器").Close
End Function

```

MemoryCollection 1,1,"QTPro.exe" , "D:\QTPTemp\Memory State.xls"

### 93. 图片比较

#### 不用 checkpoint 的 Bitmap 比较

以下是我用文件对 BITMAP 进行比较的方法。  
但是这种方法只能比较两张图片的全部,不能比较比较图片中的部分区域.  
LCDTitle = "LCD Device(μP D 1 6 4 3 2 B)"  
LCDTitle1 = "LCD1 Device(μP D 1 6 4 3 2 B)"  
dialog(LCDTitle).CaptureBitmap "c:\lcd.bmp",true  
dialog(LCDTitle1).CaptureBitmap "c:\lcd1.bmp",true

```

Dim FS, File1, File2
FilePath1 = "c:\lcd.bmp"
FilePath2 = "c:\lcd2.bmp"
Set FS = CreateObject("Scripting.FileSystemObject")
If (FS.GetFile(FilePath1).Size <> FS.GetFile(FilePath2).Size) Then
 msgbox "FilePath1:=" & FS.GetFile(FilePath1).Size & "end"
 msgbox "FilePath2:=" & FS.GetFile(FilePath2).Size & "end"
 CompareFiles = 1
 'Exit Function
End If
Set File1 = FS.GetFile(FilePath1).OpenAsTextStream(1, 0)
Set File2 = FS.GetFile(FilePath2).OpenAsTextStream(1, 0)

CompareFiles = False
Do While File1.AtEndOfStream = False
 Str1 = File1.Read(1000)
 Str2 = File2.Read(1000)
 CompareFiles = StrComp(Str1, Str2, 0)
 If CompareFiles <> 0 Then
 CompareFiles = 1
 Exit Do
 End If
Loop

```

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

```

File1.Close()
File2.Close()
通过像素点的比较
Dim mybmp1
Dim mybmp2
Set mybmp1 = LoadPicture ("c:\lcd.bmp")
Set mybmp2 = LoadPicture ("c:\lcd1.bmp")
msgbox mybmp1.Height
msgbox mybmp1.Width
For Y = 0 To mybmp1.Height
 For X = 0 To mybmp2.Width
 fColor1 = mybmp1.Point(X, Y)
 fColor2 = mybmp2.Point(X, Y)
 If fColor1 <> fColor2 Then
 msgbox "不一样!"
 End If
 Next X
 Next Y

```

#### 94. 退出循环

ExitTest -退出整个 TEST。

ExitAction - **退出**当前操作，无论其**循环**属性如何。

ExitActionIteration - **退出**操作的**当前循环**。

ExitRun - **退出测试**，无论其**循环**属性如何。

ExitGlobalIteration - **退出**当前全局**循环**。

定位 XLS 文件中的单元格可以直接用 CELL 方法

定位 GlobalSheet，或 Action 表中的单元格可以这样

先用 setcurrentrow 方法定位行，再用 datatable.RawValue(i, "Action1") 方法定位列  
i 可以是列名，也可以是列号，例第一列，i 可以是 A 也可以是 1, 是 A 时需加双引号，  
Action1 是表名

#### 95.一些鼠标相关事件

[QTP](#)

右键双击事件

object.FireEvent "ondblclick",,, 11

注解：FireEvent "ondblclick",,, 11

object.FireEvent EventName, [x], [y], [BUTTON]

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

BUTTON 是可选的. 可以定义为常量或者数字.

关于常量的描述:

0 是鼠标左键

1 是右键

2 是中间键

tool->web event recording configuration 相关设置

用 object spy 看菜单下选项的属性里按住 ctrl 键选择, 否则选不上, 不能查看

tools -> web event recording configuration 点击 custom settings... 展开 Web objects, 选种 webelement, 点击 event -> add -> onmouseover

## 96. [总结]QTP Timeout Settings

### QTP

1. **Object Sync Timeout**: 这是 QTP 在等待一个对象显示的时间

具体设置方法: Test Settings, Run Tab, the "Object Synchronization timeout:" setting.

请注意在 8.x 版本中单位是毫秒, 而在 9.x 就变成秒了。

程序中设置: Setting("DefaultTimeout") [=milliseconds]

2. **Browser Navigation Timeout** 帮助里这样定义 “sets the maximum time (in seconds) that QuickTest waits for a Web page to load before running a step in the test”. 那和上面的 Object Sync Timeout setting 有什么区别的, MI 没有说, 我猜想是在 [测试](#)一个显示很缓慢的页面的时候使用吧。针对 Navigation 而不是一个对, 从脚本看:

**Browser("Browser").Page("Page").WebRadioGroup("Name:=txt\_Name","html tag:=INPUT").set "Test"**

这可能是一开始 Browser 的等待时间吧。

具体设置方法: Test Settings, Web tab, "Browser navigation timeout:" X "seconds"

在程序中怎么实现还没有解决, 谁可以帮助我?

3. **Default Load Time** 在 QTP help 这样定义 “Instructs QuickTest to add a specified number of seconds to the page load time property specified in each Page checkpoint.”. 很迷惑, 按照帮助的理解就是检查点额外的等待时间, 过了这个时间再没有出现检查的数据可能就会返回错误了。

具体设置方法: Options, Web Tab, "Add" X "seconds to page load time"

程序中设置: Setting("DefaultLoadTime") [=seconds]

4. **Activate Window** 定义“specifies the time (in tenths of a second) that QuickTest waits before it sets the focus on an application window when using the pointing hand to point to an object in the application (for Object Spy, checkpoints, Step Generator, Recovery Scenario Wizard, and so forth)”. 我一直使用的默认的值, 挺好的。呵呵。

具体设置方法: Options, General tab, "When pointing at a window, activate it after" X "tenths of a second"

程序中设置, 暂无。

5. **.Exists(Timeout)** 用的是毫秒作为单位.

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

6. **.WaitProperty(x, y, Timeout)** 用的是毫秒作为单位.

### 7. Wait seconds [, milliseconds]

这个就是我们经常用的 wait, 不过后面可以选择毫秒可能有的人就不知道了。

以上是我总结的一些 QTP 关于时间的设置的, 很多概念自己也不算清楚, 希望和大家一起在学习中进步。

## 97.QTP 中读取 Access 数据

[关于测试](#)

以下是我写的 [QTP](#) 脚本, 目的是用来读取 Access 数据库中的数据, 代码如下:

```
Option explicit
Dim Cnn , strDb, Rst, strCnn, iLen '//定义变量
strCnn="Provider=Microsoft.Jet.OLEDB.4.0;OLE DB Services=-4;Data
Source=D:\test\calc.mdb" //将获得的连接字符串赋值到
Set Cnn=CreateObject("ADODB.Connection") //生成数据库连接对象
Cnn.Open strCnn //打开数据库
Set Rst=CreateObject("ADODB.Recordset") //生成记录集对象
Rst.Open "select * from calc" ,Cnn
Do While Not Rst.EOF
 For iLen=1 to 3 //我的表中只有 3 个字段, 当然这种方法通用性不是很好
 Select Case iLen
 Case 1
 strDb=strDb&Rst.Fields.item(iLen)&" "
 Case 2
 strDb=strDb&" "&Rst.Fields.item(iLen)&" "
 Case 3
 strDb=strDb& "=" &Rst.Fields.item(iLen)
 End Select
 Next
 strDb=strDb&chr(13)
 Rst.MoveNext
Loop
Rst.Close
cnn.close
msgbox strDb
希望这对 QTP 新手找这方面资料时有点帮助.
```

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

## 98. 解决 AJAX 写的控件不能录制方法

QTP

Tools --> Web Event Recording Configuration, 把 Event configuration level 的设置从默认的 Basic 改为 Medium , 然后 关闭 QTP, 重新录制你的脚本就可以了。

<http://www.microsoft.com/china/technet/archives/columns/scripts/sg1002.asp>

[code]

让我们立即着手编写一个示例脚本。事实上，我们应该在编写脚本之前，先确保具备一个用来保存这个脚本的物理位置。如果在您所使用的计算机上并不存在一个名为 C:\scripts 的文件夹，就请返回根目录，并新建这个文件夹。当您完成上述工作后，应打开记事本或其它文本编辑器，并键入以下脚本代码，然后，使用 RunIPConfig.vbs 文件名将其保存至您的 C:\scripts 文件夹。

```
Set objShell = CreateObject("Wscript.Shell")
objShell.Run "ipconfig"
```

在尝试运行刚刚生成的脚本之前，让我们先对其进行检查，并设法事先指出可能产生的运行结果。在这个脚本中存在着两个有助于我们预测运行结果的暗示。第一行代码中包括动词 Create（生成，实际上是 CreateObject），而第二行代码则包含动词 Run（运行）。可见，这个脚本将有可能先生成某种被称作对象的东西，并随后运行某一特定程序。事实上，情况看上去更象是该脚本将运行一个文件名为 ipconfig.exe 的命令行工具。

以上预测的确与该脚本所产生的运行结果相符。该脚本首先生成一种被称作对象的东西。在这个示例中，脚本所生成的对象是 Shell (WshShell) 对象。您可将某一对象理解成类似于命令行工具的东西；而您的脚本则可借助它执行某一特定类型的任务。Shell 对象将允许您从脚本中执行原本可在 Windows 命令解释程序内执行的任务——就像运行程序一样。

与命令行工具不同的是，您必须在开始对其加以应用之前，先行生成一个新对象。当生成该对象时，还应为它指定一个将在脚本其余部分中被用来对它进行引用的名称。在这个示例中，我们所使用的名称是 objShell。当对象生成完毕并被赋予相关名称后，我们便可利用该对象执行相关任务，具体方法为：在对象名后加上一个圆点（.），并在圆点后加上所需完成的任务名称。这个例子中的任务名是 Run。而 Run 则是众所周知的 Shell 对象方法。

现在，请打开一个命令行窗体，并导航至 C:\scripts 文件夹。如需运行您刚刚编写的脚本，则请输入 cscript RunIPConfig.vbs 命令，并按回车键。您将看到一个窗体在屏幕上一闪而过，但该窗体所显示的内容却在您能够看清之前消失了！您可反复尝试运行该脚本，并试图破解消失在窗体中的信息。这也许是一件饶有兴味的事，但却可通过一种更加简便的方法完成。

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

为了便于您进行理解，我们必须针对在您运行命令行工具的同时所发生的事件给予关注。当您在命令行方式下输入 ipconfig、并在键盘上敲击回车键时，便对命令解释程序发出了一个指令。而命令解释程序则将对您所输入的指令进行检查。如果您所输入的命令代表着一个有效程序，那么，命令解释程序便会为您运行该程序。如果您所输入的命令并不代表有效程序，那么，命令解释程序则将通过显示错误信息的方式就有关情况向您进行告知。基于 Windows NT 操作系统的命令解释程序是 Cmd. exe。而 Windows 9x 操作系统中的命令解释程序则是 Command. exe。

此时此刻，上述脚本中的 objShell.Run “ipconfig” 命令将在其运行 ipconfig 程序的同时绕过命令解释程序；换句话说，该命令将在无须先运行命令解释程序再由命令解释程序运行 ipconfig 的前提下直接运行 ipconfig。在某些情况下，这种方式或许利大于弊，但在这个示例中，我们却并不希望绕过命令解释程序。为什么？因为命令解释程序可帮助我们发现“窗体一闪而过”问题的答案。正如您所看到的，命令解释程序可接受大量开关选项（如果您有兴趣了解 cmd. exe 或 command. exe 所具备的全部功能，则请在计算机帮助系统中查找该程序的可执行文件），而这些选项中的/k 开关则恰恰用来指示命令解释程序在某一程序运行完毕后保持输出窗口的开启状态。我们需要借助这个选项来确保程序窗体不会在打开后立即关闭。

为实现上述目的，我们应将正在运行的命令从 ipconfig 修改为 cmd. exe /k ipconfig 或 command. exe /k ipconfig，这里，cmd. exe 对应于 Windows NT 操作系统，而 command. exe 则适用于 Windows 9x 操作系统。但是，如果您必须同时面向 Windows NT 和 Windows 9x 操作系统提供支持，又将如之奈何？针对我们正在使用的[操作系统](#)进行跟踪的确是一件颇为麻烦的事。也正是出于这种原因，我们才引用了一个名为%COMSPEC%的环境变量，以期提供通往适当命令解释程序的完整路径。该环境变量可确保下列脚本代码既可在 Windows NT 操作系统上运行，又能在 Windows 9x 操作系统上执行：

```
Set objShell = CreateObject("Wscript.Shell")
objShell.Run "%COMSPEC% /k ipconfig"
```

请尝试运行这个经过修改的脚本。您只需启动记事本（或其它习惯使用的文件编辑器），并将%COMSPEC% /k 添加到先前编写的脚本中，然后，将修改过的文件另存为 RunIPConfig.vbs。接下来，在命令行窗体中导航至 C:\scripts 目录，再通过输入 cscript RunIPConfig.vbs 命令并按回车键的方法运行该脚本。您这次将看到一个用来显示 ipconfig 命令输出结果的新窗体，具体情形如下图所示。

如果您所使用的浏览器无法支持嵌入式框架，请，[点击此处](#) 以便在单独页面中进行浏览。

您接下来应该做的就是进行适用性测验。上述脚本所能运行的程序绝不仅限于 ipconfig。而您所需要完成的全部工作仅仅是修改第二行代码中跟在 Run 方法后面的字符串。您可尝试将%COMSPEC% /k ipconfig 分别替换为%COMSPEC% /k netstat 或%COMSPEC%

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

/k mmc。举例来说，您将可从某一脚本中运行 GUI（图形用户界面）工具，而这个脚本则以运行计算器程序为例：

```
Set objShell = CreateObject("Wscript.Shell")
objShell.Run "Calc.exe"
```

您不仅能从某一脚本内部运行其它脚本，而且，还可从当前脚本中调用其它批处理文件；如果您碰巧拥有一些批处理文件，不妨尝试运行它们。当然，如果这些批处理文件的存储位置与您计算机上 PATH 环境变量的设定范围并不相符，您就应将相关存储位置添加到全局访问路径或单独指定通往批处理脚本的完整访问路径。举例来说，如果您在 C:\Batchscripts 目录下拥有一个名为 Checkserver.bat 的批处理脚本，那么，应由您传递给 Run 方法的字符串就应类似于：%COMSPEC% /k C:\batchscripts\checkserver.bat。

如果您已完成了上述试验，我们就将继续讲解后面的内容。您将学习到借助 Run 方法运行单一程序的具体方法。如果您需要连续运行两个程序，应如何是好？这恰恰是由下列脚本即将执行的任务：该脚本将在运行 ping 程序后，紧接着运行 nslookup 程序。请注意，您可将对应于所需运行程序的命令行参数包含在传递给 Run 方法的字符串当中。在此，我们将把 127.0.0.1 作为参数分别传递给 Ping 和 Nslookup。

```
Set objShell = CreateObject("Wscript.Shell")
objShell.Run "%COMSPEC% /k ping 127.0.0.1"
objShell.Run "%COMSPEC% /k nslookup 127.0.0.1"
```

请按针对先前所述脚本的处理方法保存并运行这个脚本。脚本运行结果应该是两个命令行窗体同时打开，其中，Nslookup 程序在一个窗体中运行，而 Ping 程序则在另一个窗体中以同步方式运行，具体情形正如以下屏幕快照所示。

如果您所使用的浏览器无法支持嵌入式框架，请，[点击此处](#) 以便在单独页面中进行浏览。

在这个例子中，事实表明，实现 Ping 和 Nslookup 的同时运行根本不成问题。然而，如果您需要运行两个批处理脚本，并希望在第一个脚本运行完毕后再开始运行第二个脚本，又该如何是好？举例来说，第一个程序将针对某一特定类型的信息执行检索，并将其保存为一个文本文件；而第二个程序则将在前者基础上通读刚刚生成的文本文件，并针对相关内容执行特定操作。在这种情况下，两个程序的同时运行将是无法接受的；只有在第一个程序执行完毕的前提下，第二个程序才能开始运行。

到目前为止，我们一直都在传递给 Run 方法一个用来指定所需运行程序的字符串。Run 方法将可再接收两个参数，而这两个参数中的每一个均为可选参数。第二个参数将允许您针对某一程序在开始运行时所呈现的窗体外观加以指定。并非所有程序都将使用到该参数，但对于那些使用该参数的程序来说，该参数却可帮助您启动一个呈现多种窗体风格（包括最小化、最大化和隐藏在内）的程序。我们并不打算在这篇专栏文章中过多涉及该参数。如果您需要了解有关该参数的更多信息资料，敬请查阅 WSH 在线文档

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

<http://msdn.microsoft.com/library/default.asp?url=/nhp/Default.asp?contentid=28001169> (点击接近本页面顶部的“Windows script Host 文档”链接)。

专供 Run 方法接收的第三个参数将允许我们防止相关程序以并发方式运行。您可将第三个参数赋值为 True，以便指示脚本停留在使用 Run 方法的代码行并保持等待状态，直到已被激活的程序运行完毕。我们已经知道了因未设定该参数所产生的后果，该参数在缺省状态下被设定为 False，而这种设置必将导致两个程序同时运行。

让我们将前面的脚本修改为先运行 Ping 程序，待 Ping 程序运行完毕后，再运行 Nslookup 程序的状态。

