项目编号				密级	秘密
修订历史					
生效日期	版本号	版本说明	作者	审核	批准
	V1.0	草稿版	刘毅		

基于QC管理的自动化设计框架规范指南

目录

1 引言

1.1.1 目标与范围

本文档将从流程管理的角度对自动化测试框架行综合概述，涉及到自动化测试框架的总体流程定义、相关的活动、角色、输入、输出等。本文档主要针对自动化测试岗位上的人员。

1.1.2 术语和缩略语

序号	术语/缩略语	全称和说明
1.	QTP	Quickly Test Professional
2.	QC	Quality Center
3.	文件服务器	映射的网络驱动器和共享公共存储空间

1.1.3 参考资料

2 整体说明

2.1.1 自动化测试框架概述

小规模自动化，写几十个上百个自动化测试用例，就无所谓框架了，随便录制一下脚本再参数化一下就可以了。但是写成千上万个自动化测试用例的时候，不仅开发时非常费力，写出来的脚本大量冗余，而且开发后根本无法维护。其实大规模的测试自动化，要录制编写大量的脚本，从本质上说也是开发一个测试系统。

2.1.2 目标和约束

QTP 只是一个基础工具，它的 KeywordView、Action、CheckPoint 等概念，都是针对小规模自动化的用户。不适合大规模自动化的专业用户自动化测试的管理不仅仅局限于脚本的层面，而是从整个自动化测试需求管理的流程出发，将自动化测试需求管理流程分解为相应的活动，详细定义各个活动的相关角色，输入、输出等等。

3 自动化框架组件规范

3.1 自动化测试项目的目录整体结构

下面是自动化回归测试架构，可扩展 Business Component 或其他组件，只需在子系统下陆续添加新的目录即可，每做变更需要考虑清楚历史的脚本影响和后续维护的成本，不能单凭现有的优势或好处就做草率的决定，给已经完成的系统带来负担。

3.2 数据管理（TestData）

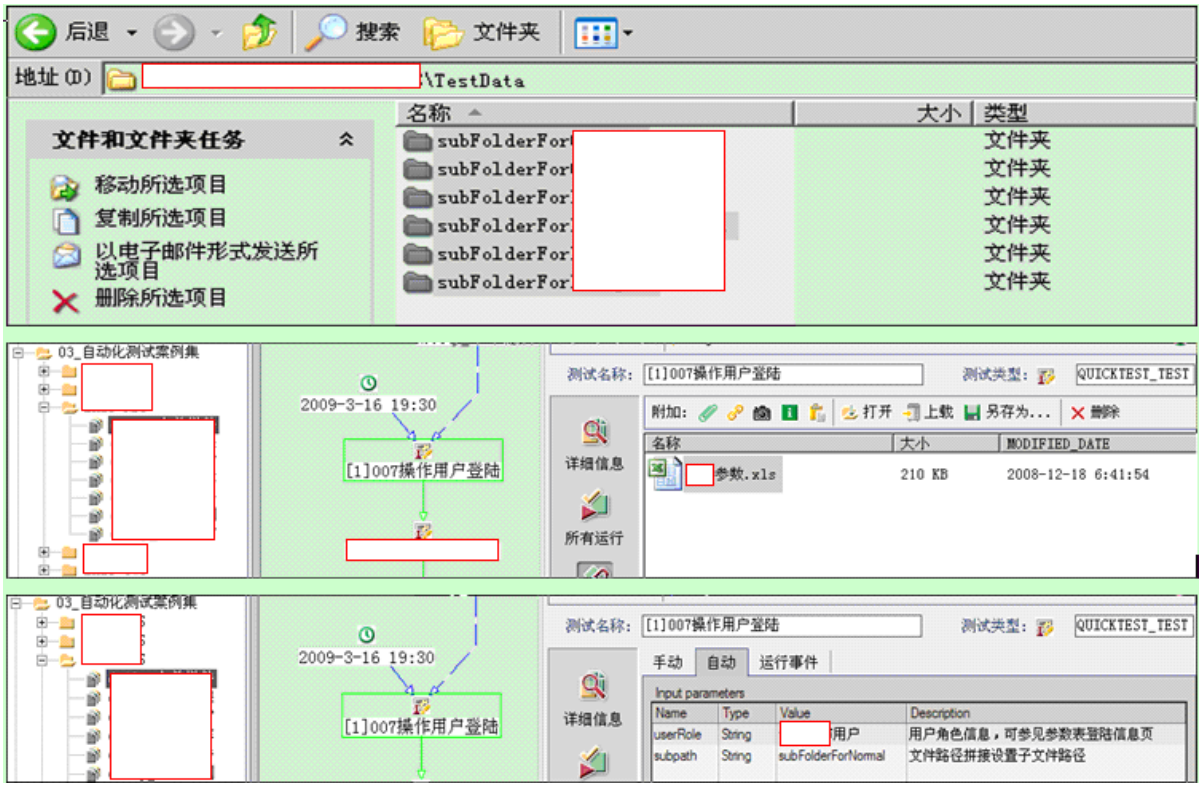
3.2.1 参数文件规范

1. EXCEL 参数文件：原则上一个执行流对应一个参数文件，便于流程控制管理；
2. EXCEL 文件 SHEETS：一个脚本或一个 ACTION 对应一个 SHEET；
3. 字段命名：与对象库中的对象域定义名称保持一致，使用中文定义字段名称；
4. 所有页面可编辑域全部参数化。

3.2.2 参数配置基础方法

测试数据通常使用 EXCEL 文件保存，并保存于指定文件服务器的 TestData 目录下。例如：共享目录\\192.168.0.158\share\【subSystemName】\TestData\【subTestPath】目录用来保存测试数据，传递测试流程参数。子路径和参数文件自主命名即可。

