QuickTest Professional 错误处理机制概述

作者: gy21st
电子邮件: gy21st@yahoo.com

目录

1. 概述

本文针对 QuickTest Professional（后面将简称 QuickTest 或 QTP）提供的运行时错误处理机制进行详细描述。内容包括各种错误处理方式的功能的介绍，各种机制的作用范围的分析，以及错误处理机制同时作用时的优先级别的分析。本文可以为 QTP 用户采用何种错误处理机制对测试脚本的运行时错误进行处理提供参考。

注 1： 本文只对错误机制进行描述，并不涉及对各种错误机制适应性的分析。
注 2： 所有功能介绍及实例基于 QuickTest Profession 9.2 版本，其他版本未经验证。
注 3：知识能力所限，错误疏漏之处在所难免，若有问题，请与作者联系。

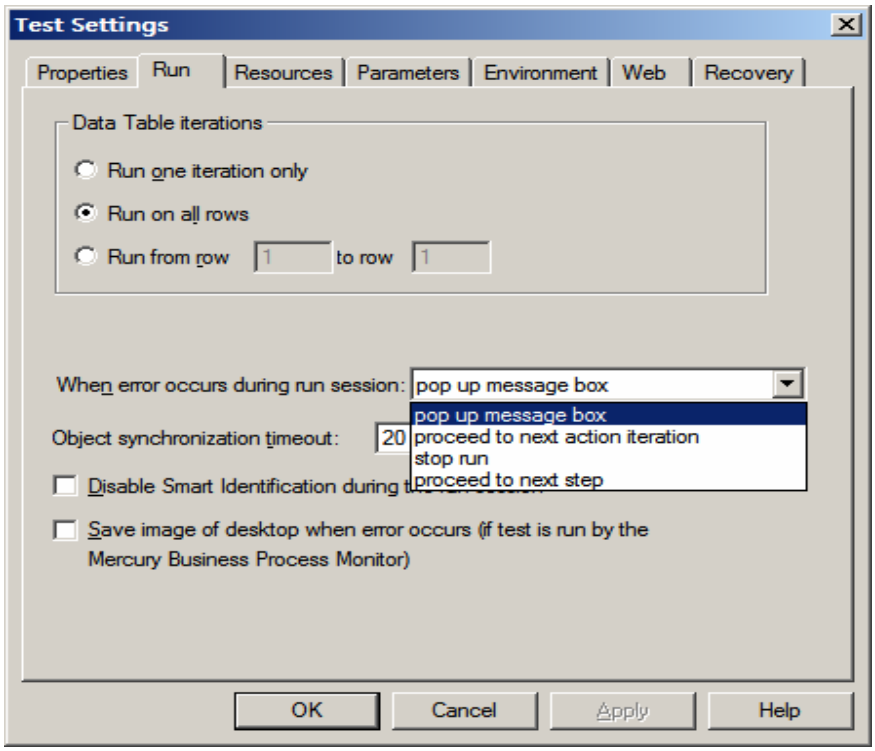
2. 三种错误处理方式介绍

QuickTest Professional 中有三种错误处理机制，在这里定义为：

- 全局错误响应： Test Setting 中 Run 设置
- VBScript On Error 错误处理: On Error Resume Next
- 错误恢复场景： Recovery Scenarios

2.1 全局错误响应

全局错误响应在 Test Setting 的 Run 选项中进行设置：File ➔Settings➔Test Settings Dialog ➔Run Tab➔When error occurs during run session



四种设置选项如下：

- pop up message box： QuickTest 在出现错误时显示一个错误消息对话框。要继续或结束运行会话，您必须单击该消息框中的某个按钮
- process to next action iteration： QuickTest 在出现错误时继续下一个 Action 循环。
- stop run： QuickTest 在出现错误时停止测试
- process to next step：QuickTest 在出现错误时跳过错误语句，继续下一步骤

全局错误响应是 QTP 的系统缺省错误处理，就是说当没有使用其他错误处理方式时，系统会自动调用在这里指定的方式进行错误处理。

2.2 VBScript 的 On Error 错误处理方式

由于 QTP 使用了 VBScript 作为脚本语言，自然地，VBScrip 的错误处理方式在 QTP 中都适用。在 VBScript 中，与错误处理相关的三者为：On Error Resume Next 语句, On Error GoTo 0 语句，以及 Err 对象