```
Set objShell = CreateObject("Wscript.Shell")
objShell.Run "%COMSPEC% /k ping 127.0.0.1",,True
objShell.Run "%COMSPEC% /k nslookup 127.0.0.1"
```

请注意，我们已将被赋值为 True 的第三个参数添加到应在脚本第二行中被调用的 Run 方法之后。我们之所以连续使用两个逗号的原因在于，如果我们只使用一个逗号，那么，Run 方法便会认为我们随后指定的是第二个参数，而非第三个参数。如您所见，两个连续逗号之间并不存在任何内容，因此，也就不会有任何设置被当作第二个参数传递给 Run 方法。如果我们没有对某一参数专门赋值，那么，Run 方法便会使用缺省值。举例来说，第三个参数的缺省值应该是 False。

现在就让我们试着运行经过修改的脚本。您会马上注意到，只有 Ping 程序得到运行。当 Ping 运行完毕后，该程序运行时所使用的窗体仍在屏幕上保持开启状态；而 Nslookup 程序则尚未开始运行。请试着关闭这个窗体并查看接下来发生的情况。当您关闭 Ping 程序所使用的窗体后，Nslookup 便会开始运行。请记住，您正在使用/k 选项运行 cmd.exe (或 command.exe)，也就是说，您已经告诉 Run 方法，直到 cmd.exe 运行完毕，方可继续执行后续指令。好的，由于我们事先设定了/k 选项，因此，只有曾被要求保持开启状态的窗体得到关闭，cmd.exe 才会执行完毕。

如果您需要运行两个批处理文件，并且无需查看它们的输出结果，那么，便可将/k 选项删除，而这两个批处理文件仍将接续（而非同时）运行，这也恰恰是您在第二个程序信赖第一个程序运行结果的前提下所需确保的运行方式。

两个程序之间应就以下信息进行沟通，那就是，第一个程序实现运行的情况必须被告知第二个程序。某些程序将返回一个错误代码（也就是一个数字），用来指示自身是否已得到成功运行。而 Run 方法便可将这种错误代码返回给您所编写的脚本。正处于运行状态的程序必须在得知即将被返回的错误代码之前运行完毕，为此，您必须通过将 Run 方法的第三个参数设定为 True 的方式强制脚本等候程序运行完毕。

我们将在接下来的脚本中运行 Ping 程序，并通过针对 Wscript 对象使用 Echo 命令的方式将该程序所返回的错误代码显示出来。Wscript 对象将不必被生成为类似于 WshShell 对象的状况，而您则可在脚本当中的任意控制点对它所提供的方法加以运用。

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

```
Set objShell = CreateObject("Wscript.Shell")
iErrorCode = objShell.Run("ping 127.0.0.1", , True)
Wscript.Echo iErrorCode
```

除具备错误代码检索功能外，以上脚本还在两方面与先前脚本有所区别。首先，请注意，即将被传递给 Run 方法的字符串已被加上了一对圆括号。这主要是因为，Run 目前被要求返回一个数值。这属于 VBScript 语言的一项要求；如果您并未在此使用圆括号，便会在脚本运行过程中收到一条错误信息。应该被返回的数值是 Ping 程序的错误代码，并将被保存至变量 iErrorCode，然后，借助 Wscript.Echo 命令被显示出来。

需要给予关注的第二件事就是，我们并没有运行命令解释程序。我们希望错误代码来自 Ping，而非来自 Cmd.exe（或 Command.exe），为此，我们直接运行了 Ping 程序。而由 Ping 返回的错误代码既不是很有趣，也不是很有用。当然，某些命令行工具却可返回可供您所编写的脚本做出相关判断（或生成日志）的有用错误代码。

从借助 Run 实现运行的某一程序中所获得的相关信息仅能满足最低使用需求，并很有可能不具备多少使用价值。举例来说，许多程序都将只返回一个 0 数值，以此表明自身已实现运行并且运行完毕。当然，它们并没有必要就自身成功运行且运行完毕的情况向您进行汇报。

当然，如果您已拥有 WSH 5.6 版（可从 <http://msdn.microsoft.com/library/default.asp?url=/nhp/Default.asp?contentid=28001169> 页面下载），便具备了另一种可供使用的方法，以便运行那些可帮助您与当前所运行程序改善信息交流方式的程序。而这种方法便是 Exec。它还是从属于 WshShell 对象的一种方法，这里所说的 WshShell 正是为使用 Run 而必须生成的那个对象。这对我们来说主要意味着，不必为开始利用 Exec 方法而针对脚本的第一行代码进行修改。

让我们直接编写一个利用 Exec 方法运行 ipconfig 程序的脚本。

```
Set objShell = CreateObject("Wscript.Shell")
objShell.Exec "%COMSPEC% /k ipconfig"
```

上述脚本同我们为您展示的第二个脚本之间所存在的唯一区别就是，我们已在这个脚本中将 objShell.Run 修改为 objShell.Exec。然而，当您运行这一个脚本时，将难以觉察到上述修改所产生的效果。您甚至无法看到那个一闪即逝的窗体。实际上，您根本看不到任何东西。如果我们此时告诉您，ipconfig 实际上已运行完毕，不知您是否愿意相信？好吧，还是不要轻易相信我们所说的话；取而代之的做法是，在 ipconfig 命令后方添加一个重定向符号(>)，并在针对 ipconfig 发出的调用后紧跟一个文本文件名。上述代码将导致命令执行结果被输出到指定文件。如果您尚未习惯使用重定向操作，不妨这样设想：如果我们将真实情况告诉您，而 ipconfig 程序的确处于运行状态，那么，它便会生成自己的标准文本显示信息。遗憾的是，我们将无法看到相关信息。重定向操作将允许我们捕捉到任何命令的输出结果，并将它们强行导向一个指定文件。这将有助于我们针对

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

ipconfig 命令是否处于运行状态加以确认。接下来的脚本可供用来将所需查看的输出结果呈现给我们。

```
Set objShell = CreateObject("Wscript.Shell")
objShell.Exec "%COMSPEC% /k ipconfig > ipconfig_output.txt"
```

在上述脚本运行完毕后，请查看 ipconfig\_output.txt 文件内容。如需查看这个文本文件的内容，则请按下列屏幕快照所示键入 type 命令。

如果您所使用的浏览器无法支持嵌入式框架，请，[点击此处](#) 以便在单独页面中进行浏览。

这就是您所得到的信息！我们的脚本就是实实在在地运行 ipconfig，只不过没有让我们看到命令输出结果。无论您相信与否，这都算得上一件好事。Exec 之所以没有为我们显示输出结果，恰恰是因为我们原本并未提出相关要求。正如您所看到的那样，我们可借助 Exec 针对输出结果的呈现和操作方式加以控制。

您是否还记得令 Run 方法返回错误代码的具体实现方式？Exec 事实上可返回一个被称作 WshScriptExec 的完整对象。正如 WshShell 对象为我们提供针对 Run 和 Exec 这两种方法的访问调用权限那样，WshScriptExec 对象将为我们提供针对附加功能特性的访问调用权限，而这些附加功能特性则允许我们围绕某一程序的输出结果展开操作。脚本中大量操作的实现都会以额外复杂性为代价，而上述附加功能特性也不例外。我们即将或多或少地遭遇这种复杂性，为此，请继续阅读本文，并在接下来的章节中确保集中精力。

Exec 方法可供用来返回一个 WshScriptExec 对象。而在某一脚本中促成这一事件的发生则需要使用 VBScript 语言中的 Set 关键字，具体代码如下所示：

```
Set objWshShellExec = objShell.Exec("ipconfig")
```

**说明：**我们并未在此继续使用%COMSPEC%环境变量；该变量已不具备继续存在的必要，其原因主要体现为，我们不需要为确保某一并不存在的窗体处于开启状态而将/k 选项传递给相关命令。那么，我们又该如何将这个窗体关闭并继续运行脚本代码呢？我们将可在此使用/c 选项（该选项表示只运行一条命令，然后，接着执行后续代码）；事实上，如果我们需要运行一条已被内建于命令解释程序的命令（例如 dir），便会需要使用该选项。而像 dir 这样的工具本身并不属于工具范畴；它们更加近似于 cmd.exe（或 command.exe）命令的参数；正因如此，您才需要为确保它们发挥作用而指定%COMSPEC%这一环境变量。

当您将与 WshShellExec 对象相关的引用存储于 objWshShellExec 变量后，便已具备了开始围绕程序输出结果展开操作的必要条件，但此时的条件尚不够充分。而这正是复杂性乘虚而入的大好时机。正如 Exec 方法返回 WshShellExec 对象那样，WshShellExec 对象亦可返回另一对象；这个对象将通过调用 StdOut 属性的方式被返回，而这恰恰允许您针对程序输出结果执行相关操作。

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

如果不对照一个示例脚本，上述思路几乎不可能得到理解。为此，我们专门编写了一个最为简单的示例，旨在针对由嵌套于 WSH 脚本的某一程序所产生的输出结果加以运用。请将该脚本当作一种特殊类型的重定向操作。只有在这种情况下，程序输出结果才不会被导入某一文件，取而代之的是，相关输出结果将被赋予一个名为 strOutput 的变量。遗憾的是，我们这回将不能仅仅使用一个简单的重定向符号 (>)，而必须围绕两个不同对象执行操作。

```
Set objShell = CreateObject("Wscript.Shell")
Set objWshscriptExec = objShell.Exec("ipconfig")
Set objStdOut = objWshscriptExec.StdOut
strOutput = objStdOut.ReadAll
Wscript.Echo strOutput
```

由上述脚本生成的输出结果将被显示在专供脚本运行的窗体内；这里并没有新的窗体生成。事实上，如果您从命令行方式中运行 ipconfig 本身的话，输出结果看上去也将别无二帜。既然如此，为何不干脆采取这种方式？我们已经在脚本当中为输出结果生成了一个中间存储位置—strOutput 变量。尽管我们将在上述脚本的下一行内利用 Wscript.Echo 命令针对输出结果加以显示，然而，却往往需要在执行显示操作前先对输出结果实施操控，只有这样，才能对由 ipconfig 工具显示的信息进行修改。

在尝试运行上述脚本之前，让我们先对它的最后三行代码给予更多关注；当然，您已经对前两行代码非常熟悉了。在第三行代码中，WshscriptExec 对象所配备的 StdOut 属性将被用来检索代表正处于运行状态之程序（标准）输出结果的相关对象；而 ipconfig 则是本例中待处理结果的输出程序。在第四行代码中，ReadAll 方法将被用来针对全部输出结果执行检索，并将其保存于 strOutput 变量。最后，也就是在第五行代码中，Wscript.Echo 将被用来显示输出结果。

我们现在已拥有了经常被当作 ipconfig 工具（或可供我们用来在上述脚本中替代 ipconfig 的其它任何工具）封装程序的对象。封装程序是一种可就另一程序功能特性加以应用、并能够以简便快捷方式针对已被封装程序功能进行修改的特殊程序。封装程序为我们创造了针对已被封装程序的输出结果进行修改的理想时机。而这恰恰是我们将利用本文剩余篇幅加以介绍的内容。

在围绕 ipconfig 工具开展自定义的过程中，我们将需要了解以每次一行的频率对程序输出结果进行相关处理的具体方法。当前，我们主要借助 objStdOut.ReadAll 对 ipconfig 工具的全部输出结果执行检索操作，并将检索结果保存于 strOutput 变量。而我们将在下一个脚本中执行的任务则是利用 ReadLine（而非 ReadAll）针对输出结果进行逐行检索。我们将对输出结果的每一行进行检查，如果任意一行包括“Physical Address”（物理地址）字符串，就会将它显示出来；否则，便不会显示这行输出结果。我们只对包含“Physical Address”（物理地址，也就是对应于网卡的 MAC 地址）字符串的输出结果行执行了筛选操作，并未将其它任何内容显示出来。

```
Set objShell = CreateObject("Wscript.Shell")
```

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

```

Set objWshscriptExec = objShell.Exec("ipconfig /all")
Set objStdOut = objWshscriptExec.StdOut

While Not objStdOut.AtEndOfStream
 strLine = objStdOut.ReadLine
 If InStr(strLine, "Physical Address") Then
 Wscript.Echo strLine
 End If
Wend

```

上述脚本的前三行看上去多少有些眼熟；请注意，我们必须将/all 选项添加至 ipconfig 命令后方，以确保 Physical Address（物理地址）被显示出来。这个脚本同上个脚本之间的主要区别在于，前者使用了 While/Wend 循环语句。我们将使用 ReadLine 方法对单个输出结果行执行检索操作，但却需要掌握针对是否超出行检索范围的状态加以判定的具体手段。而这恰恰是 While/Wend 循环和 objStdOut.AtEndOfStream 指令的用武之地。只要程序尚未触及（标准）输出结果数据流的结尾，上述循环便会保持接续执行状态（而符合检索条件的输出结果行则将被保存至 strLine 变量）。您可将一个数据流想象为由一系列字符排列而成的长串，而您则可按排列顺序对其进行相关处理；本例中的数据流便是 ipconfig /all 命令的完整输出结果。

每当我们完成一次程序循环时，便会将下一行内容保存至 strLine 变量，并利用 If/Then 语句决定是否应对此行内容加以显示。我们主要借助 InStr（字符串命令）功能对存储在 strLine 变量当中的输出结果行是否包含“Physical Address”（物理地址）加以判断。如果当前行确实包含有“Physical Address”（物理地址），就应使用 Wscript.Echo 命令将它显示出来。以下屏幕快照展现了由脚本生成的部分典型输出结果。

如果您所使用的浏览器无法支持嵌入式框架，请，[点击此处](#) 以便在单独页面中进行浏览。

我们的下一个脚本将使用与上一个脚本完全相同的基本组件，所不同的是，我们已不再针对输出结果执行筛选操作，而是向它们当中添加特定内容。鉴于 ipconfig /all 命令可针对 DHCP 释放获取时机及其超时状态信息执行检索，因此，将当前日期和时间包含于命令输出结果的处理方法或许具备一定实用价值。接下来的脚本正是用来实现这一设想。

脚本一开始便针对输出结果的前四行执行检索和显示操作，并且对每一行所采取的方法完全相同——先调用 ReadLine 命令，紧接着调用 Wscript.Echo 命令。当输出结果的第四行已被检索并显示出来后，我们便生成了一个包含有当前日期和时间标注信息的字符串。而这里所说的当前日期和时间则是由 VBscript 所配备的 Now() 函数提供的。

```

Set objShell = CreateObject("Wscript.Shell")
Set objWshscriptExec = objShell.Exec("ipconfig /all")
Set objStdOut = objWshscriptExec.StdOut

```

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

```

,
' Skip first four lines
,
strLine = objStdOut.ReadLine
Wscript.Echo strLine

,
' Add date/time information
,
strCurrentTime = " Current Date/Time. : " & Now()
Wscript.Echo strCurrentTime

,
' Display the rest of the output
,
While Not objStdOut.AtEndOfStream
 strLine = objStdOut.ReadLine
 Wscript.Echo strLine
Wend

```

毋庸置疑，上述脚本已开始变得多少有些复杂了。而解读这一脚本的最佳方法则是逐段分析。您应对该脚本的前三行较为熟悉。而下一个附有注释的部分则主要借助 ReadLine 和 Echo 命令对 ipconfig /all 输出结果的前四行执行读取和显示操作。

附有注释的下一部分自然以编排将被临时存储于 strCurrentTime 变量的新信息行为起点。这个信息行将由一个与 VBScript Now() 函数调用操作相关联的简单字符串构成，而 VBScript Now() 函数则用来提供当前日期和时间信息。一旦这个信息行编排完毕并被保存于 strCurrentTime 变量，我们便可当即使用 Wscript.Echo 命令对其加以显示。

在最后一个附有注释的部分中，我们将对输出结果的剩余内容执行循环操作，并在无需进行任何修改的前提下对其进行逐行显示。以下屏幕快照展现了脚本运行产生的典型输出结果。请注意，脚本输出结果看上去与 ipconfig 的标准输出结果别无二帜。而由我们插入

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

Host Name (主机名称) 行下方的 Current Date/Time (当前日期/时间) 行则与全新的上下文环境结合得天衣无缝。

如果您所使用的浏览器无法支持嵌入式框架, 请, [点击此处](#) 以便在单独页面中进行浏览。

最后, 让我们使用 For/Next 循环语句执行前四行内容的输出操作, 以便使脚本变得更加精简, 具体代码如下所示。