【注】：subTestPath 可依据系统特点决定是否使用，如系统中存在复用脚本或 Action，这些脚本、Action 使用了参数表并且有并行运行的可能，则需要使用子路径进行多流程并行的控制支持：即，将二级路径作为一个参数写在脚本的导入参数文件路径中，在不同的执行流中进行不同的子路径配置。这样操作需要注意测试实验室中只能通过测试执行流视图进行配置，而执行网格则对执行流无效，配置过程如下：



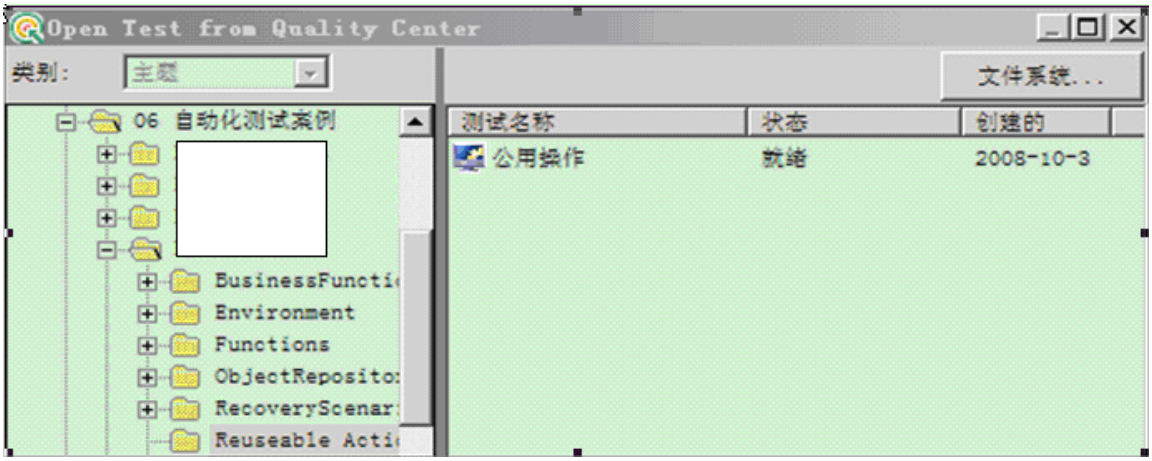
3.2.3 数据驱动高级方法

使用 EXCEL 文件存储测试数据，保存于 QC 指定目录下（测试集附件），如 Root\自动化测试案例集\【SubSystemName】\【TestSetName】,并且使用运行时 VBS Function 操作 QC 测试集附件(attachment)和 QC 测试集运行时参数（runtime parameter），读写这些文件进行流程控制。简单示例如下：

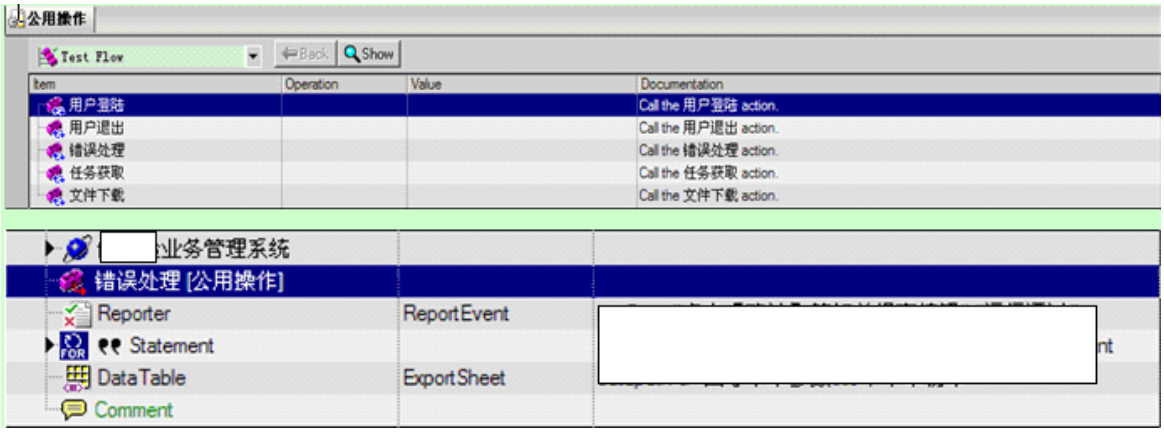
```
Function UpLoadAttachmentToQC(FilePath)
        Set ObjCurrentTest = QCUtil.CurrentTest.Attachments
        Set ObjAttch = ObjCurrentTest.AddItem(Null)
        ObjAttch.FileName = FilePath
        ObjAttch.Type = 1
        ObjAttch.Post
        ObjAttch.Refresh
End Function

Function RemoveQCAttachements(NameOfTheFile)
        var_count= QCUtil.CurrentTest.Attachments.NewList("").count
        For i= 1 to var_count
                If    QCUtil.CurrentTest.Attachments.NewList("").Item(2).Name =NameOfTheFile Then
                        AttachmentID = QCUtil.CurrentTest.Attachments.NewList("").Item(1).ID
                        QCUtil.CurrentTest.Attachments.RemoveItem(AttachmentID)
                Else
                        Reporter.ReportEvent micFail,"Please provide the valid filename","Please provide the valid filename"
                End If
        Next
End Function
```

3.3 可复用操作(ReusableActions)

在 QTP 里开发，要真正做到脚本分层复用、减少冗余的程度，自己组织脚本结。我们基于功能性创建了可重复使用脚本,也有基于单个功能创建的脚本以适应不同的测试用例测试他们不同的功能需求来调用。



例如，在契约里将【工作台任务处理】、【错误处理】、【登陆退出】、【文件下载】等这些操作频率较高的 Action 做成可复用脚本 Action 提供给其他脚本调用，免除每个脚本都要重新编写的额外工作。