2.2.1 On Error Resume Next

一旦这个语句已被处理，脚本引擎将继续运行后面的程序，而不理会已经发现的任何错误。

2.2.2 On Error GoTo 0

使用 On Error Goto 0 语句恢复缺省的错误处理行为。在运行这个语句后，发生的运行期错误将导致缺省错误处理。在 QTP 中，缺省错误处理就是全局错误响应。

2.2.3 Err 对象

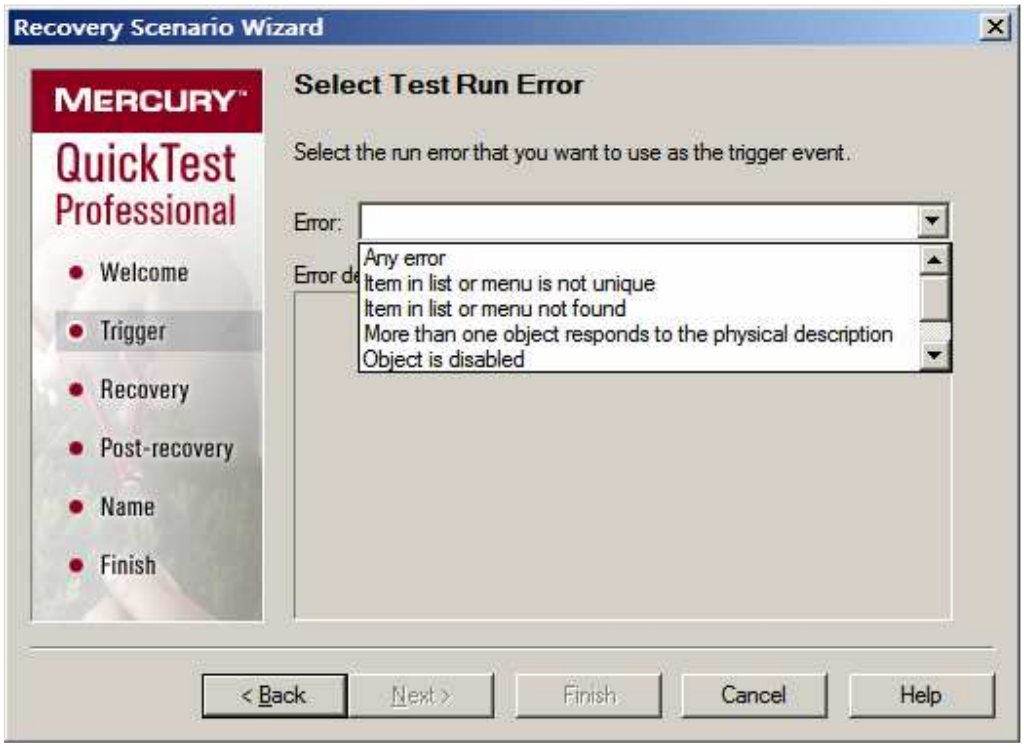
Err 对象重要的属性有三个：Number, Source, Description。分别是错误号，错误来源，错误描述