```

Set objShell = CreateObject("Wscript.Shell")
Set objWshscriptExec = objShell.Exec("ipconfig /all")
Set objStdOut = objWshscriptExec.StdOut

,
' Skip first four lines
,

For i = 1 To 4
 strLine = objStdOut.ReadLine
 Wscript.Echo strLine
Next

,
' Add date/time information
,

strCurrentTime = " Current Date/Time. : " & Now()
Wscript.Echo strCurrentTime

,
' Display the rest of the output
,


```

```

While Not objStdOut.AtEndOfStream
 strLine = objStdOut.ReadLine
 Wscript.Echo strLine
Wend
```

事实上, 还存在着从 WSH 脚本中运行其它程序的另一种替代方法: 即针对 Windows 管理规范 (WMI) 加以应用。您将可借助 WMI 完成一系列 Windows 管理任务! 虽然借助 WMI 实现程序运行的方式将涉及相对复杂的语法, 但却可帮助您完成 Run 和 Exec 无法实现的任务, 例如, 设定程序运行优先级或基于远程计算机运行相关程序。在此, 我们诚恳地建议

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

您尽可能了解有关 WMI 的更多信息资料。如果您确实对此抱有兴趣（我们希望您能够这样），则请到 MSDN 网站上一个名为 Scripting Clinic（脚本诊所）的专栏中查阅由三部分组成的系列文章（<http://msdn.microsoft.com/columns>）。

好在我们已回答了您针对从 WSH 脚本中运行其它程序而产生的某些疑难问题。我们还为您提供了有关 Run 和 Exec 方法的简要概述，并初步涉及了存在于两者之间的某些差异--其中，最明显的区别体现为，Exec 方法将允许您针对程序输出结果进行处理，而 Run 方法则不具备这种能力。另一个重要区别是：只有 WSH 5.6 及其后续版本方可提供 Exec 方法。



[/code]

## 99. QTP 中读取 Excel 数据

[关于测试](#)

以下是我用 [QTP](#) 进行 [Excel](#) 数据的读取，希望可以给找这方面资料的朋友一点帮助，但也有可能代码有错误，请务必指出！以下是代码：

```

Option explicit
Dim srcData, srcDoc, i, j
set srcData = CreateObject("Excel.Application")
srcData.Visible = true
set srcDoc = srcData.Workbooks.Open("D:\test\calc.xls")
 srcDoc.Worksheets("Sheet1").Activate
Dim myarray(7, 4), temp
Dim str
For i=2 to 7
 For j=1 to 4
 myarray(i, j)= srcDoc.Worksheets("Sheet1").Cells(i, j).value
 Select Case j
 Case "1"
 str=str&myarray(i, j)&" "
 Case "2"
 temp=myarray(i, j)
 Case "3"
 str=str&myarray(i, j)&" "&temp
 Case "4"
 str=str&" = "&myarray(i, j)
 End Select
Next

```

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

```

str=str&chr(13)
Next
srcData.Workbooks.Close
msgbox str

```

## 100. QTP 学习点滴

发布时间：2007-5-16 13:50 作者：风过无息 来源：51testing 软件测试论坛

字体：[小](#) [中](#) [大](#) | [上一篇](#) [下一篇](#) | [打印](#)

1. 在[测试](#)中我们使用[QTP](#) 调试脚本的时候一般就是 DEBUG 或者 MSGBOX 察看一些信息, 其实有时候也可以使用 print 来实现批量的察看信息但是不影响程序运行.

运行脚本:

```

a="100"
print a
~~~~~
```

2. 取 datatable 特定行的数据可以这样使用

运行脚本:

```
DataTable.GetSheet("Action1").GetParameter("test").ValueByRow(4)
~~~~~
```

3. Wait Seconds [, Milliseconds]可以精确到毫秒.

~~~~~

4. 在自定义的 function 里面数组作为返回值.

运行脚本:

```

circuit = "399937"
Function trimString(circuit)
Dim holdArray(5)
holdArray(0) = Left(circuit, 2)
holdArray(1) = Right(circuit, 2)
msgbox holdArray(0) ' showed 39
trimString = holdArray' I get an out of range error here
End Function
dim myArray
' here I want to assign the return array to another array
myArray = trimString(circuit)
' and then call one element from it
msgbox myArray(1)
```

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

~~~~~

5. 计算一个操作的时间.

运行脚本:

```
Browser("Browser").Page("Page").Image("getRates").Click
var_StartTime = Timer
Browser("Browser").Page("Page").Sync
Browser("Browser").Page("Page").Check CheckPoint("Check1")
var_EndTime = Timer
intRespTime = round ((var_EndTime - var_StartTime), 2)
msgbox (intRespTime)
~~~~~
```

6. 取得指定 sheet (datatable) 的行数和列数(也可以理解为参数个数)

```
paramcount = DataTable.GetSheet("Action1").GetParameterCount
msgbox "There are " & paramcount & "columns in the data sheet."
rowCount = DataTable.GetSheet("Action1").GetRowCount
msgbox "There are " & rowCount & "rows in the data sheet."
```

## 101. QTP 的 Test 参数以及顶级 Action 参数的使用

发布时间: 2007-5-17 13:07 作者: songfun 来源: 51testing 博客

字体: [小](#) [中](#) [大](#) | [上一篇](#) [下一篇](#) | [打印](#)

网上有些关于如何使用 Action 参数的文章，不过关于 [QTP](#) 的 Test 参数和 Top-Level Action 参数的使用例子几乎没有。

有些人甚至不知道这个参数做什么用的，尤其是 Test 的 output 不知道怎么取。其实它是外部对象传给它的（这个外部对象可以是 Quality Center，也可以是 vbs 这样的驱动程序）。

以下给大家讲解一个关于 QuickTest 的 Flight 的例子。

首先，在 QTP 里录制一段脚本，代码如下：

CODE:

```
SystemUtil.Run "C:\Program Files\Mercury\QuickTest
Professional\samples\flight\app\flight4a.exe", "", "C:\Program
Files\Mercury\QuickTest Professional\samples\flight\app\", "open"
Dialog("Login").WinEdit("Agent Name:").Set Parameter("InAction1")
wait 5
Dialog("Login").WinEdit("Agent Name:").Type micTab
```

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

```

Dialog("Login").WinEdit("Password:").SetSecure
"4649c633fffc8803c10097292953c6334fde3e923"
Dialog("Login").WinEdit("Password:").Type micReturn
Window("Flight Reservation").Close
Parameter("OutAction1") = True

```

然后，鼠标选中 Keyword View 中的 Action1，点右键---Action Property，在 Parameters 的 Tab 标签下，分别加入：输入参数 InAction1，类型 String；输出参数 OutAction1，类型 Boolean。如下图。再然后，在 QTP 的菜单 File--->>Settings 的 Parameters 的 Tab 标签下，分别加入：输入参数 InParam1，类型 String；输出参数 OutParam1，类型 Boolean。如下图。

接着，鼠标还是选中 Keyword View 中的 Action1，点右键，这次点“Action Call Properties”，在 Parameter Values 里进行参数化传递设置，把 InParam1 的值传递给 InAction1，把 OutAction1 的值传递给 OutParam1。如下图。

以上设置完毕后，点“保存”，保存到 C:\下，存为 Test1 好了。

最后，在你的硬盘上新建一个 vbs 文件，文件内容如下：

CODE:

```

Dim qtApp ,pDefColl,pDef ,rtParams, rtParam
Set qtApp = CreateObject("QuickTest.Application")
qtApp.Launch
qtApp.Visible = True
qtApp.Open "C:\Test1"
Set pDefColl = qtApp.Test.ParameterDefinitions
cnt = pDefColl.Count
Indx = 1
While Indx <= cnt
Set pDef = pDefColl.Item(Indx)
Indx = Indx + 1
Wend
Set rtParams = pDefColl.GetParameters()
Set rtParam = rtParams.Item("InParam1")
rtParam.Value = "songfun"
qtApp.Test.Run , True, rtParams
MsgBox rtParams.Item("OutParam1").Value

```

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

做完这步之后，保存这个 vbs 文件，双击执行这个 vbs 文件，你会发现它自动启动了 QTP，而且进行了自动[测试](#)，最后还取到了运行成功与否的布尔值。这就是关于 Test、Top-Level Action 参数使用的例子，它的参数的整个传递过程是：外部 vbs 文件 传参数给 QuickTest 的 Test 的输入参数 InParam1，然后 InParam1 传参数到 InAction1 去驱动了 Action1 的测试，然后通过这个 Action1 得出了 OutAction1 的值，然后通过 OutAction1 传给 OutParam1，最后再传回到 vbs 文件中。示例用 MsgBox 来打出重新传回到 vbs 文件中的字符串。

## 102. 用 QTP 调用外部 vbs 读写 XML 的例子（一）

发布时间：2007-5-21 14:23 作者：songfun 来源：51blog

字体：[小](#) [中](#) [大](#) | [上一篇](#) [下一篇](#) | [打印](#)

目前，企业中对 XML 的应用越来越广泛，作为[自动化测试](#)的测试工程师，也应该掌握 XML 的读写操作。

以下我使用 XML DOM 技术演示一个例子，用以读取 XML 指定节点的节点内容值。

读取函数原型 GetXml strXmlPath,nodeName

这个函数的第一个参数表示 xml 文件所在路径，第二个参数表示希望获取到的 xml 节点名，请结合下列例子看

首先，新建一个 vbs 文件（取个名字叫 readXml.vbs），输入代码：

**CODE:**

```
'=====
' Author : songfun
'
' Description : Read XML
'=====

Dim strXML

GetXml "c:\search.xml","TestResult" '这个函数的第一个参数表示 xml 文件所在路径，第二个参数表示希望获取到的 xml 节点名，请结合
下列例子看
MsgBox strXML
```

```
Function GetXml (ByVal strXmlFilePath,ByVal xmlNodeName)
Dim xmlDoc,xmlRoot

Set xmlDoc = CreateObject("Microsoft.XMLDOM") '创建 XML DOM 对象
xmlDoc.async = False '控制加载模式为同步模式（xml 树加载完毕后再执行后续代码）
```

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

```

xmlDoc.load strXmlFilePath '载入 xml 文件
If xmlDoc.parseError.errorCode <> 0 Then
MsgBox "XML 文件格式不对，原因是：" & Chr(13) & xmlDoc.parseError.reason
Exit Function
End If
Set xmlRoot = xmlDoc.documentElement
xmlRecursion xmlRoot,xmlNodeName '调用 xml 递归函数传入指定的根和节点名
GetXml = True 'xmlRecursion (xmlRoot)

End Function

Function xmlRecursion(byval xmlNode,byval strNodeName)
If xmlNode.nodeName = strNodeName And xmlNode.hasChildNodes Then
If xmlNode.childNodes.item(0).nodeName = "#text" Then
strXML = strXML & xmlNode.nodeName & ":" & xmlNode.childNodes.item(0).nodeValue & Chr(13)
End If
End If
If xmlNode.hasChildNodes Then
For Each childNodeItem In xmlNode.ChildNodes
If childNodeItem.hasChildNodes Then
xmlRecursion childNodeItem,strNodeName
End If
Next
End If
End Function

```

接着，自己构造一个 xml 文件（取个名字叫 search.xml），如：

### CODE:

```

<?xml version="1.0" encoding="GB2312"?>

<ROOT>
<TestCase>
<TestNumberOne>1</TestNumberOne>
<TestNumberTwo>2</TestNumberTwo>
<TestNumberThree>+</TestNumberThree>
<TestResult>3</TestResult>
</TestCase>
<TestCase>
<TestNumberOne>3</TestNumberOne>
<TestNumberTwo>2</TestNumberTwo>
<TestNumberThree>-</TestNumberThree>
<TestResult>1</TestResult>
</TestCase>
<TestCase>
<TestNumberOne>3</TestNumberOne>
<TestNumberTwo>7</TestNumberTwo>
<TestNumberThree>*</TestNumberThree>
<TestResult>21</TestResult>
</TestCase>
<TestCase>
<TestNumberOne>2</TestNumberOne>
<TestNumberTwo>5</TestNumberTwo>
<TestNumberThree>/</TestNumberThree>

```

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

```
<TestResult>0.4</TestResult>
</TestCase>
```

</ROOT>然后，在 **OTP** 的 expert view 中调用刚才写的 vbs 文件：  
executefile "c:\readXml.vbs"

点“Run”，就能看到这个效果了。

当然，如果你不用调用外部文件的方式，也可以，如下图。

|       |              |
|-------|--------------|
| 自动化测试 | Version: 1.3 |
| 文档    | 2006-5-17    |

QuickTest Professional - [Test\*]