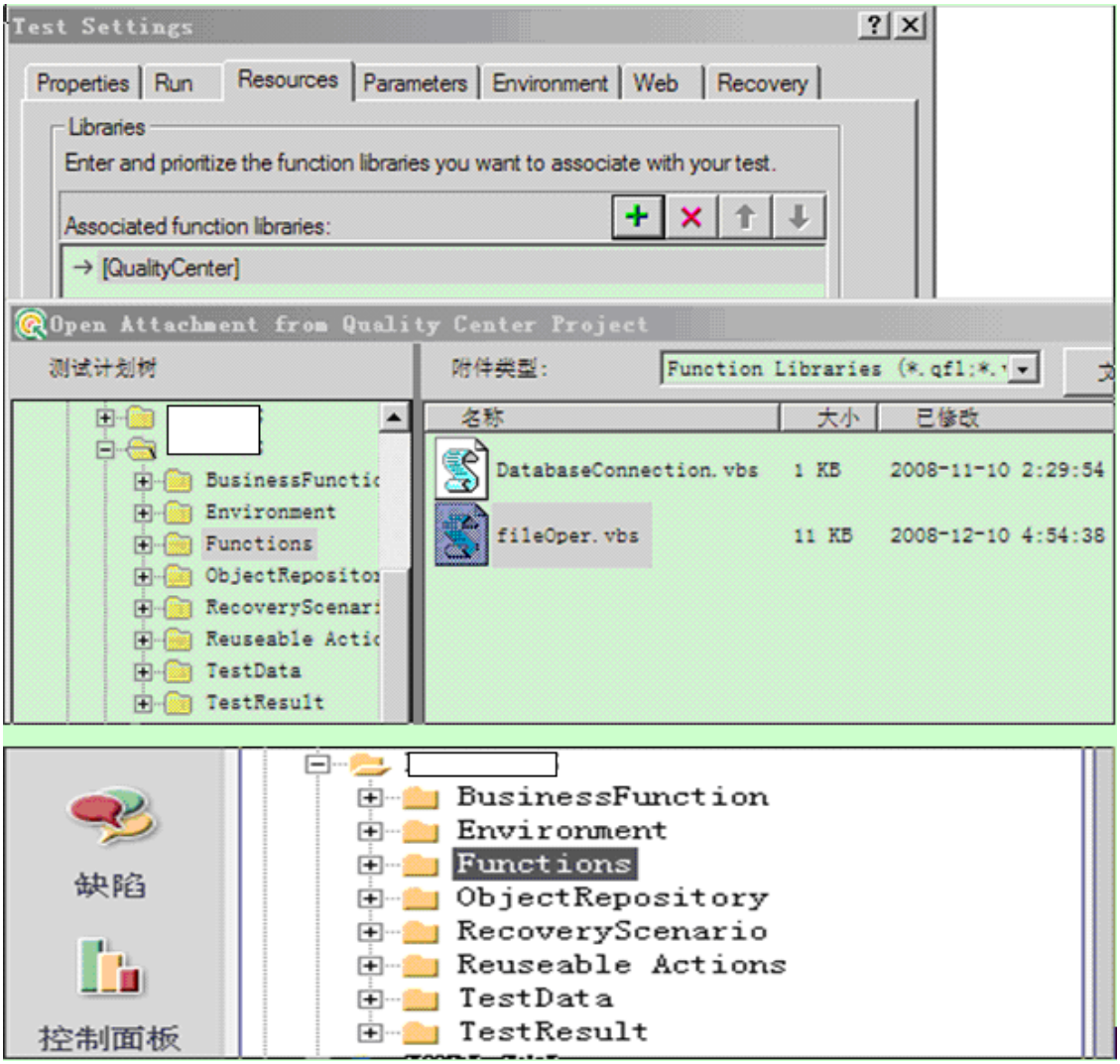


3.4 外部驱动程序(Function)

这些是独立于程序的可重复使用程序,可以包括数据库操作、文件系统操作、QC_API 调用操作等等应用程序。建立独立文件路径同时保存与文件服务器和 QC 指定路径下，在 QTP 的设置菜单 File-->Settings-->Resources 下引用。路径结构如下：

Ø 文件服务器：\\hsh-0045\share\【subSystemName】\Fuctions

Ø QC 服务器：Root\自动化测试案例\【SubSystemName】\Fuctions



3.5 对象库管理(ObjectRepository)

3.5.1 对象管理规范

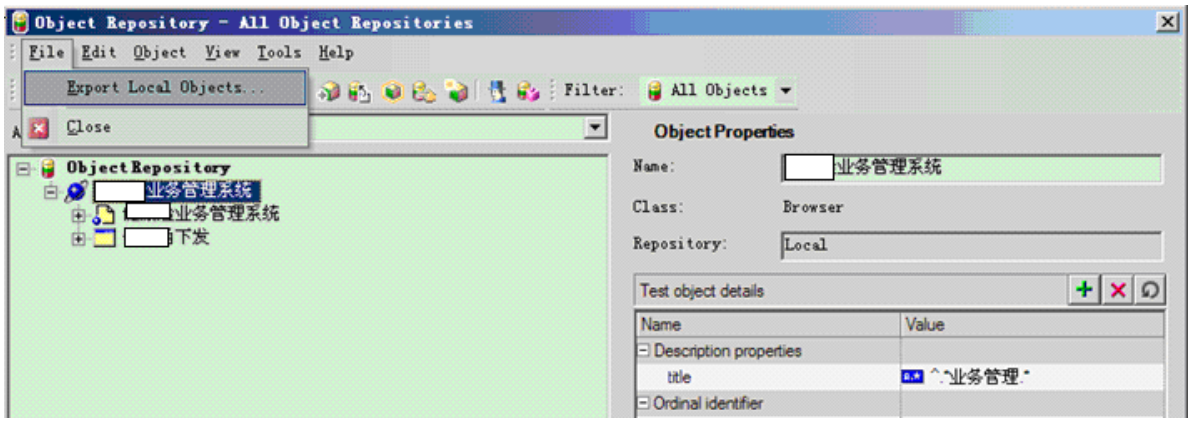
所有对象按层次自上而下，其命名必须遵从汉语说明的原则，如条形码用【条形码号】表示对象名称，不能够使用 barCode 甚至 newbusinessPreliminaryAcessbarCode.do 这种编码原始命名方式，除非能够提供一份精确的所有对象的中文名称说明附件。因为脚本维护者是系统测试负责人而不一定是脚本编写人员，这种命名会给脚本维护带来很大的障碍，所以这种习惯必须抛弃。