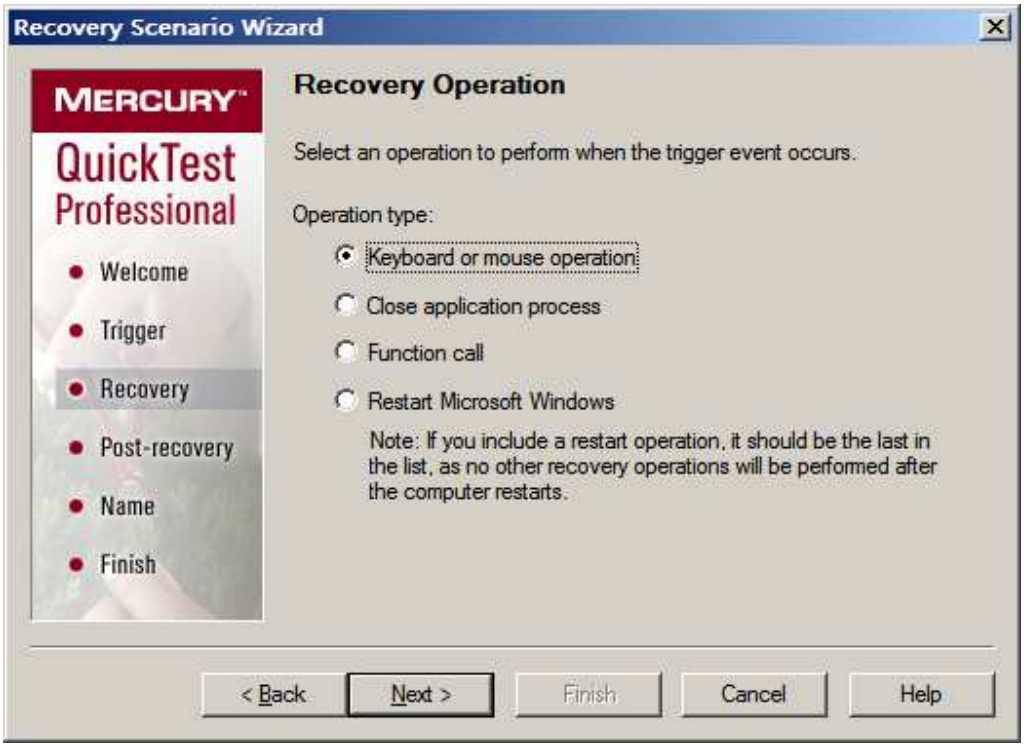
Err 对象会在运行时错误发生的时候赋予新的值，旧的值会被舍弃。
Err 对象不会受到函数调用的影响，它完全是全局的。
Err 对象可以用 Clear 方法清空
任何时候调用 On Error Resume Next 或者 On Error GoTo 0 的时候，都会清空 Err 对象

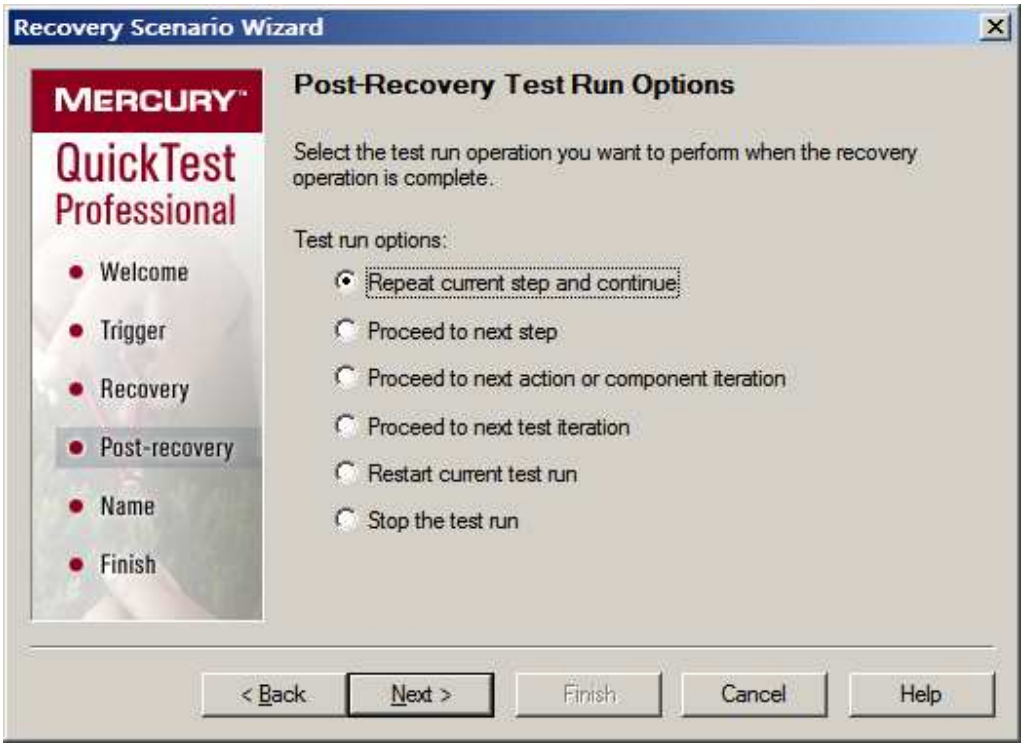
2.3 恢复场景 Recovery Scenarios

Resources➔Recovery Scenarios Manager...可以创建恢复场景。具体请参考帮助文档，这里只是说明对于运行错误的处理。以下是恢复场景设置的几张截图。