File Edit View Insert Automation Resources Debug Tools Window Help

New Open Record Run Stop

Test\* Action1

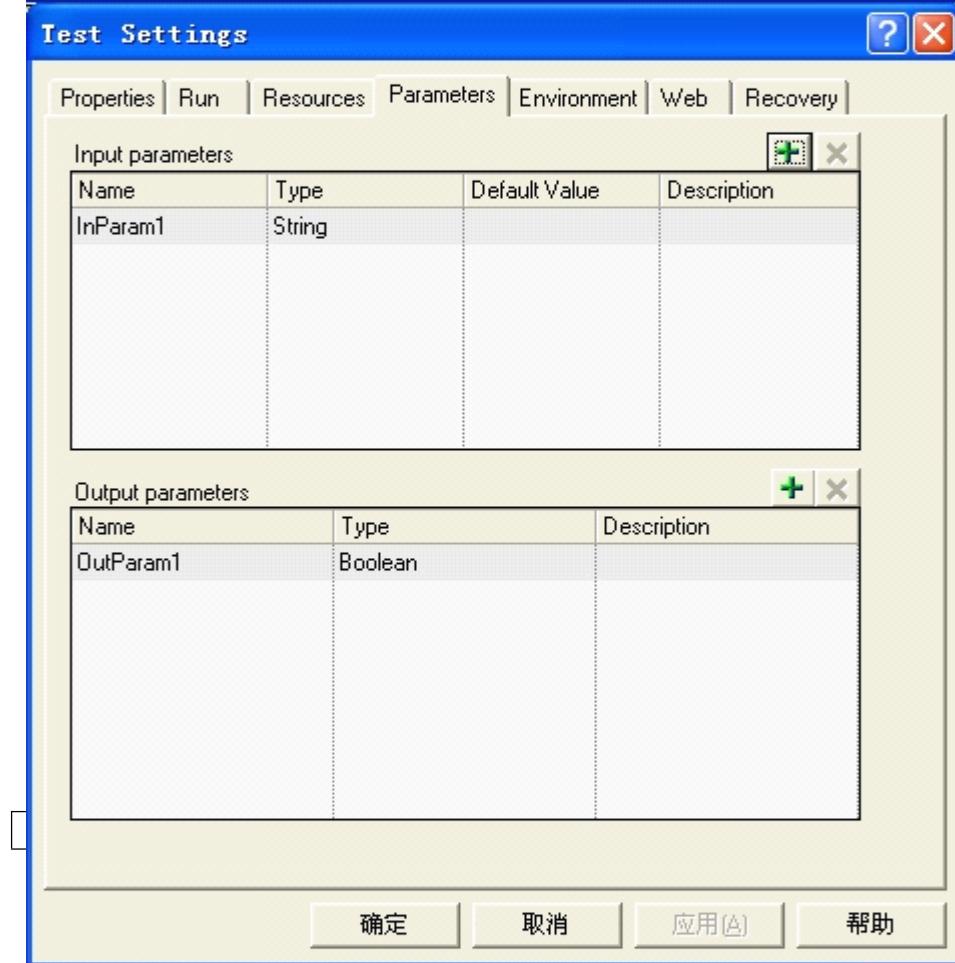
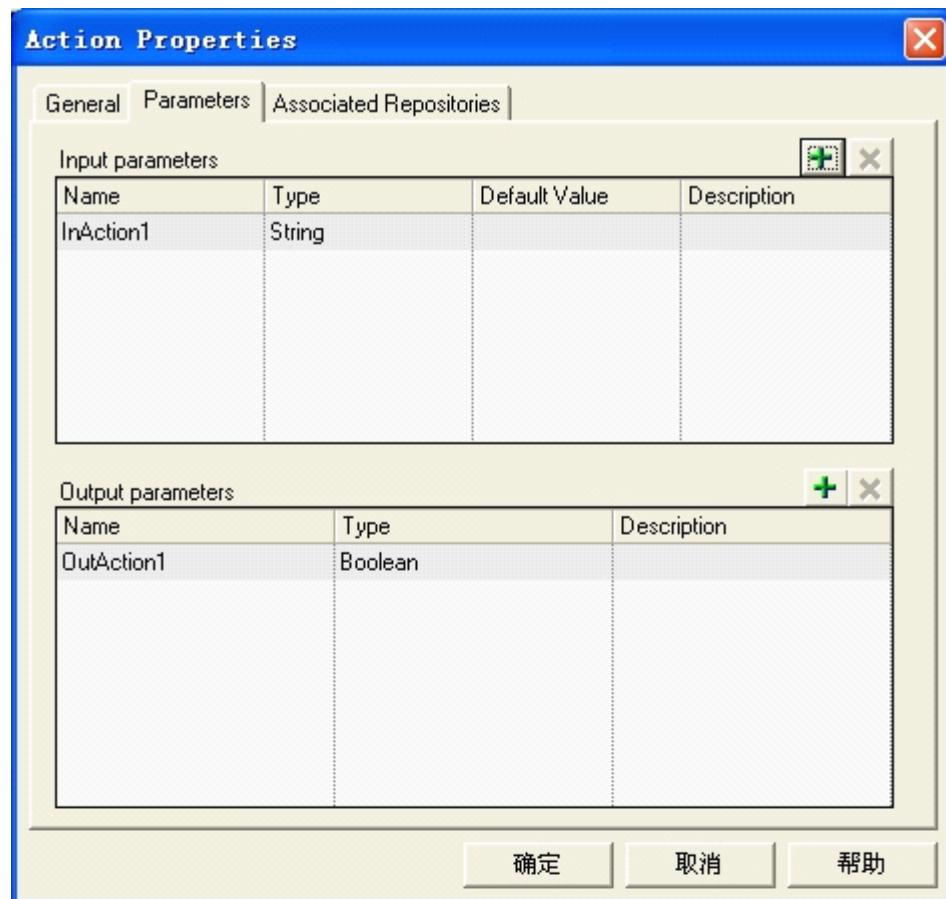
```

1: '=====
2: ' Author : songfun
3:
4: ' Description: Read XML
5: '=====
6:
7: Dim strXML
8:
9: GetXml "c:\calc1.xml","TestCase"
10: MsgBox strXML
11:
12:
13: Function GetXml (ByVal strXmlFilePath,ByVal xmlNodeName)
14:     Dim xmlDoc,xmlRoot
15:
16:     Set xmlDoc = CreateObject("Microsoft.XMLDOM") '创建XML DOM对象
17:     xmlDoc.async = False '控制加载模式为同步模式 (xml树加载完毕后再执行后续代码)
18:     xmlDoc.load strXmlFilePath '载入xml文件
19:     If xmlDoc.parseError.errorCode <> 0 Then
20:         MsgBox "XML文件格式不对, 原因是: " & Chr(13) & xmlDoc.parseError.reason
21:         Exit Function
22:     End If
23:     Set xmlRoot = xmlDoc.documentElement
24:     xmlRecursion xmlRoot,xmlNodeName '调用xml递归函数传入指定的根和节点名
25:     GetXml = True 'xmlRecursion (xmlRoot)
26:
27: End Function
28:
29: Function xmlRecursion(byval xmlNode,byval strnodeName)
30:     If xmlNode.nodeName = strnodeName And xmlNode.hasChildNodes Then
31:         If xmlNode.childNodes.item(0).nodeName = "#text" Then
32:             strXML = strXML & xmlNode.nodeName & ":" & xmlNode.childNodes.item(0).nodeValue & Chr(13)

```

Keyword View Expert View

|       |              |
|-------|--------------|
| 自动化测试 | Version: 1.3 |
| 文档    | 2006-5-17    |



|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

此文来自 51testing 博客, 转载请注明出处

原始链接:[http://www.51testing.com/?35/action\\_viewspace\\_itemid\\_10695.html](http://www.51testing.com/?35/action_viewspace_itemid_10695.html)

## 103. Ruby+Watir 经验谈：缘起

发布时间: 2007-5-29 10:56 作者: lifr 来源: 51testing 博客

字体: 小 中 大 | [上一篇](#) [下一篇](#) | [打印](#)

在半年前，我想很多人，包括我自己，都不知道 [watir](#) 是个什么东东。在国内几乎没有介绍的文章，尝鲜者聊聊。但半年后，随着用 Ruby+Watir 开发了一个具有较大规模的[自动化测试](#)框架，再回头看来，我[个人](#)认为，用 RAW 是做 [web](#) 自动化测试的第一选择。事实上，现在网上关于 Watir 的介绍和经验也多了起来。

在用 Watir 之前，对于自动化[测试工具](#)，我只有一些使用 [Rational Robot](#) 的经验。Robot 对于我来说最大的问题是，从一个 C/C++/java 程序员看来，它的 script 语言实在太不敢恭维了。包括语法‘丑陋’，提供的外部库功能匮乏(net, file, remote control, database ....)，还有编辑器也非常简陋。总的来说，我在写 robot script 的时候，我都在怀念在 eclipse 里使用 [java](#) 无所不能的感觉。

Robot 并不全是坏的印象。它最 cool 的地方就是 Record&Replay。但是，如果是对某个不断进化的产品进行测试，而且这个产品还不是那么简单。那么直接 RecordReplay 是远远不够的。套用一句不知道是哪个专家说的“自动化测试程序也是程序”，也需要设计，也需要架构，也需要设计模式。

[其他](#)商业的自动化测试工具我还简单体验过 Silktest，也是我所在公司曾经采用的。但它的 script 语言更不讨人喜欢，而且据使用它的同事说稳定性存在问题。更好的工具比如 [Mercury](#) 的 [QTP](#)，我没有使用过。

现在看来 Ruby+Watir 确实是一个正确的选择，但半年前这并不是显而易见的。公司原来的测试工具是 silktest，并在上面开发了一定量的 testcases。不过由于人力不够和 license 的问题，这个测试套件已经有很长一段时间停步不前了，并没有随着产品的更新而更新。silktest 形如鸡肋，Robot，QTP... 价格又高不可攀，而对于一个做产品的公司来说，自动化测试似乎是不可避免的，所以公司招聘了一个自动化测试 Consultant，希望她能带来价廉物美的利器。而这个 consultant 此前就在用 Ruby+Watir 做项目。

Consultant 带来了 Ruby+Watir，并计划做一个 SmokeTest suite，集成进 Nightly build。她希望能在 QA 组里找一个助手，因为我以前就做了一些自动互测试的工具并在 team 里使用，所以我也参加了进来。当然这个时候，我对 Watir 一无所知，而 Ruby，虽然曾经进入了我的视野，但在一年前，我选择了 Python 作为我的第一脚本语言。

2 个人的 Team 建立了。计划是 consultant 着手搭建框架，我先学习 Ruby 和 Watir。

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

随后的一个月，我不得不放下放下 ruby，而参加更紧迫的产品 releas 测试。当然我还是抽空翻看了 ProgramingRuby 2Ed 一书，写了一些程序来热身一下，虽然是蜻蜓点水，但也被 ruby 的优美和强大折服。当 consultant 再次来到中国，我对她准备如何实现这个框架还是一无所知。我也不知道我得职责到底是什么。

当我提出我得问题后，consultant 给我看了她已经完成的一些东西，我看了感觉比较失望。因为从她的代码看，她并不想提供一个框架，只是针对 smoketest 提供一些例程。对这些例程配合不同的输入数据，构成 smoketest 的全部。而我此时已经雄心勃勃，希望看到的是一个精心设计的面向对象框架，对 QA 所面对的产品的功能有一个比较完备的抽象层，一它不仅仅是为了 Smoketest

这个框架要实现的目标就是，在它基础上应该能够把大部分(>80%)的 Manual [test](#) 比较轻松的翻译为自动化脚步。对每一个 step 都能直接而且方便的翻译为一句 ruby 语句。也就是说需要达到一种效果：读自动化脚本就像在读用人类语言描述的 testcase。自动化脚本完全是在描述测试逻辑，这真是一个够“理想化”的目标！

我提出了我的想法，发给她一份我对整个框架设计的构思。她显然并不愿意大动干戈。我于是试图说服我们这边的 QA Manager 趁此机会做一个全功能的测试框架，并获得了成功。consultant 只能作出让步，但要求不能 miss smoketest 的目标。当然我同意了。

接下来的一个月 consultant 休假，她给我了一份 list，罗列了要达到的目标，然后就闪人了。我则全心投入了实现这个框架的[工作](#)。

经过几个月的开发，重要的是在给同事们使用后并得到了好多反馈。最后我觉得是初步达到了目标。随后，我会谢谢在这个过程中遇到的问题，解决的方法和得到的经验。

此文来源于 51testing 博客, 转载请注明出处

原始链接: [http://www.51testing.com/?764/action\\_viewspace\\_itemid\\_11524.html](http://www.51testing.com/?764/action_viewspace_itemid_11524.html)

## 104. Ruby+Watir 经验谈: Understanding Watir

发布时间: 2007-5-29 10:58 作者: lifr 来源: 51testing 博客

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

字体: 小 中 大 | [上一篇](#) [下一篇](#) | [打印](#)

使用 Watire，主要通过是使用 [Watir](#)::IE 的一个实例来获得页面上的各种元素，比如 button, link..., 然后你可以对这些元素进行操作，比如点击一个 button，点击一个 link...

```
ie = IE.new
ie.goto("http://somewhere")
ie.button(:caption, /save/).click
```

实际上，需要注意的是，Watir::IE 封装了一个当前页面的 DOM tree，而不是 html source。比如如果用 javascript 动态产生一个元素，在这里仍然可以访问。

这种访问 DOM tree 的能力并不是 Watir::IE 所独有的，div, frame 都具有这样的能力。实际上，这种能力是 Watir::SupportedSubElements 提供的。一般来说所有内部能包含任意子 tag 的元素都 include Watir::SupportedSubElements, like Watir::Frame, Watir::Div, Watir::TableCell, Watir::Form... 所以我建议把关于 Watir::SupportedSubElements 的 API 页面地址放在收藏夹，你会经常需要访问他们 :).

获得元素的方法参数一般来说有两种

1) how, value

大多数 element 都可以通过这种方式获得。

how 给出了方式，比如:id, :name

value 给出了值，比如

```
<input type="button" id="saveBtn" value="Save"/>
```

获得 button 的代码是: ie.button(:id, "myId")

2) how, what, value

这种方式适用于访问 radio 或者 checkbox。我们知道可能有多个 radio 具有同样的 name，只是 value 不同。

比如一个选择男或女的的 form field 有两个 radios,

```
<input type="radio" name="gender" value="male"/>Male
<input type="radio" name="gender" value="female"/>Female
```

获得第一个 radio 的代码是: ie.radio(:name, "gender", "male")

注意，如果给定条件的 element 不存在，方法返回的不是 nil，仍旧是这个类型的元

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

素的一个实例。要用 element.exists? 来判断这个元素是否真的存在。调用一个不存在的 element 的任何方法，都会触发一个 Exception。

下面是 Watir 的源代码结构。

```
module Watir
  # 所有的 Exception 都定义在另外一个文件里,
  # 且都位于 Watir::Exception 这个 Module 里面,
  # 这句话, 把所有的 Exception 都引入 Watir 这个 Module, 方便使用
  include Watir::Exception

  # 这个 module 为每一个 Dom Tree 上的元素定义了一个访问方法
  # 比如, 为 button 定义一个 button 方法, 当要得到一个 button 时就可以用
  # ie.button(:name, 'b1')这样的方式
  #
  # IE 和 Frame 都要 include 这个 module, 这样 IE 和 Frame 都具有这种能力
  # 理论上, 任何可以包括任意别的 tag 的 tag 都可以引入这个 module,
  # 比如<td>, <div>
  module SupportedSubElements
    def button(how, what=nil)
      end

      def link(how, what=nil)
      end
    end

  class IE
    # IE 拥有了访问 Dom tree 上 element 的方法
    include SupportsSubElements

    # 一个工厂方法, 你总是可以用这个方法来初始化一个 browser 对象
    def IE.start(url)
      end

    # IE 对象提供了很多操控 IE window 的方法比如
    # 前进, 后退, 刷新, goto, maximize, minimize
    def back
      end
  end
end
```

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

```

# 还提供了访问 HTML code 的方法
def html
    return document.body.outerHTML
end
def text
    return document.body.innerText.strip
end
end

class Form < IE
#Form 可以访问它内部的 Dom tree。另外, Form 还提供了一些 Form 特定的方法,
#比如 submit, 方法和获取 form 的 name, action, method 的方法
def name
end
end

# Element 是大多数页面元素的基类.
# 它定义了取得 适合大多数元素的成员变量
#   id, name, type, value, title, disabled(某个值不存在的话, 返回"");
#
# 和成员方法
#   click, exists?, enabled?, html, ...
class Element
end

# ElementCollections 是 代表某类页面元素的集合
# 比如 links, spans,
# ElementCollections 引入了 Enumerable module
# 所以你可以方便的操作里面的元素
class ElementCollections
  include Enumerable
end
end

```

Watir, 不得不说是一个非常精彩的 library, 它提供的功能非常强大, 还能保持 API 直观, 统一, 易于使用。再加上, 它的原生操纵语言是强大的 ruby, 用 Watir 写对于 [web](#) 的 [测试](#) 简直就是手拿一把锋利的多功能瑞士军刀。

Watir 的不足之处是它不能处理超出 Dom Tree 之外的东西, 比如 javascript popup box, 不过借助第三方库, 也能比较优美的解决。

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

## 105. Ruby+Watir 经验谈: javascript popup box

发布时间: 2007-5-29 11:00 作者: lifr 来源: 51testing 博客

字体: 小 中 大 | [上一篇](#) [下一篇](#) | [打印](#)

**Watir** 对 Dom Tree 有超强的能力, 但一旦超出了它的势力范围, 比如 javascript popup box, 它就无能为力了。

解决这个问题的思路是启动另外一个进程, 利用第三方工具(AutoIt), 来完成对这些对象的操作。

Watir 安装已经自带了 AutoIt, 但它的辅助方法还不太好。主要是缺少判断是否有 box 弹出的方法。

附件里提供有两个 script, 一个是作为另外一个进程运行的脚本 clickJsDialogButton.rb, 它的功能描述如下

- 1) 探测 pop box
- 2) 如果发现了 popup box, click "enter" key, or "escape" key.  
"enter" key 和 click OK button 效果一样  
"escape" key 和 click Cancel button 效果一样

```

1 # argument: {ENTER}, or{ESCAPE}
2 # return 0(true) if there is a box pops up, 1(false) if not.
3 require 'watir/WindowHelper'
4
5 AutoItRetTimeout = 0
6 AutoItRetSucceed = 1
7
8 DefaultTimeOut = 30
9
10 IETitle = "Microsoft Internet Explorer"
11 PRESSKEY = ARGV.shift
12
13 autoit = WIN32OLE.new('AutoItX3.Control')
14 foundAndClicked = false
15
16 ret = autoit.WinWait IETitle, "", DefaultTimeOut
17 if ret==AutoItRetSucceed # 0 timeout, 1 succeed
18   autoit.Send PRESSKEY
19   foundAndClicked=true
20 end
21
22
23 ShellExitTrue = 0
24 ShellExitFalse = 1
25
26 exit foundAndClicked ? ShellExitTrue : ShellExitFalse