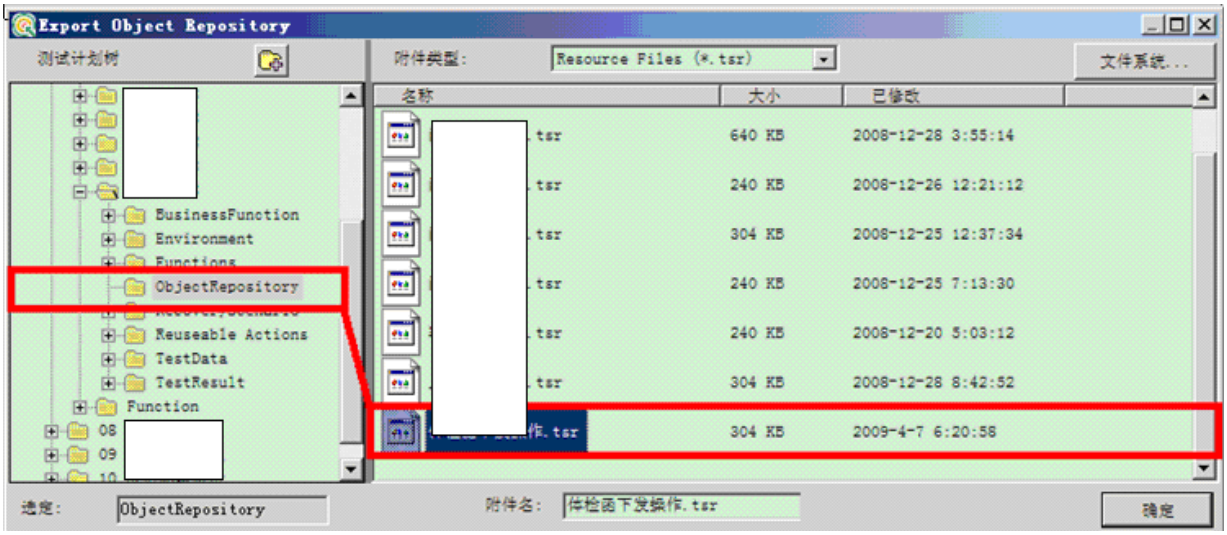
另外，对象库的集中管理是一个可以研究的课题，目前没有考量出集中统一管理和各自分散管理的利弊和成本，目前允许自主实验。