Recovery Scenario Wizard
MERCURY
QuickTest Professional
Welcome
Trigger
Recovery
Post-recovery
Name
Finish
Select Trigger Event
Select the trigger event type that will trigger the recovery operation.
Pop-up window
A window pops up in an opened application during the test run.
Object state
The property values of an object in your application match specified values. You can specify property values for each object in the hierarchy.
Test run error
A step in your test does not run successfully.
Application crash
An open application fails during the test run.
< Back
Next >
Finish
Cancel
Help


Recovery Scenario Wizard
MERCURY
QuickTest Professional
Welcome
Trigger
Recovery
Post-recovery
Name
Finish
Select Test Run Error
Select the run error that you want to use as the trigger event.
Error:
Any error
Item in list or menu is not unique
Item in list or menu not found
More than one object responds to the physical description
Object is disabled
< Back
Next >
Finish
Cancel
Help





在恢复场景中触发事件 TriggerEvent 中的一个选项是 Test Run Error，表示在运行过程中遇到错误时所采取的恢复处理方式。我们看一看下图可以对哪几种错误进行恢复处理，即错误恢复的触发条件：

- Any Error
- Item in list or menu is not unique

- Item in list or menu not found
- More than one object responds to the physical description
- Object is disable
- Object not found
- Object not visible

恢复处理(Recovery Operation)有以下几种方式：

- Keyboard or mouse operation
- Close application process
- Function call
- Restart Microsoft Windows