```

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

还有一个是提供操纵 popup box 的 jsDlg.rb

```

module Everbright
  module Util
    # module JsDialog helps to handle javascript popup box.
    #
    # JavaScript popup box is out of the control of Watir::IE,
    # so we cannot detect/click an javascript box by a Watir::IE instance.
    #
    # Basically, methods of JsDlg accept an block by running which there
    # will pops up an javascript dialog box. Then you can click "OK" or "Cancel" button of it.
    #
    # Also you can detect whether or not there will pop up a dialog box by method JsDlg.popAlertbox?.
    # This method is especially useful to test input validation.
  module JsDlg

    # click button of a pop-up js dialog box
    #   button can be
    #     :OK
    #     :CANCEL
    #
    # return thread
    def clickJsDialogButton(button)
      path = File.join(File.dirname(__FILE__), "clickJsDialogButton.rb")

      key = button==:OK ? "{ENTER}" :
        button==:CANCEL ? "{ESCAPE}" : nil
      raise "button can only be :OK or :CANCEL" if key.nil?

      #puts "cmd = +"ruby "#{path}\\" "#{key}\\""
      t = Thread.new { system("ruby \"#{path}\" \"#{key}\"") }
      return t
    end

    def toClickAlertOk #:yield:
      clickJsDialogButton(:OK)
      yield
    end

    def toClickConfirmOk #:yield:
      clickJsDialogButton(:OK)
      yield
    end

    def toClickConfirmCancel #:yield:
      clickJsDialogButton(:CANCEL)
      yield
    end

    # Whether running the block leads an alert dialog box pops up?
    # if yes, click "OK" button of the alert dialog box and return true
    # if no, return false
  end
end

```

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

```

def popAlertBox?(&block)
path = File.join(File.dirname(__FILE__), "clickJsDialogButton.rb")

ret = true
t = Thread.new { ret = system("ruby '#{path}' '{ENTER}'") }
block.call
t.join
return ret
end

module_function :toClickAlertOk, :toClickConfirmOk, :toClickConfirmCancel
module_function :popAlertBox?, :clickJsDialogButton
end
end
end

```

使用 JsDlg，操纵 popup box 就非常简单了，比如删除操作往往都会弹出一个 confirm 窗口

```

JsDlg.toClickConfirmOK {
  ie.button(:caption, /delete/).click
}

```

在输入了一些非法数据后，[测试](#)提交 form 时有 alertbox 弹出

```

assert_true(JsDlg.popAlertBox?{ ie.button(:caption, /Submit/).click })

```

注意，这里并没有处理 prompt 窗口。因为在我的测试中一直也没有用到

## 106. 开源 Web 自动化测试框架——Watir 试用手记

发布时间：2007-4-12 09:16 作者：jackei 来源：cnblog

字体：[小](#) [中](#) [大](#) | [上一篇](#) [下一篇](#) | [打印](#)

[Watir](#) 是一个使用 Ruby 实现的开源 Web [自动化测试](#) 框架，相对于那些庞大的商业工具来说，它很小巧，也很灵活，提供的功能也足够用。最近抽时间试用了一下，感觉还不错，准备下一步在公司推广使用。

因为 Watir 的网站上用户手册、示例代码以及 FAQ 都维护的不错，所以已有的东西我就不重复了，在这里简单介绍一下，如果同行们有兴趣，可以一起研究一下。

### 1. 脚本示例

先丢一段脚本给大家看看使用 Watir 来书写脚本是多么的方便。下面的例子是 Watir 自带的一段测试 Google 的搜索功能的脚本，不过我只保留了最主要的部分，以便它看起来更简洁一些：

```

require 'watir' # the watir controller
# open the IE browser
ie = Watir::IE.new
# Step 1: go to the test site: http://www.google.com
ie.goto (http://www.google.com)

```

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

```

# Step 2: enter 'pickaxe' in the search text field
ie.text_field(:name, "q").set("pickaxe")      # q is the name of the search field
# Step 3: click the 'Google Search' button
ie.button(:name, "btnG").click                # "btnG" is the name of the Search button
# Actual Result: Check that the 'Programming Ruby' link appears on the results page
if ie.contains_text("Programming Ruby")
  puts "Test Passed. Found the test string: 'Programming Ruby'. Actual Results match Expected Results."
else
  puts "Test Failed! Could not find: 'Programming Ruby'"
end
# End of test: Google search

```

这段脚本要做的事情是打开 Google 的主页，然后在 Google 唯一的那个文本框内输入“pickaxe”这个字符串，然后按下“Google 搜索”按钮，之后验证搜索结果的页面中是否包含了“Programming Ruby”这个字符串，并根据结果使用 puts 函数在屏幕上打印不同的信息。脚本中“#”后面的绿色部分是注释的内容。简单吗？说实话要比那些商业工具录制的脚本还要简洁和简单。

## 2. 所需要的环境

Ruby：因为是使用 Ruby 实现的，脚本也是 Ruby 的脚本，所以需要在本机安装 Ruby。可以点击[这里](#)下载。根据文档中说的，最好选择 Ruby 1.8.2-14 或者更高的版本，我安装的是 Ruby 1.8.2-15 Stable Release。

Watir：可以点击从[这里](#)下载，我下载的是 Watir 1.4，是一个.zip 文件，解压缩以后执行 install.rb 就可以了，具体的安装和配置请参见 [Watir 用户手册](#)。不要怕，虽然是英文的，但是很简单。

WINDOWS 2000 或 XP + IE 5.5 以上版本：根据 Watir 网站上的描述，这是他们目前支持的环境。

## 3. 所需的知识背景

因为毕竟还是要写代码的，所以开发经验是必需的，任何语言的开发经验都可以——虽然 Ruby 是一个面向对象的脚本语言，不过你可以暂时不理它那么多（别被面向对象、脚本语言之类的词汇吓倒），如果你写过 VBScript 或者[其他](#)什么代码那么 Watir 就是很简单的。不过，需要了解 Web 开发，比如 HTML 的基本语法，因为在开发脚本时需要根据 Web 页面的源代码来确定对 Web 对象的识别方式——当然，也可以借助其他工具来实现，例如 Mozilla Firefox 中提供的“DOM 查看器”（可以在 Mozilla Firefox “工具”菜单下找到，具体的使用方法参见 Mozilla Firefox 帮助）。

另外，测试和自动化测试方面的基本知识还是要有的。

## 4. 脚本的生成

录制功能就别想了，Watir 没有提供这项功能。如果你非用录制不可，那就选择其他吧。不过就自己的使用来看，Watir 的脚本写起来比用 [QTP](#) 或者 Rational 的工具录制还要方便和快捷。

## 5. 对象的识别、操作和自动验证

Watir 提供了对多种常见 Web 对象的识别和支持，例如 Hyperlinks 的点击、Checkboxes 的选中和清除、Radio Buttons 的选中和清除、下拉框和列表框的选择、文本框的输入、各种按钮的点击以及

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

Frame 的访问、弹出窗口的控制等。当然，既然可以识别和操作这些对象，也可以访问这些对象，使测试结果的验证自动化。具体信息可以参见 [Watir 用户手册](#)。

## 6. 其他特性

如果你熟悉了 Ruby，再结合其他一些工具，可以在 Watir 框架的基础上扩展出很多特性。例如：外部文件或[数据库](#)的读取、模块化的开发、可重用的函数库、数据驱动、关键字驱动、脚本的版本化控制以及测试结果的管理，等等。

## 7. 相关链接

Watir 主页：<http://wtr.rubyforge.org/>

Ruby 中文手册：<http://www.ruby-cn.org/doc.html>

Ruby 主页：<http://ruby-lang.org/en/>

就如前面所说的，Watir 的用户手册、FAQ 等方面维护的很不错，如果完整的看完 Watir 主页上的 sample test 和 User's Guide 基本上就可以用 Watir 开始测试 Web 应用了。如果有兴趣进一步研究，可以参考一下 Technical Doc，也可以读一下 Watir 自带的几个示例。当然，也可以在这里留言或者 email 我，大家一起讨论。

## 107. Ruby+Watir 经验谈：设计 RUTF 的 TestRunner 以产生格式化的测试结果

发布时间：2007-5-30 11:39 作者：lifr 来源：51testing 博客

字体：[小](#) [中](#) [大](#) | [上一篇](#) [下一篇](#) | [打印](#)

对一个[测试](#)框架来说，测试结果报告是很重要的部分。但是 ruby unit [test](#) framework 并没有提供一个 testrunner 来输出格式化的结果，所以必须自己写一个。

### RUTF 的架构

在开始写自己的 testrunner 之前，我们需要理解 ruby unit test framework（简写为 RUTF）的结构。

RUTF 执行 TestCase 的过程简单明了，它里面有 4 个概念

原料：TestCase/TestSuite

执行配置者：TestRunner

执行体：TestRunMediator

产品：TestResult

处在中枢位置的是 TestRuner，它会创建和配置 执行体，把原料输入给执行体，并让执行体跑起来。

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

## TestRunner 和 TestRunMediator 的关系

需要说明一下 TestRunner 和 TestRunMediator 的关系。这是一个 监听器 模式的应用。  
被监听者是 TestRunMediator，可以为下列事件向一个 TestRunMediator 注册 listener。

事件的宿主      事件类型

|                 |                             |
|-----------------|-----------------------------|
| TestRunMediator | :STARTED, :FINISHED, :RESET |
| TestCase        | :STARTED, :FINISHED         |
| TestSuite       | :STARTED, :FINISHED         |
| TestResult      | :FAULT, :CHANGED            |

而 TestRunner 正是通过提供不同的 listener，而获得不同形式的结果报告。所以为了获得格式化的测试结果报告，我们需要提供自己的 TestRunner。

!!!TestCase 的含义，在定义时和运行时是不一样的

另外一个需要关注的地方是，对于 RUTF 来说，一个最小的执行单位（我叫它测试单元 TestUnit，--注：这个不是 RUTF 的概念）不是一个 TestCase 实例，而是一个 TestCase 实例的一个 test 方法。所以在定义时有下面的事实

一个 TestCase 是 TestUnit 的集合

一个 TestSuite 是 TestCase 的集合

但在运行时，在 RUTF 的看来，一个 test 方法就是一个' testcase'，而一个 TestCase，作为' testcase' 的集合，叫做' testsuite'。所以事件类型 TestCase::STARTED 表示的其实是一个 test 方法开始执行的事件。同理，TestSuite::STARTED 表示的是我们定义的一个 TestCase 开始执行的事件。

## 转换 xml 形式的结果到 Html

```
# <run time='0.031'>
#   <suite name='TC_MyTest'>
#     <test name='test_a(TC_MyTest)' passed='true'>
#       <result error_count='0' assertion_count='1' failure_count='0' />
#     </test>
#     <test name='test_b(TC_MyTest)' passed='false'>
#       <result error_count='0' assertion_count='1' failure_count='1'>
#         <fault msg='test_b(TC_MyTest): failure in test_b.'>Failure:
#           test_b(TC_MyTest) [testrunner.rb:162]:
#             failure in test_b.&lt;false&gt; is not true.
#         </fault>
#       </result>
#     </test>
```

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

```
#      <suite>
# <run>
```

xml 的结果适合作为中间数据保存，但其可视性和可理解性还不好，还需要把 xml 转换为 html。in ruby way，最方便的是用 erb 模版，在加上一个 css。我们就能得到漂亮的报表了。

## 108. QTP 应用程序编程

### [QTP](#)

#### 1. 介绍

使用 QTP 的 automation object model，可以对 QTP 进行自动化编程，从而自动对 QTP 进行选项设置、自动运行[测试](#)（或组件），而无需使用 QTP 界面手工进行这些操作。

如果要重复多次的完成同一项[工作](#)任务，或重复多次运行同一测试（或组件），或需要快速配置 QTP 选项，则 QTP 的自动化编程是非常有用的。

本部分内容包括：

自动化 QTP 操作

决定什么时候使用 QTP 自动化编程

选择设计或运行自动化程序的语言与开发环境

了解自动化程序的基本元素

创建自动化程序脚本

#### 1.1. 自动化 QTP 操作

使用 QTP 的 Automation Object Model 进行自动化编程，实现 QTP 操作的自动化。QTP 的 Automation Object Model 提供了 objects、methods、以及 properties，使我们可以在其它的应用程序中控制 QTP 的操作。

#### 1.2. 决定什么时候使用 QTP 自动化编程

就象我们使用 QTP 设计测试（或组件），创建一个有用的自动化程序也需要一个计划、设计、测试的过程。

下面是几个实用的 QTP 自动化编程的例子：

- Initialization programs（初始化编程）—你可以编写一个程序，来自动地启动 QTP，配置操作选项及录

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

制设定。

- Maintaining your tests or components (维护测试或组件) —你可以编写一个程序，作用于一系列已有的测试或组件上，以统一达到某一目的。例如：
  - Updating values—使用合适的插件环境打开每个测试或组件，并以 update 的模式运行它（当然必须对应相应的已升级的被测应用程序），然后保存测试或组件以保存更新后的值。
  - Applying new options to existing tests of components (对已有的测试或组件应用新的操作选项) —当 QTP 版本升级以后，你可能发现有某些新的选项设置可应用到当前已有的测试或组件中。你可以编写一个程序，打开每个已有的测试或组件，为它们设置新的选项设置，然后保存并关闭它。
- Calling QuickTest from other application (在其它应用程序中调用 QTP) — You can design your own applications with options or controls that run QuickTest automation programs. For example, you could create a Web form or simple Windows interface from which a product manager could schedule QuickTest runs, even if the manager is not familiar with QuickTest.

### 1.3. 选择设计或运行自动化程序的语言与开发环境

现有多种面向对象的编程语言，可用于实现 QTP 的自动化编程。对于每种语言，都有多种开发环境的支持，在该开发环境中，可用以设计或运行自动化程序。**编写自动化程序**

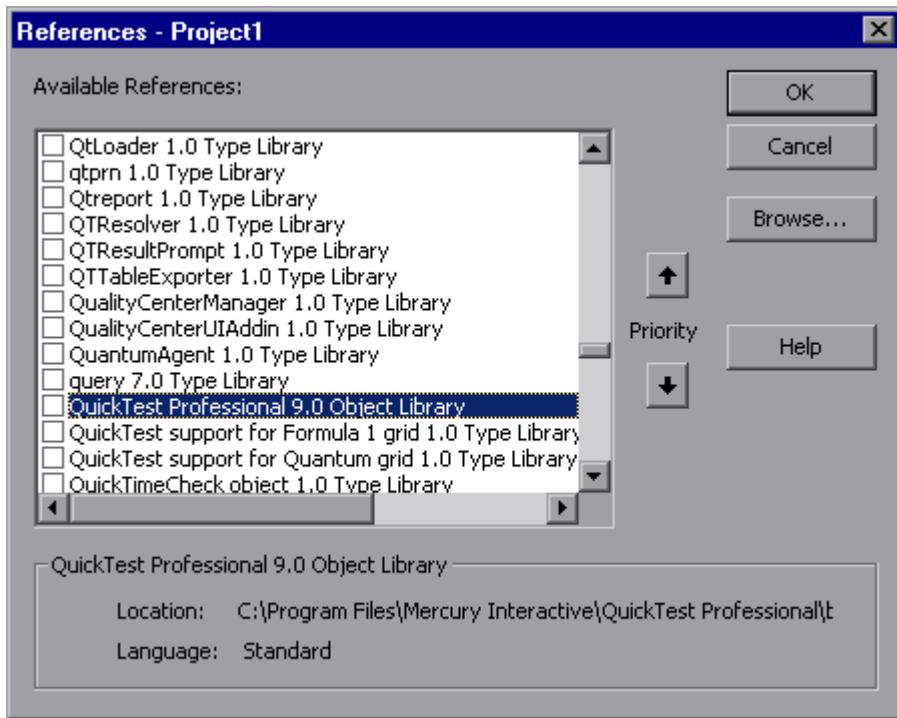
你可以在任何一个支持 automation 的语言与开发环境中编写 QTP 自动化程序，如 VBscript、JavaScript、Visual Basic、Visual C++ 或 Visual Studio.NET 等等。

有的开发环境 support referencing a type library。A type library 是一个二进制文件，包括了对象模型的描述、界面、以及其它定义。

如果你选择的环境 support referencing a type library，在你编程的过程中，你就可以享受到以下好处，如智能提示的功能、语句自动完成的功能、在状态栏显示帮助提示的功能。QTP 的 automation object model 提供了一个 type library 文件，文件名为 QTObjectModel.dll，这个文件存储于 QTP 安装目录的 bin 目录下。

如果你选择了一个 support referencing a type library 的开发环境，在编写或运行程序之前，确保装载了 QTP 的 type library。例如：如果你选用了 Microsoft Visual Basic，在 Microsoft Visual Basic 的菜单中选择 Project>References，打开 References 对话框，然后选择 QuickTest Professional<版本号>Object Library。

|       |              |
|-------|--------------|
| 自动化测试 | Version: 1.3 |
| 文档    | 2006-5-17    |