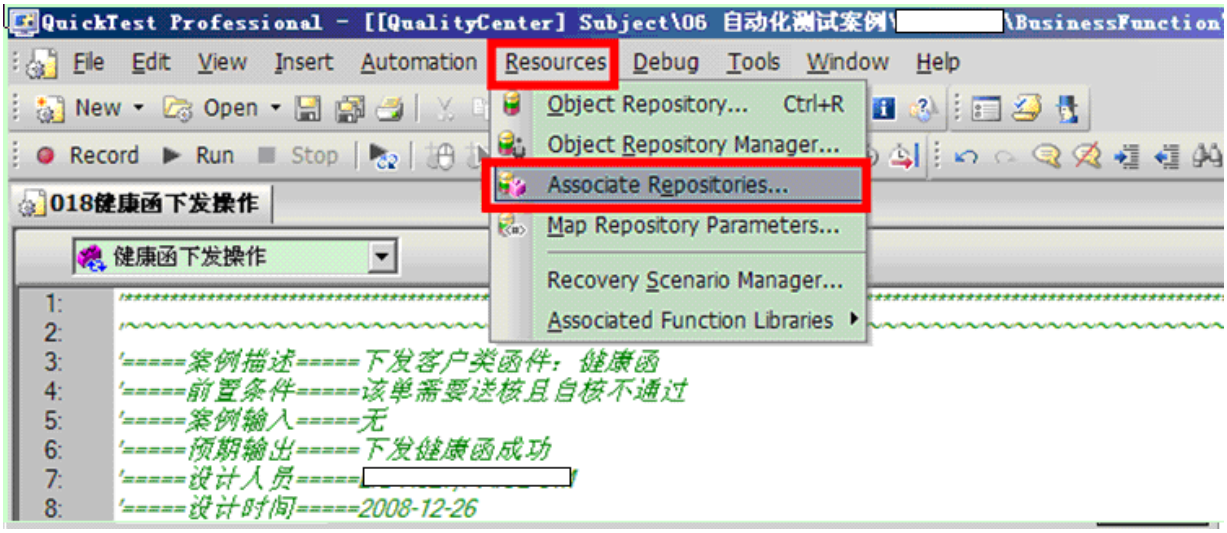
3.5.2 对象复用

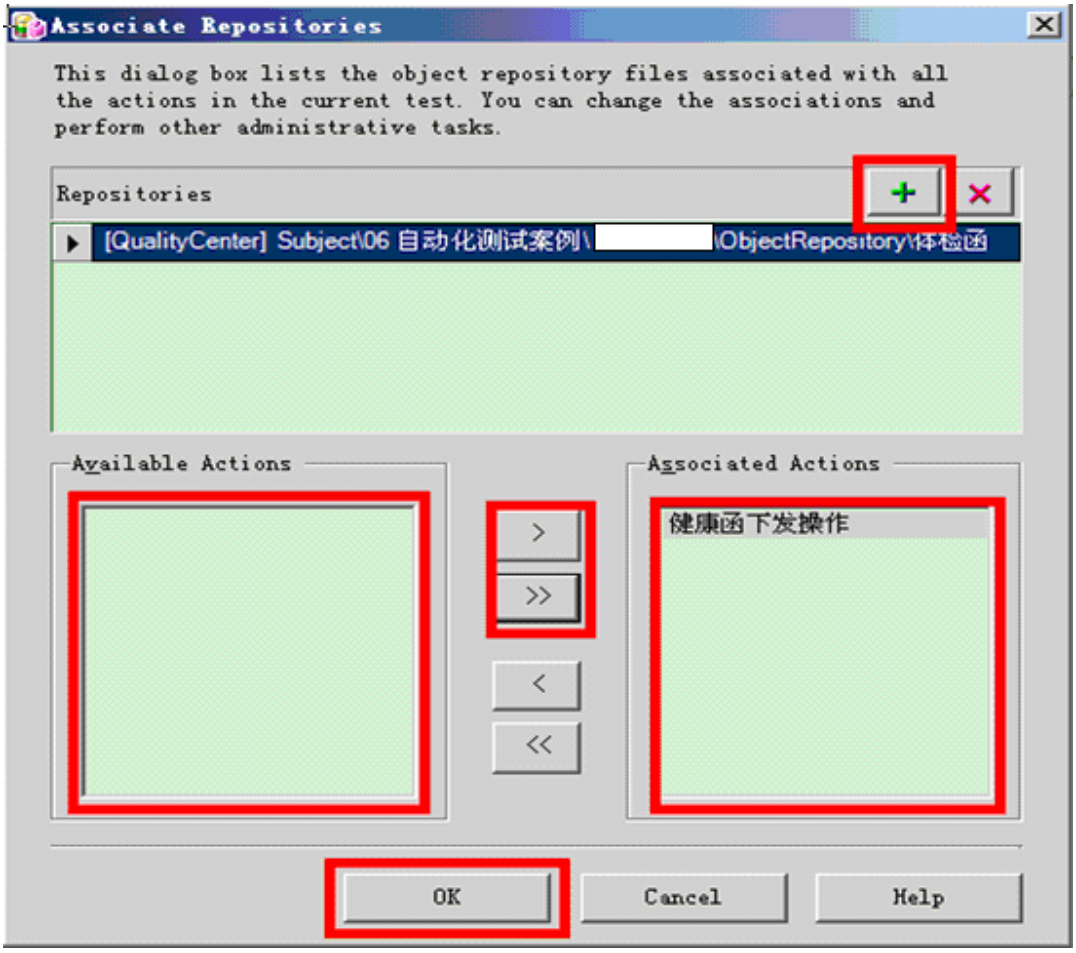
对象库用来存储常用的窗体，模块或工程，增强资源共用，减少重复劳动。在实际自动化测试应用中,对于同一系统来说操作界面的相似性在所难免，所以有很多对象的添加是重复的、没有必要的，对于这种相似、相同或部分相同的界面可以保存一个基准的对象库（无需全面兼容），提供后续其他脚本的复制使用。

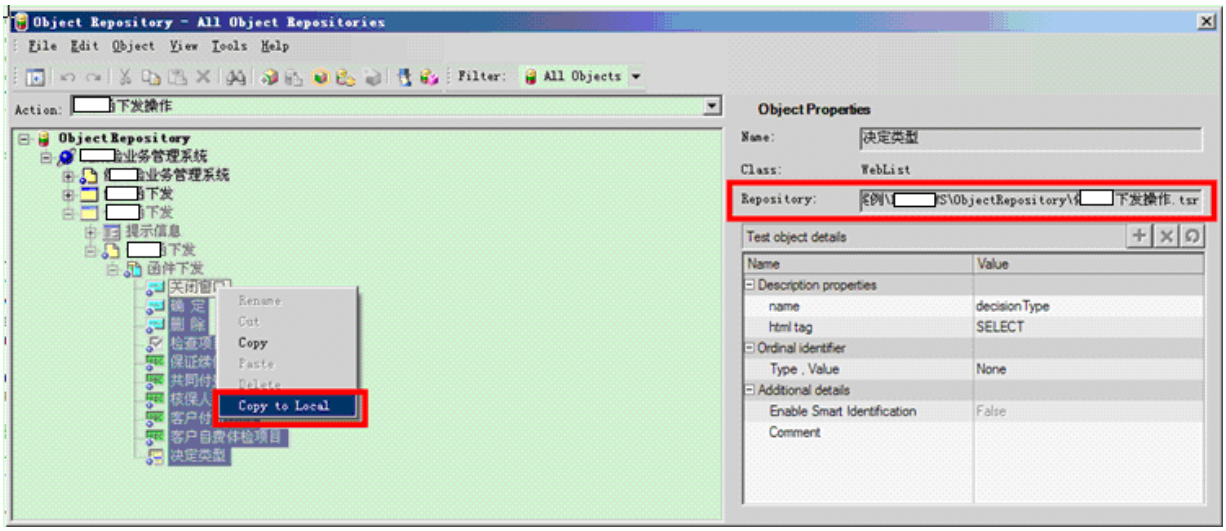
这个基准的对象库应该包含几类页面（元素），例如：某业务系统有 5 类客户类函件的下发、打印、扫描、回销操作，共计脚本 4×5= 20 个，可以保存其中一类函件的操作几面元素到对象库中，其余的 4 类函件可以引用、复制到当前脚本、做简单编辑即可，如图：




Export Object Repository
测试计划树
附件类型:
Resource Files (*.tsr)
文件系统...
名称
大小
已修改
BusinessFunction
Environment
Functions
ObjectRepository
Reuseable Actions
TestData
TestResult
Function
640 KB
2008-12-28 3:55:14
240 KB
2008-12-26 12:21:12
304 KB
2008-12-25 12:37:34
240 KB
2008-12-25 7:13:30
240 KB
2008-12-20 5:03:12
304 KB
2008-12-28 8:42:52
304 KB
2009-4-7 6:20:58
ObjectRepository
确定