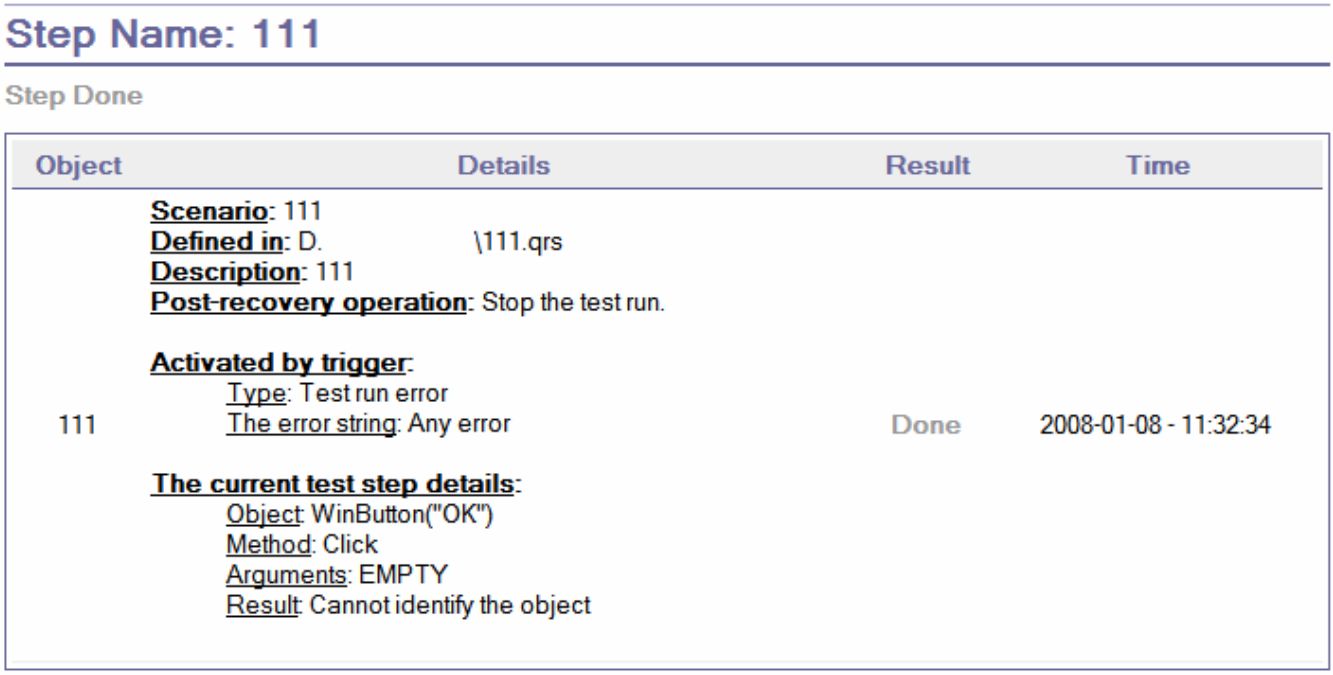
这里不再详述。具体参考帮助文档。

从恢复场景的触发条件可以看出，所有错误处理都是针对测试对象发生的错误而言的（Any Error 指列表中所有的其他错误类型），它并不处理被零除、非法赋值，内存分配错误等等 VBScript 的运行时错误。这是和其它两种错误处理方式不同之处。为了说明这一点，我们不妨再看看 Recovery Operation 中 Operation Type 如果选择 Function Call，函数原型定义如下所示，所有的函数参数都必须包含有被测对象，触发函数必须遵循原型定义，否则函数无法执行。

```
Following is the prototype for each trigger type:
Test run error trigger
OnRunStep
(
[in] Object as Object: The object of the current step.
[in] Method as String: The method of the current step.
[in] Arguments as Array: The actual method's arguments.
[in] Result as Integer: The actual method's result.
)

Pop-up window and Object state triggers
OnObject
(
[in] Object as Object: The detected object.
)

Application crash trigger
OnProcess
(
```

```
[in] ProcessName as String: The detected process's Name.
[in] ProcessId as Integer: The detected process' ID.
)
```

以下是一个测试结果中当触发错误恢复场景时的错误描述，也可以看出触发条件和被测对象相关。



Step Name: 111

Step Done

Object	Details	Result	Time
111	**Scenario**: 111 **Defined in**: D. \111.qrs **Description**: 111 **Post-recovery operation**: Stop the test run. **Activated by trigger**: Type: Test run error The error string: Any error **The current test step details**: Object: WinButton("OK") Method: Click Arguments: EMPTY Result: Cannot identify the object	Done	2008-01-08 - 11:32:34

再来看错误恢复的后处理方式(Post-Recovery Test Run Options)有以下几种：

- Repeat current step and continue
- Proceed to next step
- Proceed to next action or component iteration
- Proceed to next test iteration
- Restart current test run
- Stop the test run

我们看到 Proceed to next step/ Proceed to next action or component iteration/ Stop the test run 这三种方式在全局错误处响应理方式中也是存在的，Proceed to next step 这种方式更是和 On Error Resume Next 处理方式也是相同的。稍后会就此作进一步叙述。

3. 三种错误处理方式的作用域

本节阐述三种错误处理方式的作用范围，为了方便比较说明，这里假设三种错误处理方式为：

- 全局错误响应： Process to next step
- VBScript 错误处理： On Error Resume Next
- 错误恢复后处理方式： Proceed to next step

因为这三种处理方式效果一样，以后没有特殊说明，都采用以上设置，可以方便比较

3.1 全局错误响应

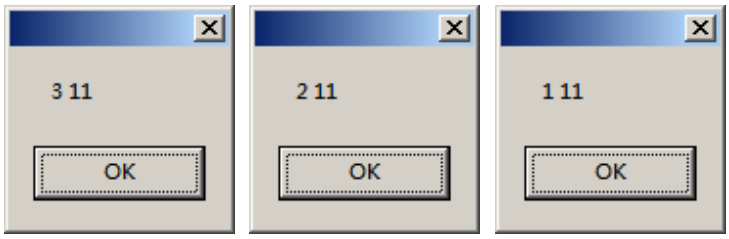
原则： 对该 test 中的所有 Action 均生效。也就是说在 Action 里没有其他错误处理方式的时候，脚本运行过程中遇到的错误都会按照全局错误响应的设置进行处理

3.1.1 对 Action 中直接定义的函数：

对所有函数生效。如下例，如果全局错误响应设置成 process to next step 函数 testA 中的被零除语句 i=1/0 会跳过，三个 msgbox 均可以执行。错误代码会逐级返回。

```
Sub testA()
    Dim i
    i = 1/0
    Msgbox " 3 "  & Err.number
End Sub

Sub testB()
    testA
    Msgbox " 2 " & Err.Number
End Sub

testB
Msgbox " 1 " & Err.Number
```