## 运行自动化程序

很多应用程序都可以运行自动化程序，你甚至可以在命令行使用 WSH 来运行自动化程序。

例如：Wscript.exe /E:VBscript myscript.vbs。

### 1.4. 了解自动化程序的基本元素（Element）

与大部分 automation object modes一样，QTP 的 automation object model 的根对象是 Application 对象。这里的 Application 对象代表 QTP 应用程序。使用它，可以返回其它 QTP 元素如 Test 对象（它代表一个测试或组件文件）、Options 对象（它代表 Options 对话框）、Addins 集合（它代表 Add-in 管理对话框中一系列的 add-in）等，并可以完成以下操作如装载 add-in、启动 QTP、打开并保存测试或组件、以及关闭 QTP。

由 Application 对象返回的每一个对象，又可以返回其它对象，以完成其它与该对象相关的操作，并获取（或设置）与该对象相关的属性值。

每个自动化程序从创建 QTP Application 对象开始，创建此对象并不是启动 QTP，它仅仅是提供一个对象，以便于你从该对象获得通道，以通向其它 QTP automation object model 的对象、方法以及属性。

如果你在程序中需多次的启动或退出 QTP，你可以只创建一个 Application 对象实例，而不必多次创建 QTP Application 对象。

#### 1.4.1. 创建 Application 对象

对于不同的语言，创建 Application 对象的过程稍有不同。下面例子中，是使用 VBscript 语言，实现创建 Application 对象，并以 visible 的模式打开 QTP 程序的操作。

```
Dim qtApp
```

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

|                                                 |                    |
|-------------------------------------------------|--------------------|
| Set qtApp=CreateObject("QuickTest.Application") | '创建 Application 对象 |
| qtApp.Launch                                    | '启动 QTP            |
| qtApp.Visible=True                              | '设置 QTP 为可见模式      |

## **4.2. 建设自动化程序的其它部分**

如何建设自动化程序的其它部分，与你的编程目标有关。

在启动 QTP 之前，你可以完成一些其它操作，如获取相关的 add-in、装载 add-in、并指导 QTP 以 visible 模式打开 QTP 等。

当完成这些预备步骤后，如果 QTP 还没有被打开，你可以使用 Application.Launch 方法打开它。自动化程序的其它操作一般都是位于 Launch 方法之后。

当完成这些必需的操作之后，或你想要实现的操作需要关闭并重新启动 QTP（如修改插件的装载），则使用 Application.Quit 方法。

在远程电脑中运行自动化程序

默认情况下，当你在自动化程序中创建一个 Application 对象时，该对象创建于本地电脑中（使用 QTP 的本地副本）。你也可以选择在远程 QTP 电脑中运行自动化脚本。为了实现这个目标，你必须：

- 确保远程电脑的 Distributed COM (DCOM) 配置属性已设置为允许你运行 QTP 自动化脚本。
- 在自动化程序的创建 Application 对象的脚本行，指定远程电脑的名称，例如在 VBscript 的 CreateObject 函数中使用可选参数 location。

名词：COM—组件对象模型。COM 提供了一套允许同一台计算机上的客户端和服务器之间进行通信的接口。

DCOM—分布式组件对象模型。是一系列微软的概念和程序接口，利用这个接口，客户端程序对象能够请求来自网络中另一台计算机上的服务器程序对象。

**设置远程电脑中的 DCOM 配置属性**

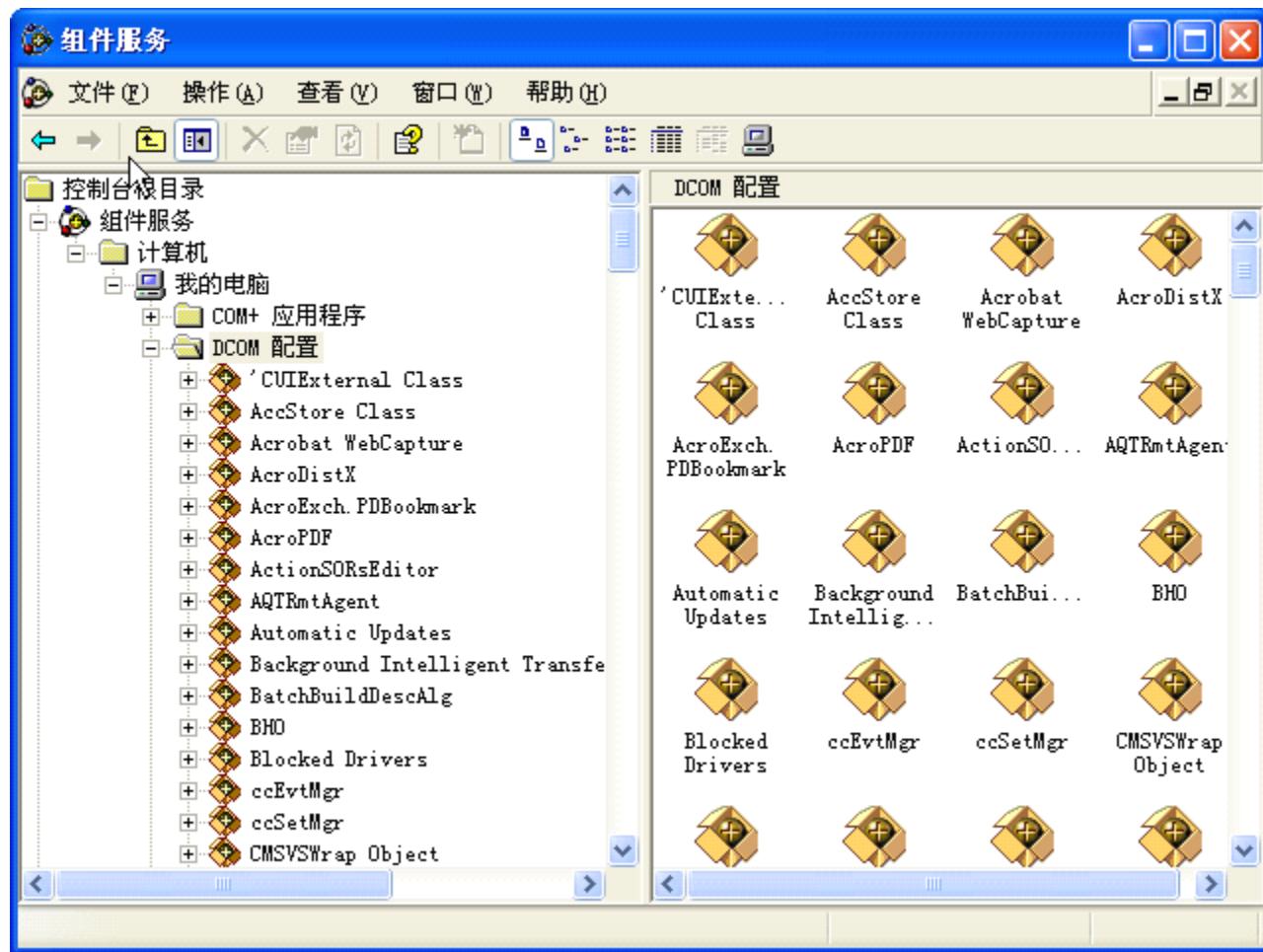
要在远程电脑中运行自动化程序，必须确保在远程电脑的 DCOM 设置中，给了你装载及设置 QTP COM 服务器的权限。

下面描述了在远程电脑中设置 DCOM 的过程。注意因为远程电脑的[操作系统](#)的不同，DCOM 配置对话框中显示可能不同。

（下面的图片都是 WinXP 操作系统中的）：

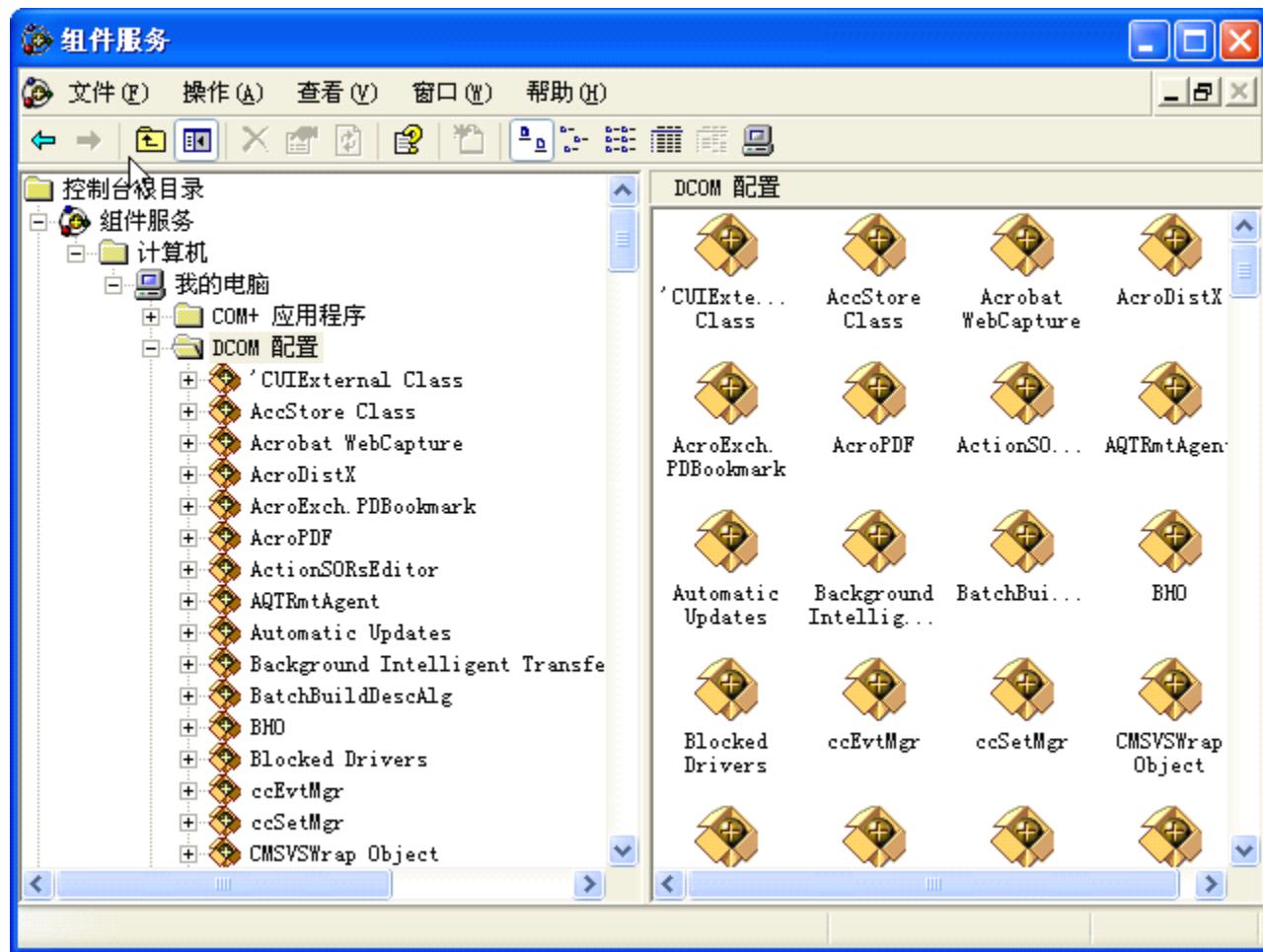
1. 在远程电脑中，选择 Start>Run，打开 Run 对话框。
2. 在命令栏输入“dcomcnfg”后，点击 OK，打开 DCOM 配置对话框或组件服务窗口（这与操作系统有关），并显示当前电脑中有效的 COM 应用程序。

|       |              |
|-------|--------------|
| 自动化测试 | Version: 1.3 |
| 文档    | 2006-5-17    |



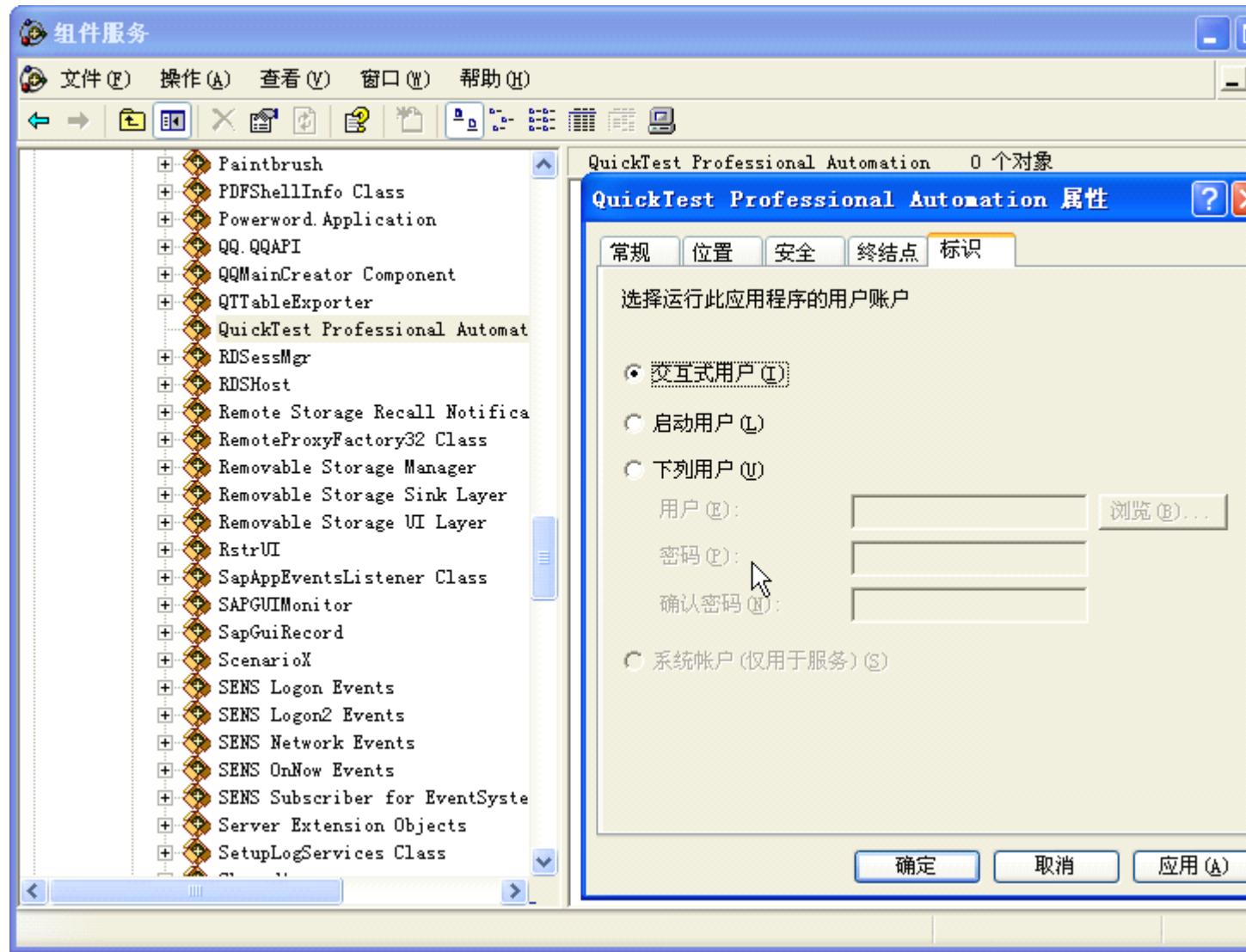
3. 在列表中选择“QuickTest Professional Automation”，打开属性对话框。

|       |              |
|-------|--------------|
| 自动化测试 | Version: 1.3 |
| 文档    | 2006-5-17    |



4. 在 QuickTest Professional Automation 对话框中，选中 Security（安全）页签。
5. 在 “launch permissions”（启动和激活权限）部分，选择 custom（“自定义”）选项，然后点击 EDIT（“编辑”）按钮。
6. 使用 Add 及 Remove 操作选择网络用户或组，让用户或组就可以在本地电脑中通过自动化程序，远程让 QTP 脚本在本电脑上运行。当完成设置用户或组后，点击 OK 保存设置。
7. 重复 5、6 步，设置 “configuration permissions” 部分，设置用户或组，让用户或组可以通过自动化程序，远程修改本电脑的 QTP 的选项配置。
8. 在 QuickTest Professional Automation 对话框中，点击 Identity（标识）页签，选择 interactive user 选项。

|       |              |
|-------|--------------|
| 自动化测试 | Version: 1.3 |
| 文档    | 2006-5-17    |