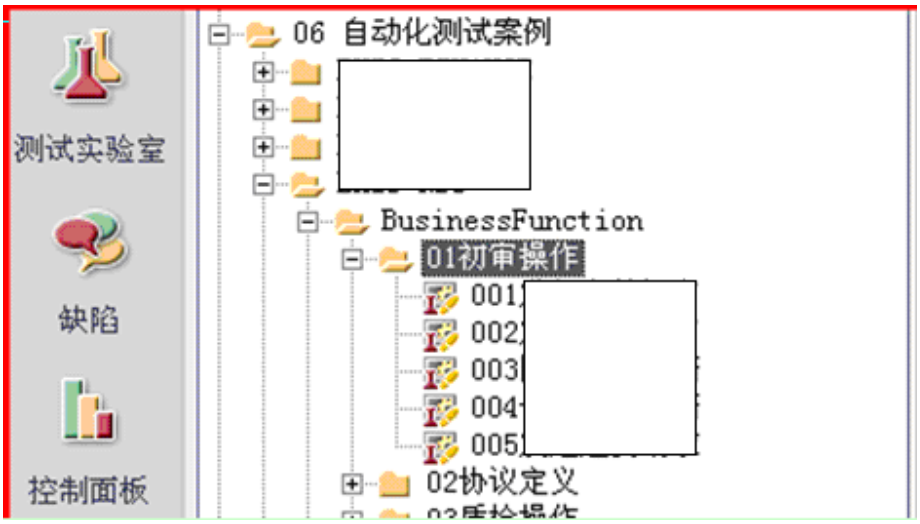

QuickTest Professional - [[QualityCenter] Subject\06 自动化测试案例\
\BusinessFunction
File Edit View Insert Automation Resources Debug Tools Window Help
New Open
Record Run Stop
018健康函下发操作
健康函下发操作
Object Repository... Ctrl+R
Object Repository Manager...
Associate Repositories...
Map Repository Parameters...
Recovery Scenario Manager...
Associated Function Libraries
'=====案例描述=====下发客户类函件：健康函
'=====前置条件=====该单需要送核且自核不通过
'=====案例输入=====无
'=====预期输出=====下发健康函成功
'=====设计人员=====
'=====设计时间=====2008-12-26


Associate Repositories
This dialog box lists the object repository files associated with all the actions in the current test. You can change the associations and perform other administrative tasks.
Repositories
[QualityCenter] Subject\06 自动化测试案例\
\ObjectRepository\体检函
Available Actions
Associated Actions
健康函下发操作
OK
Cancel
Help


Object Repository - All Object Repositories
File Edit Object View Tools Help
Filter: All Objects
Action:
下发操作
ObjectRepository
业务管理系统
业务管理系统
下发
下发
提示信息
下发
函件下发
关闭窗口
确 定
删 除
检查项
保证体
共同付
核保人
客户付
客户自费体检项目
决定类型
Rename
Cut
Copy
Paste
Delete
Copy to Local
Object Properties
Name:
决定类型
Class:
WebList
Repository:
例\ S\ObjectRepository\ 下发操作.tsr
Test object details
Name
Value
Description properties
name
decisionType
Html tag
SELECT
Ordinal identifier
Type , Value
None
Additional details
Enable Smart Identification
False
Comment

3.6 功能性脚本(BusinessFunction)

用于存放主要测试模块脚本，该目录可以有不同功能相关的子目录组成，其目录结构可以根据系统特点调整，建议目录层次统一，示例如下：



3.7 测试结果(TestResult)

- 在测试脚本中通过 Reporter.ReportEvent 的方法实现结果报告，结合错误处理等其他操作对系统操作的每一个步骤（对照测试案例原始步骤）做相应的结果报告，保证运行 PASS 则系统功能 PASS。

- 在测试脚本中通过 EXCEL 测试结果文件保存测试结果（不推荐使用）。所有关于脚本执行的结果都保存为 EXCEL 文件格式。该文件应保存于对应的子系统目录下，主要是文件系统操作来实现人工的判断。对于测试结果统计，我们可以通过脚本的方法控制来报告,让测试脚本自动将测试相关的结果填写到“测试结果”的 EXCEL 表格中，该结果主要包括测试过程中间状态以及比对结果，测试时间，测试机器，测试结果，还有与测试具体相关的唯一数据标识，如条形码，险种代码等。

- 关键功能点的测试快照，对关键点（类似数据提交、确认等）的处理提交都需要进行判断，并且最低要求对错误信息必须有截图：

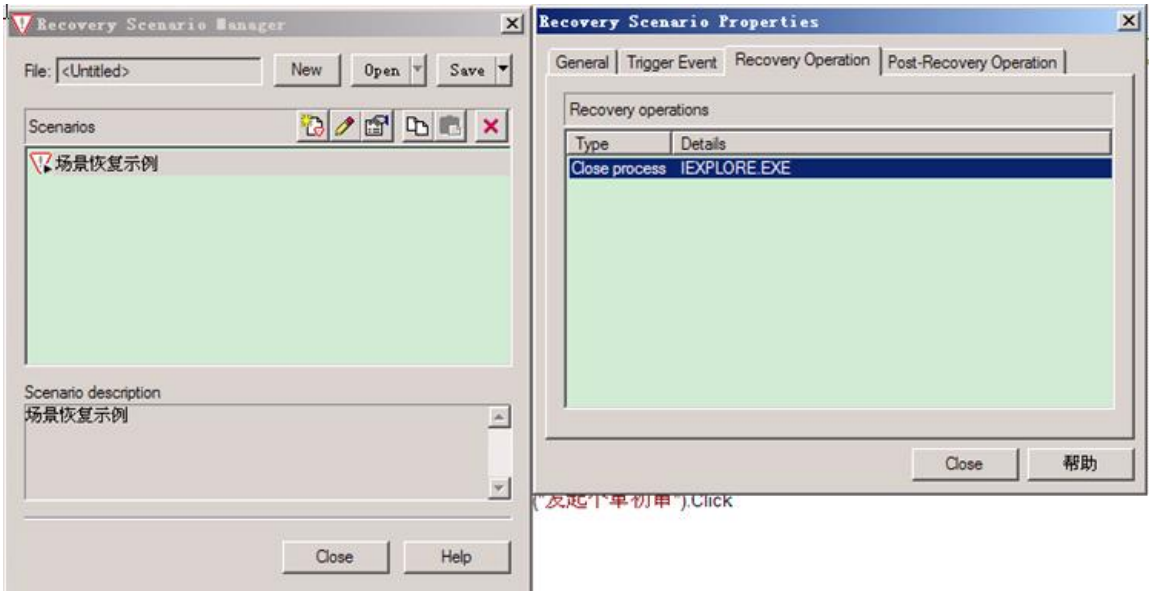
```
Browser("title:=^.*业务管理.*").CaptureBitmap respath&fileName
```