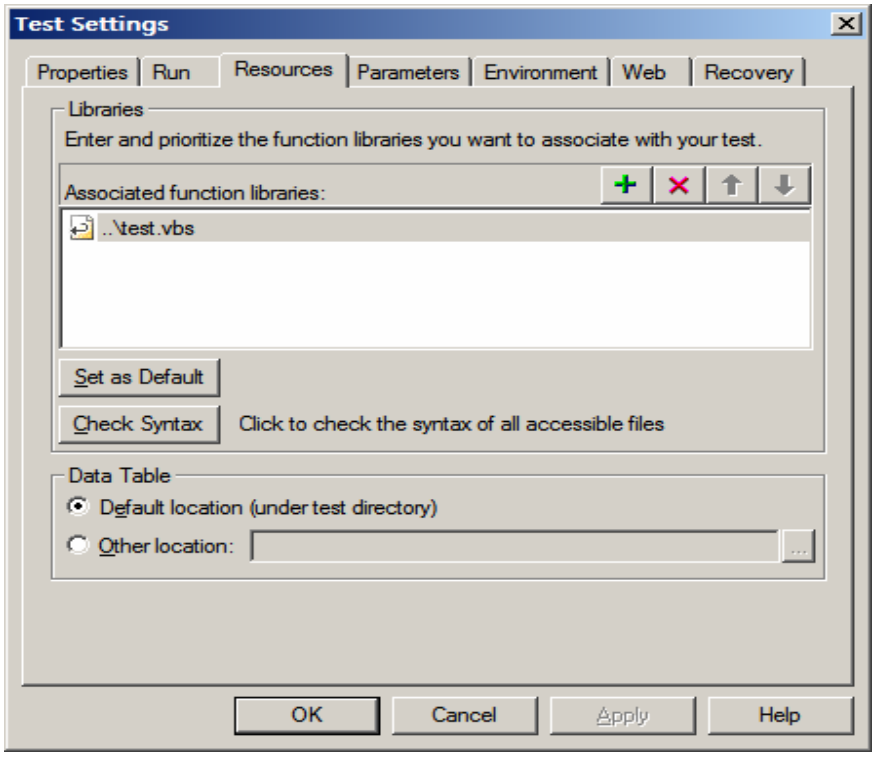
运行结果为：



3.1.2 对用 ExecuteFile 引入的函数：

与 Action 中直接定义的函数处理方式完全一样

3.1.3 对于 Function Library 中定义的函数



与 Action 中直接定义的函数处理方式基本一致，对所有函数生效。但不同的是，对于错误代码，只能在 Function Library 中传递，而不会传递到 Action 中来。

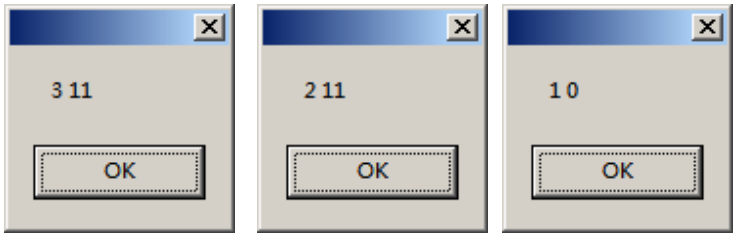
如下例， 在 test.vbs 中同样定义如下函数，并把 test.vbs 加入到 Libraries 中

```
Sub testA()
   Dim i
   i = 1/0
   Msgbox " 3 "  & Err.Number
End Sub

Sub testB()
   testA
   Msgbox " 2 " & Err.Number
End Sub
```

然后再 action1 中调用

```
testB
Msgbox " 1 " & Err.Number
```

三次 msgbox 输出结果如下：



可以看出，Error Number 最后传到调用 Action 中时，已经被清 0 了。

3.1.4 对于 Action 的嵌套调用

如之前所述，对 Test 中的所有 Action 均生效，自然嵌套调用的 Action 也不例外。同 Function Library 一样，错误码只能在 Action 内部传递，也就是说被调用 Action 中产生的错误码不能传递到调用 Action 中去。同样的例子：

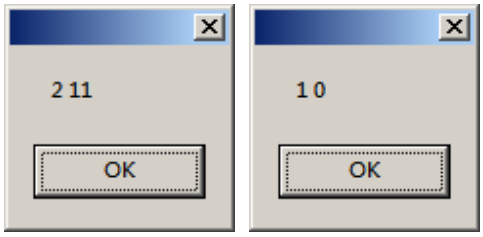
ActionA 中的代码如下：

```
RunAction "ActionB", oneIteration
msgbox " 1 " & err.number
```