9. 点击 OK 保存 QuickTest Professional Automation 属性设置。

10. 点击 OK 关闭 DCOM 配置对话框，或关闭组件服务窗口。

#### 在远程电脑中创建 Application

当远程电脑的 DCOM 设置完成后，你就可以在自动化程序中指定该远程电脑。

在 VBscript 中，你可以在 CreateObject 函数的可选参数 location 中指定远程电脑名称。例如：在一个名为 MyServer 的远程电脑上运行自动程序，你可以编写以下脚本：

```
Dim qtApp
Set qtApp=CreateObject("QuickTest.Application","MyServer")
```

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

## 1.5. 创建自动化程序脚本

Test Setting 对话框中的 Properties 页签、Options 对话框的 General 页签、以及 Object Identification 对话框中，都有一个 Generate script 按钮。点击此按钮可以生成一个自动化脚本文件 (.vbs)，该文件包含了相关对话框的当前设定。

你可以运行这些脚本，运行该脚本可以打开 QTP，并且 QTP 的 Options 设置使用的是脚本中的设置；你也可以将脚本中的内容复制到你自己的自动化脚本中。

例如，你在 Options 对话框中通过 Generate script 功能创建了一个脚本，脚本的内容应该是下面这样的：

```
Dim App 'As Application
Set App = CreateObject("QuickTest.Application")
App.Launch
App.Visible = True
App.Options.DisableVORecognition = False
App.Options.AutoGenerateWith = False
App.Options.WithGenerationLevel = 2
App.Options.TimeToActivateWinAfterPoint = 500
...
...
App.Options.WindowsApps.NonUniqueListItemRecordMode = "ByName"
App.Options.WindowsApps.RecordOwnerDrawnButtonAs = "PushButtons"
App.Folders.RemoveAll
```

## 109. Sub procedure 与 Function procedure 的区别

VBscript 的 Sub Procedure 以 Sub 开始，以 End Sub 结束，中间是 VBscript 语句。Sub procedure 完成一系列的动作，但是不产生返回值。Sub Procedure 也可以携带参数（如常量、变量、或表达式）。如果 Sub Procedure 不携带参数，则 Sub 语句也必须带有一对空括号。

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

下面的这个 **Sub** Procedure 例子使用了两个内建 VBscript 函数 **MsgBox** 以及 **InputBox**。 **InputBox** 用来弹出提示框，提示用户输入相关信息； **MsgBox** 用以显示计算结果。计算功能由用户自建的 **Function** Procedure 来实现，相关信息在接下来的内容中另行讨论。

```
Sub ConvertTemp()
    temp = InputBox("Please enter the temperature in degrees F.", 1)
    MsgBox "The temperature is " & Celsius(temp) & " degrees C."
End Sub
```

## 2. Function Procedures

**Function** Procedure 以 **Function** 开始，以 **End Function** 结束，中间是 VBscript 语句。**Function** Procedure 与 **Sub** Procedure 相似，但是 **Function** Procedure 可以返回值。**Function** Procedure 也可以携带参数（常量、变量、表达式）。如果 **Function** Procedure 不携带参数，Function 语句也必须带有一对空括号。**Function** Procedure 在语句中，通过赋值语句将值赋给 Funciton 的名称，由此 **Function** Procedure 得到返回值。返回值的数据类型总是 **Variant** 类型。

在下面的例子中，用户定义的 Function **Celsius** 将华氏温度换算为摄氏度。当在 **Sub** procedure **ConvertTemp** 中用 **Celsius(temp)** 调用这个 Function 时，将参数 “Temp” 传到 Function 中（注意：此时参数 Temp 已经有明确的数据了）。然后在 Function 中，进行换算**工作**，并将换算结果赋给 Function 名称 “Celsius”，由此换算结果就成了 Function 的返回值，返回给调用它的 Sub procedure，并通过 Message 语句显示出来。

```
Sub ConvertTemp()
    temp = InputBox("Please enter the temperature in degrees F.", 1)
    MsgBox "The temperature is " & Celsius(temp) & " degrees C."
End Sub
Function Celsius(fDegrees)
    Celsius = (fDegrees - 32) * 5 / 9
End Function
```

## 3. 将数据传入或传出 Procedure

Procedure 只能通过参数获得数据。参数名可以是任何有效的变量名称。用户可以用 Sub 语句或 Function 语句创建 Procedure，无论是用哪种方式创建，Procedure 名称后面都必须带有一对括号。所有的参数包括在这一对括号之中，用逗号分隔。

在下面的例子中，“**fDegrees**”是一个参数，它将值传入 Function “**Celsius**”。

```
Function Celsius(fDegrees)
    Celsius = (fDegrees - 32) * 5 / 9
End Function
```

要想从 procedure 获取返回值，必需使用 Funciton procedure。要记住：只有 Function procedure 有返回值，Sub procedure 没有返回值。

## 4. 如何在脚本中使用 Sub Procedure 或 Function Procedure

**Function** 总是放置于变量赋值语句的右侧，或放置于表达式中。如

**Temp=Celsius(fDegrees)**

或

**MsgBox "The Celsius temperature is " & Celsius(fDegrees) & " degrees."**

在调用 **Sub procedure** 时，在 procedure 名字后带上所有必须的参数，参数之间用逗号隔开。Call 语句不是必须的，但是如果使用 Call 语句来调用它，procedure 名字后面的参数必须用括号括起来。

下面的例子表达了两种不同的调用方法：一个使用 Call 语句来调用，另一个没有使用 Call 语句。两种方法的是殊道同归。

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

Call MyProc(firstarg, secondarg)

MyProc firstarg, secondarg

注意：在使用 Call 语句调用 Sub Procedure 时，参数必须用括号括起来。

## 110. WebTable 对象

[QTP](#)

WebTable 如下图所示。

表格的标题的行 ID 是 “1” ，依次下推，表格的列 ID 也从 “1” 开始，第 1 列的 ID 是 1。

获得单元格的数据的语法如下：

- object.GetCellData (vtRow, vtColumn)

| 参数       | 描述                                           |
|----------|----------------------------------------------|
| object   | 一个 <a href="#">WebTable</a> 对象。              |
| vtRow    | 必须。一个变量。指所定位的单元格的行号。WebTable 表的第 1 行的行号是 1。  |
| vtColumn | 必须。一个变量。指所定位的单元格的列号。WebTable 表的第 1 列的列号是 1。. |

- 返回值类型

字符串，返回指定单元格中的数据。如果单元格中的数据无效，则返回 micCellDoesNotExist 。

| 用户ID : |          | 真实姓名 :   | 所在机构 :  | 南山区政府 | ▼    | 查询                                   |
|--------|----------|----------|---------|-------|------|--------------------------------------|
| 序号     | 用户ID     | 真实姓名     | 所在机构    | 电子邮件  | 联系电话 | 操作                                   |
| 1      | admin    | 系统管理员    | 南山区政府   |       |      | <a href="#">查看</a> <a href="#">编</a> |
| 2      | sun      | sun      | 粤海街道办事处 |       |      | <a href="#">查看</a> <a href="#">编</a> |
| 3      | ydy      | ydy      | 南山区政府   |       |      | <a href="#">查看</a> <a href="#">编</a> |
| 4      | testa    | testa    | 南山区政府   |       |      | <a href="#">查看</a> <a href="#">编</a> |
| 5      | qqq      | qqq      | 南山区政府   |       |      | <a href="#">查看</a> <a href="#">编</a> |
| 6      | wqy      | wqy      | 南山街道办事处 |       |      | <a href="#">查看</a> <a href="#">编</a> |
| 7      | 3        | 3        | 南山区政府   |       |      | <a href="#">查看</a> <a href="#">编</a> |
| 8      | 5        | 5        | 南山区政府   |       |      | <a href="#">查看</a> <a href="#">编</a> |
| 9      | 55       | 55       | 南山区政府   |       |      | <a href="#">查看</a> <a href="#">编</a> |
| 10     | abcdefg  | abcdefg  | 南山区政府   |       |      | <a href="#">查看</a> <a href="#">编</a> |
| 11     | abc      | abc      | 南山区政府   |       |      | <a href="#">查看</a> <a href="#">编</a> |
| 12     | abc      | abc      | 南山区政府   |       |      | <a href="#">查看</a> <a href="#">编</a> |
| 13     | 我        | 我        | 南山区政府   |       |      | <a href="#">查看</a> <a href="#">编</a> |
| 14     | aaadfdfa | aaadfdfa | 南山区政府   |       |      | <a href="#">查看</a> <a href="#">编</a> |
| 15     | test11   | test11   | 南山区政府   |       |      | <a href="#">查看</a> <a href="#">编</a> |

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

针对上图示例，获取第 7 行第 4 列的数据，并将返回值返回给变量 HELLO。示例如下：  
`HELLO= Browser("myBrowser").Page("myPage").Frame("mainFrame").WebTable("查询列表").GetCellData(7, 4)`

## 2. 如何定位表中链接的位置，并点击链接

同样，针对上表，点击第 8 行的查看链接，以下是示例。

```
set myEditLink=Browser("myBrowser").Page("myPage").Frame("mainFrame").WebTable("查询列表").ChildItem(8, 7, "WebElement", 0)
```

myEditLink.click

本操作使用了以下脚本：

- object.ChildItem (Row, Column, MicClass, Index)

| 参数       | 描述                                                                                                                                 |
|----------|------------------------------------------------------------------------------------------------------------------------------------|
| object   | 一个 <a href="#">WebTable</a> 对象。                                                                                                    |
| Row      | 必须。一个 Long 值。所定位的单元格的行号。WebTable 的第 1 行行号是 1。                                                                                      |
| Column   | 必须。一个 Long 值。所定位的单元格的列号。WebTable 的第 1 列列号是 1。                                                                                      |
| MicClass | 必须。一个 String 值。本语句返回值是单元格中的对象，这里 MicClass 指的是返回对象的对象类型。                                                                            |
| Index    | 必须。一个 Long 值。指单元格中类型为 MicClass 的对象的顺序号。当单元格中存在多个 MicClass 类型的对象时，通过此顺序号可以确定目标对象位置。第一个对象的顺序号是 0。如果单元格中只存在一个 MicClass 类型的对象，则取顺序号 0。 |

- 返回值

一个 Object 对象。

- 对上面脚本的解释：

`ChildItem(8, 7, "WebElement", 0)` 表示“查看”链接位于第 7 列，本语句实现的是点击第 8 行第 7 列的查看按钮，即用户 ID=3 的那条记录。

如果要点击编辑按钮，则用 `ChildItem(8, 8, "WebElement", 0)`。

`ChildItem(8, 7, "WebElement", 0)` 中的“WebElement”表示希望对单元格中对象类型为“WebElement”的对象。0 表示希望的对象是单元格中类型为 WebElement 的第 1 个对象。

## 111. 如何生成或修改 SQL

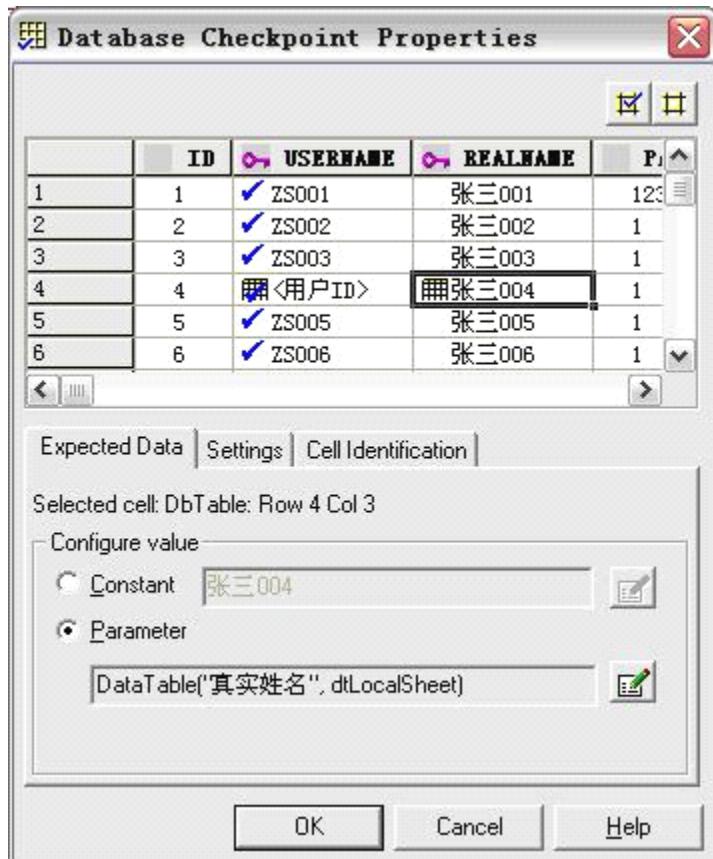
通过 Insert->checkpoint-databasecheckpoint 生成 [数据库](#) 检查点，在生成过程中，要求设置查询语句，并会根据查询语句得到一个查询结果表，该表中的数据即为预期数据。将来在运行脚本的过程中，QTP 又根据查询语句从当前的数据库中获取实际数据，然后将实际数据与预期数据相比较，从而得知检查点是否成功。

检查点语句生成以后，会产生一个 DbTable 对象，该对象在关键字视图可以看到，在仓库中也存在。可以通过设置该 DbTable 对象的 object properties 属性“Source”来修改 SQL 语句。修改了 SQL 语句就表示修改了查询结果。

修改了 SQL 语句后，不能相应的修改预期数据表。

如何理解检查点的属性（checkpoint properties）

|       |              |
|-------|--------------|
| 自动化测试 | Version: 1.3 |
| 文档    | 2006-5-17    |



上面的表是创建检查点时截取的。（无论什么时候打开本属性窗口，表中的数据都不会改变，但可以通过窗口中的“Configure value->Constant”来手工修改表中的数据，通过这种方式修改了数据后，就相当于对截图进行了修改。如果通过“Configure value->Parameter”来将单元格的数据参数化，虽然看起来是数据被修改，但是如果再选择 Constant，数据就会恢复，因此通过参数化数据的方法，不会真正影响截图的数据。）我们在表中指定想要检查的区域。（区域中的显示的值就是预期值。预期值可以是常数，也可以是参数变量。当在 Cell Identification 页中设置列 by position 时，本表的列必须与实际查询表的列相对应，否则检查不能通过。如在生成检查点语句时查询语句为“select username, id, realname from sys\_user t”，后来通过修改 object properties，将查询语句改为“select username, realname, id from sys\_user t”，则执行检查时是不能通过的，因为查询表中第 2、3 列的数据已经不能与截取的表相对应。同理，如果在 Cell Identification 页中设置行 by rownumber 时，也会有同样的问题存在。）

以下是属性框三个页签中的内容：

- [Expected Data](#)—设置预期值，可以是常量或参数。如，可以要求 QTP 从 DAta table 中取值作为预期值。
- [Settings](#)—设置预期值与实际值相比较时的语法规则。
- [Cell Identification](#)—指导 QTP 放置被检查的数据。如，假设你想检查位于检查点属性框中第一行第 2 列的数据，但是你明白，每运行一次[测试](#)脚本，查询出来的数据

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

的行的排序可能发生改变。因此，让 QTP 通过列名和行 *row containing a known value in a key column* 来定位单元格，而不是通过列数或行数来定位单元格。

Tip: The value matching settings and cell identification criteria apply to all selected cells in the checkpoint. If you want to use different value matching or cell identification criteria for different cells in the table or database, create separate checkpoints and specify the relevant cells for each.