路径使用配置好的结果路径参数（参见 Environment），fileName 建议使用【操作名称】+【时间戳】来确保文件唯一性。

3.8 场景恢复(RecoveryScenario)

在测试过程中，经常发生不能预期的事件，错误和应用程序崩溃。针对这些问题，QTP 提供了恢复场景的功能（Recovery Scenarios）。通过菜单 Tools->Recovery Scenario Manager 进入自定义，并将自定义好的场景文件保存于框架结构的 Recovery Scenario 目录中，对于自动化测试执行过程中的种种以外情况需要做出判断，已保证 QTP 和应用程序出错后，及时关闭 QTP 执行功能和关闭应用程序，使得下面的测试任务能够被执行。

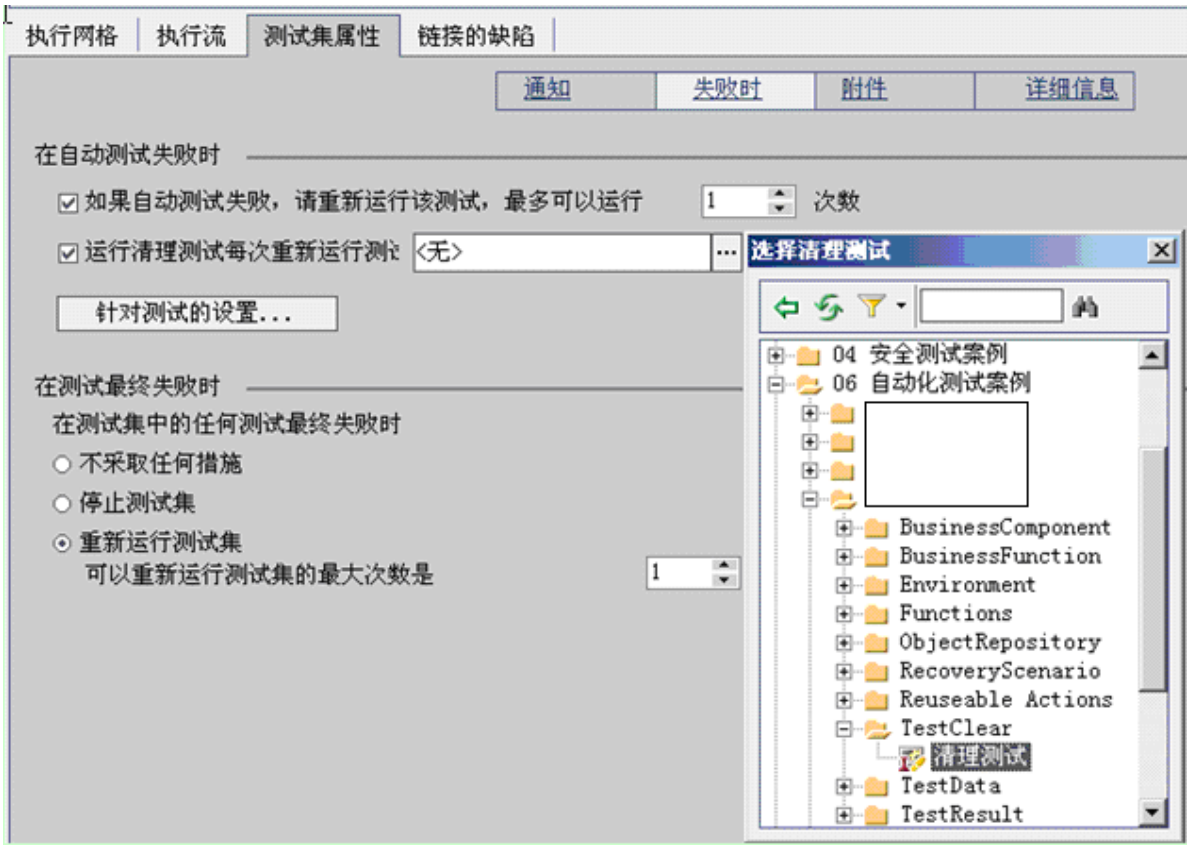
一般情况不推荐使用场景恢复，因为运行异常恢复之后的流程运行的可靠性无从验证，很可能运行的结果是不真实的，而一些没有严格先后执行顺序关系的流程是可以使用场景恢复的。