ActionB 中的代码如下

```
Dim i
i = 1/0
Msgbox " 2 "  & Err.Number
```

运行结果为：



可以看出，Error Number 最后传到 ActionA 中时，已经被清 0 了。

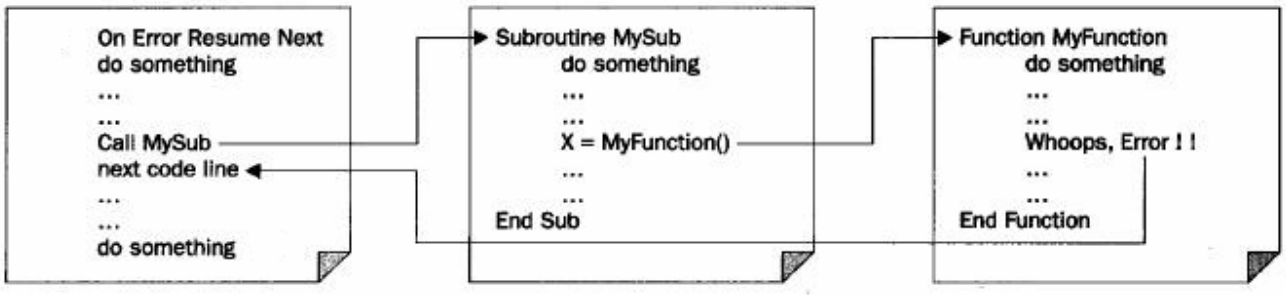
3.2 VBScript 的 On Error 错误处理方式

原则： 只对语句所在函数或 Action 中该语句之后的代码生效，对子函数，调用 Action 均无效。

3.2.1 对 Action 中直接定义的函数：

函数中错误抛出，不会继续执行错误代码

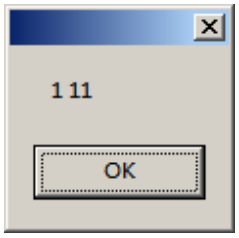
一个错误在函数/子程序中出现时，如果没有运行 On Error Resume Next 语句，那么错误将被交给调用它的环境，这个过程一直重复到找到运行 On Error Resume Next 语句的环境继续运行，或者找到缺省的脚本错误处理器。错误处理过程如下图所示：



如下例，运行到 testA i=1/0 出错，错误交给调用它的环境 testB，同样 testB 把错误交给调用它的环境，发现 On Error Resume Next，然后 Msgbox " 1 " & Err.Number 会执行，Error Number 就是在 testA 中产生的错误码

```
Sub testA()
   Dim i
   i = 1/0
   Msgbox " 3 "  & Err.Number
End Sub

Sub testB()
   testA
   Msgbox " 2 " & Err.Number
End Sub

testB
Msgbox " 1 " & Err.Number
```

运行结果为：



3.2.2 对用 ExecuteFile 引入的函数：

与对 Action 中直接定义的函数的处理完全一致

3.2.3 对于 Function Library 中定义的函数

Function Library 中定义的函数遇到错误抛出，执行系统缺省错误处理

3.2.4 对于 Action 的嵌套调用

被调用 Action 中的语句遇到错误抛出，执行系统缺省错误处理

3.3 错误恢复场景 Recovery Scenarios

原则： 任何函数调用和 Action 中，只要满足触发条件，就会触发错误恢复处理和后处理

前面讲过，错误恢复场景的触发条件是以下 6 种

- Item in list or menu is not unique
- Item in list or menu not found
- More than one object responds to the physical description
- Object is disable
- Object not found
- Object not visible

对于 VBScript 的其他运行时错误，并不会触发错误恢复场景，当然也不存在错误恢复处理及后处理。但一旦在任何函数或者 action 中满足触发条件，都会进行错误恢复处理及后处理，也就是说，其作用范围为所有 Action 和函数/子程序中。当错误恢复完成后（包括错误处理及后处理），错误会被清 0。

如果错误恢复的后处理方式为 Proceed to next step，处理完成后再次碰到 VBScript 错误时如何处理呢？如新产生错误并不能满足触发条件，则原错误恢复中设置后处理方式不会继续生效，此时会遵循缺省错误响应或其他指定的错误处理。

值得注意的是，如果错误恢复的后处理方式为 Repeat current step and continue 或者 Restart current test run，再次运行到该语句仍重复同样的错误，可能会导致死循环。

4. **三种错误处理方式的优先级别**

原则：

- 错误恢复场景中的错误处理优先级最高 (Recovery Operation)
- On Error Resume Next 优先级次之
- 错误恢复场景中的后处理方式优先级别再次之 (Post-Recovery Test Run Options)
- 缺省错误响应优先级别最低