#### 检查点属性窗：

Checkpoint timeout—指定检查点运行的最长时限，QTP 在检查点通过（在本时间范围内）或超时后，才进行下一步操作。如果在最长时限还没有检查通过，则本检查失败。例如，检查点装载数据需要一定的时间，增长 checkpoint timeout 时间，可保证检查点有足够的时间通过检查，不会超时失败。

注：Checkpoint timeout 选项仅仅对 table 检查点有效，对于 database 检查点无效。

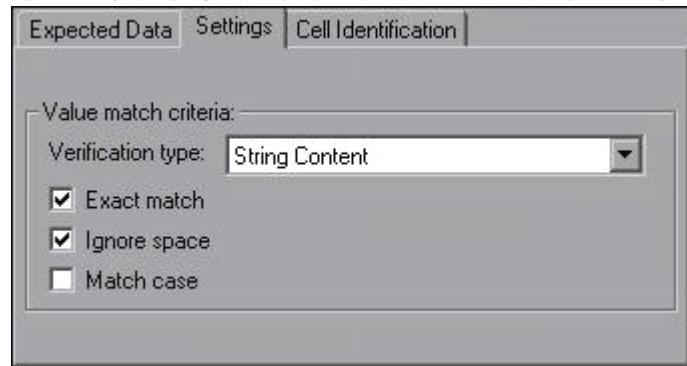
Insert statement—在新增检查点时，指明检查点语句在脚本中的插入位置。

注意：Insert statement 选项在录制或修改检查点时无效，只在新增检查点时有效。

注意：当一列是 *key column* 时，该列不一定是被检查的列，*key column* 的数据仅用来帮助确定被检查的数据行。

#### 指定数据的比较规则

本设置决定真实数据单元如何与预期数据相比较，本设置对所有单元格有效。



默认情况下，认为真实数据单元是字符串型，并进行精确检查，但是忽略空格。

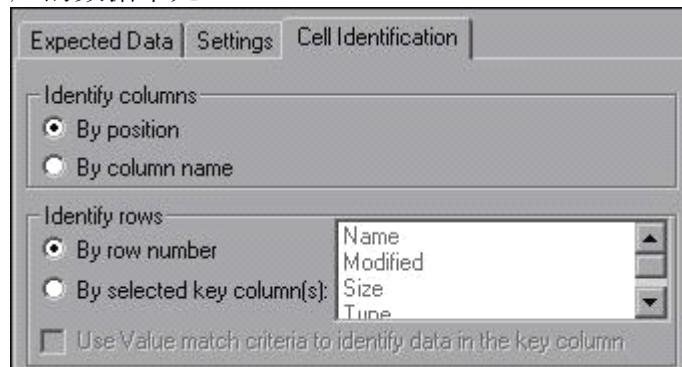
| 选项                | 描述                                                                                                                                                                                                                                                                     |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Verification type | <ul style="list-style-type: none"> <li>● 指定单元格中的内容是如何被比较的。</li> <li>● String Content— 默认。将被检查数据以字符串对待。如 2 与 2.00 会被认为是不同的值。</li> <li>● Numeric Content— 将被检查数据以数值对待。如 2 与 2.00 会被认为是同一值。</li> <li>● Numeric Range— 只要数据满足一定的范围，就会检查通过，真实数据只要在最小值与最大值之间就可以了。</li> </ul> |
| Exact match       | 默认。精确匹配。选中此项，则进行精确匹配；不选中，则只要预期值包含在真实值之中即可。                                                                                                                                                                                                                             |

|       |              |
|-------|--------------|
| 自动化测试 | Version: 1.3 |
| 文档    | 2006-5-17    |

|              |                                                                                           |
|--------------|-------------------------------------------------------------------------------------------|
|              | 注意：只有当 Verification type 为 String Content 时，才显示这个选项。                                      |
| Ignore space | 默认。在比较时忽略单元格中的空格。添加或减少空格不会影响检查结果。<br>注意：只有当 Verification type 为 String Content 时，才显示这个选项。 |
| Match case   | 区分字母的大小写。<br>注意：只有当 Verification type 为 String Content 时，才显示这个选项。                         |
| Min / Max    | 指定数据的误差范围。只要数据在误差范围之内都可视为正确数据。<br>注意：只有当 Verification type 为 Numeric Range 时，才显示这个选项。     |

### 指定 Cell Identification 设置

Cell Identification 页签中的设定，用以确定 QTP 如何在实际数据表中查找与预期值对应的数据单元。



Cell Identification 页签包括以下选项：

|                  |                                                                                                                                                                                                                                                                                              |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Identify columns | <p>指定将与预期数据相比较的实际数据单元在实际数据表中的列位置。</p> <ul style="list-style-type: none"> <li>● By position—根据列的顺序位置来定位（即预期数据表中的第 N 列对应实际数据表中的第 N 列）。如果列位置产生位移，就会导致不匹配。本选项一般用于 Table 检查点。</li> <li>● By column name—根据列名来定位（即预期数据表中的列名对应实际数据表中的列名）。列的位移不对检查点产生影响。一般用于 database 检查点，对于 Table 检查点无效。</li> </ul> |
| Identify rows    | <p>指定将与预期数据相比较的实际数据单元在实际数据表中的行位置。</p> <ul style="list-style-type: none"> <li>● By row number—默认。根据行的顺序位置来定位（即预期数据表中的第 N 行对应实际数据表中的第 N 行）。如果行</li> </ul>                                                                                                                                      |

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

|                                                             |                                                                                                                                                                                                                                                          |
|-------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                             | <p>发生位移，就会导致不匹配。</p> <ul style="list-style-type: none"> <li>● By selected key column(s) — 选中此项以后，可以设置关键列，关键列列名旁有一个“钥”标记。在定位行时，到实际数据表的关键列中查找与预期数据表关键列的值相匹配的值，所找到的第 1 个匹配值则是正确的行。行的顺序位移不影响检查结果。如果数据库中有多个行同时满足条件，则 QTP 只检查第 1 条记录。关键值可以是多个列的联合值。</li> </ul> |
| Use value match criteria to identify data in the key column | <p>在对关键列进行匹配时，使用 Setting 页签的 verification type 选项中设置的匹配规则。<br/>注：只有当选中了 By selected key column(s) 选项时，本选项才有效。</p>                                                                                                                                        |

当选择 by row number 时，属性窗中预期值所在的行数，应该与真实数据所在的行数相同，否则匹配失败。因此行移位都会导致检查失败。

当选择 by selected key column(s) 时，属性窗口中预期值可以在任意行位置。执行检查时，以该行 Key column 列的值为条件在数据库中进行筛选，如果选出了符合条件的记录，则将该记录与预期值进行比较，如果记录数据全部匹配，则检查通过。（当有多个 Key Column 列时，则筛选条件为这些列的数据的联合）。

要注意几个问题：

- (1). 只有被打勾的数据才被检查，其它没打勾的即使不匹配也没关系。
- (2). Key column 列的数据如果没有打勾，也不会被检查，它只是提供查询条件。
- (3). 属性窗口数据表中的数据本身就是默认预期数据，当然用户可以对这些预期数据进行修改。
- (4). 在属性窗口数据表中的数据是相对固定的，除非用户在本属性窗口中特殊别进行了修改。
- (5). 本属性窗口数据表中的所看到的数据都是预期数据，真实数据表是看不到的。真实数据表在每次执行脚本时都有可能不同。
- (6). 在运行结果(result)中，如果检查失败，则可点击“checkpoint ‘表名’”看到检查结果表。双击表中的单元格，弹出窗口显示该单元格的预期值与真实值。当然，本结果表中显示的仍然是预期值。
- (7). 如果在属性窗口的数据表中设置了一个检查点数据，但是数据库中有多条满足条件的数据，则只检查第一条满足条件的数据，如果第 1 条检查完全匹配，则通过，如果不匹配，则失败。
- (8). 如果在属性窗口的数据表中设置了多个相同的检查点数据，但数据库中只有一条满足条件的数据，该条数据只与第 1 个检查点进行匹配检查。其它几个检查点失败。

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

## 112. VBScript 中 SendKeys 的妙用

[QTP](#)

[quote]

### 什么是 VBscript 呢?

VBscript 的全称是:Microsoft Visual Basic script Editon. (微软公司可视化 BASIC 脚本版). VBS (VBscript 的进一步简写) 是基于 Visual Basic 的脚本语言, 其不编译成二进制文件, 而是直接由宿主 (host) 解释源代码并执行, 简单点说就是你写的程序不需要编译成 .exe, 而是直接给用户发送 .vbs 的源程序, 用户就能执行了.

### SendKeys 方法

模拟键盘操作, 将一个或多个按键指令发送到指定 Windows 窗口来控制应用程序运行,  
其使用格式为: object.SendKeys(string)

“object” : 表示 WshShell 对象

“string” : 表示要发送的按键指令字符串, 需要放在英文双引号中。

#### 1. 基本键

一般来说, 要发送的按键指令都可以直接用该按键字符本身来表示, 例如要发送字母 “x”, 使用 “WshShell.SendKeys “x”” 即可。当然, 也可直接发送多个按键指令, 只需要将按键字符按顺序排列在一起即可, 例如, 要发送按键 “happy”, 可以使用 “ WshShell.SendKeys “happy” ”。

#### 2. 特殊功能键

对于需要与 Shift、Ctrl、Alt 三个控制键组合的按键, SendKeys 使用特殊字符来表示:

Shift-----WshShell.SendKeys "+"

Ctrl-----WshShell.SendKeys "^"

Alt-----WshShell.SendKeys "%" (注意 这样使用时, 不用大括号括起这些特殊字符。)

由于 “+”、“^” 这些字符用来表示特殊的控制按键了, 如何表示这些按键呢? 只要用大括号括住这些字符即可。例如: 要发送加号 “+”, 可使用 “WshShell.SendKeys “{+}””

另外对于一些不会生成字符的控制功能按键, 也同样需要使用大括号括起来按键的名称。

例如要发送回车键, 需要用 “ WshShell.SendKeys “{ENTER}” ” 表示;

发送向下的方向键用 “ Wshell.SendKeys “{DOWN}” ” 表示。

Space-----WshShell.SendKeys " "

Enter-----WshShell.SendKeys "{ENTER}"

←-----WshShell.SendKeys "{RIGHT}"

↑-----WshShell.SendKeys "{UP}"

F1-----WshShell.SendKeys "{F1}"

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

| 按键            | 代码                          |
|---------------|-----------------------------|
| BACKSPACE     | {BACKSPACE}, {BS}, 或 {BKSP} |
| BREAK         | {BREAK}                     |
| CAPS LOCK     | {CAPSLOCK}                  |
| DEL or Delete | {Delete} 或 {DEL}            |
| DOWN ARROW    | {DOWN}                      |
| END           | {END}                       |
| ENTER         | {ENTER} 或 ~                 |
| ESC           | {ESC}                       |
| HELP          | {HELP}                      |
| HOME          | {HOME}                      |
| INS or Insert | {Insert} 或 {INS}            |
| LEFT ARROW    | {LEFT}                      |
| NUM LOCK      | {NUMLOCK}                   |
| PAGE DOWN     | {PGDN}                      |
| PAGE UP       | {PGUP}                      |
| PRINT SCREEN  | {PRTSC}                     |
| RIGHT ARROW   | {RIGHT}                     |
| SCROLL LOCK   | {SCROLLLOCK}                |
| TAB           | {TAB}                       |
| UP ARROW      | {UP}                        |
| F1            | {F1}                        |
| F2            | {F2}                        |
| F3            | {F3}                        |
| F4            | {F4}                        |
| F5            | {F5}                        |
| F6            | {F6}                        |
| F7            | {F7}                        |
| F8            | {F8}                        |
| F9            | {F9}                        |
| F10           | {F10}                       |

**Tips:**如果需要发送多个重复的单字母按键，不必重复输入该字母，SendKeys 允许使用简化格式进行描述，使用格式为“{按键 数字}”。例如要发送 10 个字母“x”，则输入“WshShell.SendKeys “{x 10}””即可。

---

按下 F5 刷新桌面

```
Dim WshShell, Path, i
Set WshShell = Wscript.CreateObject("Wscript.Shell")
WshShell.SendKeys "{F5}"
```

---

电脑的自动重启

```
set WshShell = CreateObject("Wscript.Shell")
WshShell.SendKeys "``{ESC}u"
WshShell.SendKeys "R"
```

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

---

### 启动任务管理器

```
set WshShell = CreateObject("Wscript.Shell")
WshShell.SendKeys "^+{ESC}"
```

---

### QQ 消息连发

```
Dim WshShell
Set WshShell= Wscript.createObject("Wscript.Shell")
WshShell.AppActivate "bomb"
for i=1 to 60
Wscript.Sleep 800
WshShell.SendKeys "What do you say"
WshShell.SendKeys i
WshShell.SendKeys "%s"
next
```

---

### 自动到百度搜索歌曲：white flag

```
Dim WshShell, Path, i
Set WshShell = Wscript.CreateObject("Wscript.Shell")
WshShell.Run ("IEXPLORE.EXE")
Wscript.Sleep 2000
WshShell.AppActivate "about:blank-Microsoft Internet Explorer"
WshShell.SendKeys "+{TAB}"
WshShell.SendKeys "http://mp3.baidu.com"
Wscript.Sleep 800
WshShell.SendKeys "{ENTER}"
Wscript.Sleep 3000
WshShell.SendKeys "white flag"
Wscript.Sleep 800
WshShell.SendKeys "{ENTER}"
```

---

### 自动关机

```
Dim WshShell
Set WshShell=Wscript.CreateObject("Wscript.Shell")
Wscript.Sleep 2000
WshShell.Run "shutdown -r -t 120"
wscript.sleep 6000
WshShell.Run "shutdown -a"
```

---

### 在记事本中输入 Happy Birthday! 并保存为 birth.txt

```
Dim WshShell
Set WshShell=Wscript.CreateObject("Wscript.Shell")
WshShell.Run "notepad"
```

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

```

Wscript.Sleep 1500
WshShell.AppActivate "无标题 - 记事本"
WshShell.SendKeys "H"
Wscript.Sleep 500
WshShell.SendKeys "a"
Wscript.Sleep 500
WshShell.SendKeys "p"
Wscript.Sleep 500
WshShell.SendKeys "p"
Wscript.Sleep 500
WshShell.SendKeys "y"
Wscript.Sleep 500
WshShell.SendKeys ""
Wscript.Sleep 500
WshShell.SendKeys "B"
Wscript.Sleep 500
WshShell.SendKeys "i"
Wscript.Sleep 500
WshShell.SendKeys "r"
Wscript.Sleep 500
WshShell.SendKeys "t"
Wscript.Sleep 500
WshShell.SendKeys "h"
Wscript.Sleep 500
WshShell.SendKeys "d"
Wscript.Sleep 500
WshShell.SendKeys "a"
Wscript.Sleep 500
WshShell.SendKeys "y"
Wscript.Sleep 500
WshShell.SendKeys "!"
Wscript.Sleep 500
WshShell.SendKeys "%FS"
Wscript.Sleep 500
WshShell.SendKeys "b"
Wscript.Sleep 500
WshShell.SendKeys "i"
Wscript.Sleep 500
WshShell.SendKeys "r"
Wscript.Sleep 500
WshShell.SendKeys "t"
Wscript.Sleep 500
WshShell.SendKeys "h"
Wscript.Sleep 500
WshShell.SendKeys "%S"
Wscript.Sleep 500
WshShell.SendKeys "%FX"

```

---

|       |          |           |
|-------|----------|-----------|
| 自动化测试 | Version: | 1.3       |
| 文档    |          | 2006-5-17 |

制作能自动定时存盘的记事本

’ 第一部分： 定义变量和对象

```
Dim WshShell, AutoSaveTime, TXTFileName
AutoSaveTime=300000
Set WshShell=Wscript.CreateObject("Wscript.Shell")
TXTFileName=InputBox("请输入你要创建的文件名(不能用中文和纯数字)：")
```

’ 第二部分： 打开并激活记事本

```
WshShell.Run "notepad"
Wscript.Sleep 200
WshShell.AppActivate "无标题 - 记事本"
```

’ 第三部分： 用输入的文件名存盘

```
WshShell.SendKeys "^s"
Wscript.Sleep 300
WshShell.SendKeys TXTFileName
Wscript.Sleep 300
WshShell.SendKeys "%s"
Wscript.Sleep AutoSaveTime
’ 第四部分： 自动定时存盘
While WshShell.AppActivate (TXTFileName)=True
WshShell.SendKeys "^s"
Wscript.Sleep AutoSaveTime
Wend
Wscript.Quit
```

---

**自动死机** o(∩\_∩)o...

```
DIM WSHSHELL
SET WSHSHELL=Wscript.CreateOBJECT("Wscript.SHELL")
' WSHSHELL. RUN "
' Wscript. SLEEP 1000
WSHSHELL. SENDKEYS "{ENTER}"
```

|       |           |     |
|-------|-----------|-----|
| 自动化测试 | Version:  | 1.3 |
| 文档    | 2006-5-17 |     |

[/quote]

## 113. QTP 录制模式应该分为 3 种

[QuickTest Pro](#)

**QTP 录制模式应该分为 3 种**

1. **普通模式**:只记录系统中的对象
2. **analog 模式**:记录鼠标,键盘的动作(轨迹),如:鼠标拖动,点击等等 (模拟录制)
3. **low-level 模式**:可以对 QTP 不支持的对象的记录 (低级录制)

在菜单项 test 内有两个 Analog recording,和 low-level recording,但只有在 recoding 的时候,才能切换,快捷键是 ctrl+shift+F4 和 ctrl+shift+F3