也就是说，只要满足恢复场景的触发条件，就会进行错误恢复处理(Recovery Operation)。但是其后处理方式的优先级别和错误恢复处理的优先级别是不同的。如果触发错误恢复处理的语句之前有 On Error Resume Next，就不会执行错误恢复后处理中所定义的处理方式。当没有找到以上任何错误处理方式的定义时，才执行缺省错误处理。

注意： On Error GoTo 0 其实就是指采用缺省错误处理，不可和 On Error Resume Next 处理的优先级别等同起来

举例说明如下。 假定

- 全局错误响应设置为 pop up message box

新建错误恢复场景并应用到 test 中

- 触发条件为 Test run error(Any error)
- 错误恢复处理为 Close application process (notepad.exe)
- 错误恢复后处理方式为 Stop the test run

新建一个 Action1，把 notepad.exe 相关对象都已添加进对象库，打开 notepad.exe（不打开 font 设置对话框）。

当 Action1 中代码如下

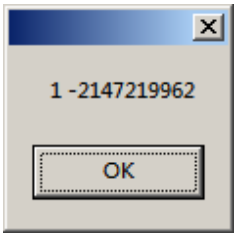
```
Dim i
Window("Notepad").Dialog("Font").WinButton("OK").Click

i=1/0
Msgbox " 1 " & Err.Number & " " & Err.Description
```

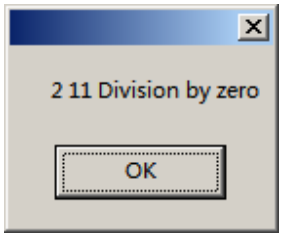
第二行由于没有找到 Dialog("Font")对象，会进行错误恢复处理，关闭 notepad.exe，并执行错误恢复后处理 stop test run。Msgbox 并不执行。

当 Action1 中代码如下

```
On Error Resume Next
Dim i
Window("Notepad").Dialog("Font").WinButton("OK").Click

Msgbox " 1 " & Err.Number & " " & Err.Description

i=1/0
Msgbox " 2 " & Err.Number & " " & Err.Description
```

第三行由于没有找到 Dialog("Font")对象，会进行错误恢复处理，关闭 notepad.exe，但并不会执行错误恢复后处理，而是继续运行直到 test 结束。msgbox 运行结果为



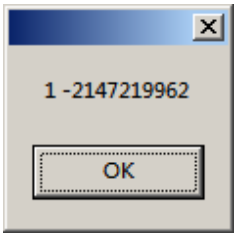


以上两种情况都不会进行缺省错误响应的处理。

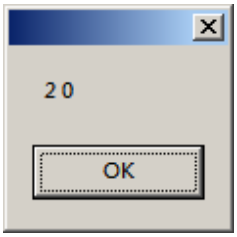
再次更改代码

```
On Error Resume Next
Dim i

Window("Notepad").Dialog("Font").WinButton("OK").Click
Msgbox "1 " & Err.Number & " " & Err.Description

On Error Goto 0
Msgbox "2 " & Err.Number & " " & Err.Description

i=1/0
Msgbox "3 " & Err.Number & " " & Err.Description
```

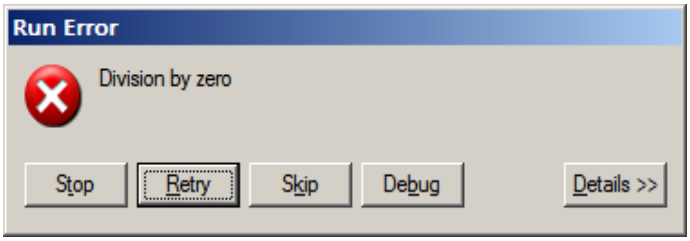
第三行由于没有找到 Dialog("Font")对象，会进行错误恢复处理，关闭 notepad.exe，但并不会执行错误恢复后处理，第一个 Msgbox 运行结果为



On Error GoTo 0 后，采用系统缺省错误处理方式（全局错误响应），同时会清空 Err 对象，第二个 Msgbox 运行结果为



执行到 i=1/0，由于缺省错误响应为 pop up message box，会弹出如下信息，之后根据用户选择继续执行或结束运行，或者进入 Debug 模式。



【参考文档】

1. Quick Test Professional Help
2. WEB 开发 - ASP - 基础教程 处理错误(http://read.gz235.com/others/web/web/asp/index1/29.htm)