3.9 清理测试（TestClear）

路径结构如下：该组件不强制要求使用，需要结合是否【失败时重新运行】来进行配置，而且需要结合特定的测试脚本设置。

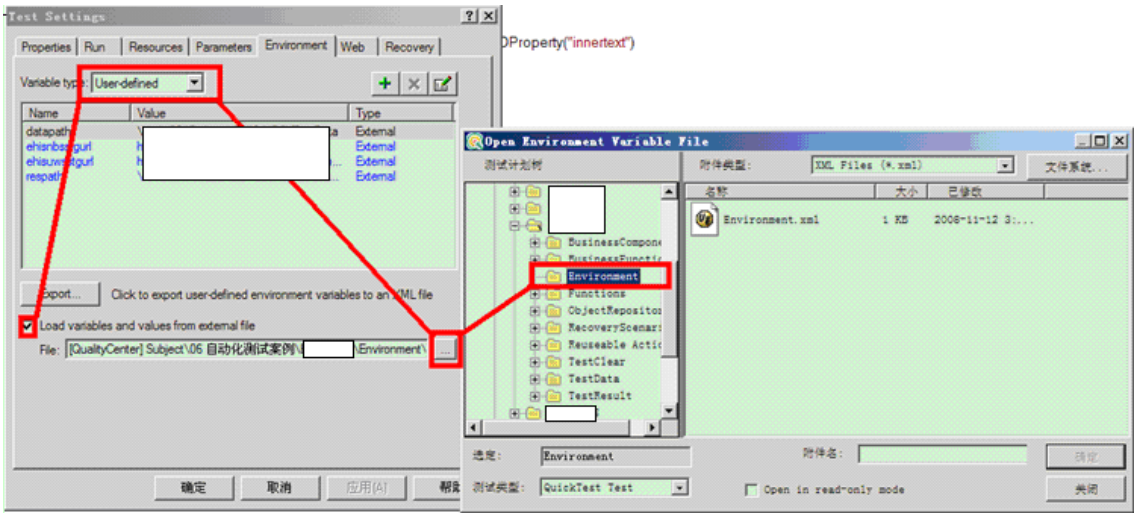
脚本内容包含 CloseProcessByName、SystemUtil.Run 等等系统进程处理操作，为重新运行提供干净的系统环境和数据环境。



3.10 配置参数（Environment）

使用 QTP 提供的 Environment.Value 方法取得事先配置好的公用变量的方法，Environment 包含文件变量和系统变量（包括 OS 信息、Host 信息等等），文件变量定义示例如下：

```
<Environment>
	<Variable>
		<Name>ehisnbsstgurl</Name>
		<Value>http://ehis-nbs-stg.paic.com.cn/ehis</Value>
</Variable>
	<Variable>
		<Name>datapath</Name>
		<Value>\\hsh-0045\share\EHIS-NBS\TestData\</Value>
	</Variable>
	<Variable>
		<Name>respath</Name>
		<Value>\\hsh-0045\share\EHIS-NBS\TestResult\</Value>
	</Variable>
</Environment>
```

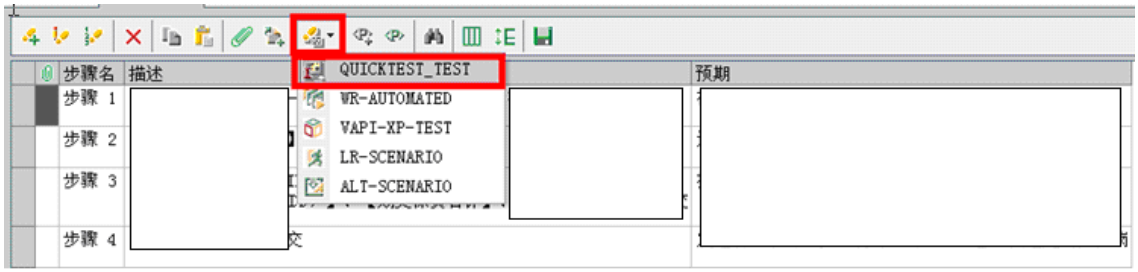


由于 Environment 文件只需在 QC 中更新所有引用的脚本即可自动更新，所以 Environment 的使用比较方便，故而要求通过使用 Environment 来配置公共参数，例如参数文件读取路径和结果保存路径、测试环境的不同 URL 等，通过参数文件路径的再次参数化来实现流程的灵活控制。

4 自动化设计过程规范

合理的自动化设计过程会给自动化开发带来很高的效率和可维护性，能够节约大量的自动化成本，而没有组织的自动化经常会出现反复开发、维护成本高、运行不稳定、扩展性不够好的问题。根据多个系统的经验总结，以下设计过程是目前比较优化的方法：

1. 手工测试案例设计（结合.do 整理），要求步骤清晰，描述和预期可衡量，不可模棱两可；



2. 生成自动化测试案例，参见下图操作，这样做的好处是，清晰的测试案例描述生成清晰的测试脚本注释，为脚本的编写提供良好好的依据和良好的可读性；

3. 分析流程组合：以最少最简单的流程组合覆盖到尽可能多的测试场景（.do），优秀的回归测试案例会标注同一个案例在什么样的输入下覆盖到哪些事务。完成流程拆分，将每个案例划分到每一个流程中去，也就是事前拟定尚未完成的自动化执行流；

4. 根据执行流的规划配置参数文件，按照参数文件配置的原则，每一个流程一个参数文件，则需要为这一个拟定的流程中的每一个案例做一个 SHEET。由回归测试案例设计人员（需求分析人员）进行参数文件设计，要求参见数据管理部分。

5. 根据生成自动化测试案例设计测试脚本并且单个调试，细节参见测试脚本编写规范；

6. 组合执行流，配置测试集运行时 Test Parameter 和测试数据参数；

7. 调试执行流，提交复审和每日回归平台验收。

5 测试脚本书写规范

Ø 注释规范

```
1:  Dim PolicyInResult              '页面中的案件号
2:  Dim InsuredNoInResult           '页面中的保单号
3:  Dim InsuredNameInResult          '页面中的被保险人
4:  Dim CarIdInResult               '页面中的车牌号
5:  Dim PolicyInDatatable           '准备好需要验证的中的案件号
6:  Dim InsuredNoInDatatable        '准备好需要验证的中的保单号
7:  Dim InsuredNameInDatatable      '准备好需要验证的中的被保险人
8:  Dim CarIdInDatatable            '准备好需要验证的中的车牌号
9:
10: Dim PolicyValueInWebTableAfterSearch        '查询后页面表格上案件号
11: Dim InsuredNoInWebTableAfterSearch          '查询后页面表格上保单号
12: Dim InsuredNameInWebTableAfterSearch         '查询后页面表格上被保险人
13: Dim CarIdInWebTableAfterSearch              '查询后页面表格上车牌号
```

1. 变量在 QTP 中无需单独做显式定义即可编译通过，但是为脚本的可读性，我们约定所有的变量都需要做显式声明，并且标注其对应的定义；
2. 关键操作步骤引用原始注释，以手工测试案例描述为准；
3. 循环体和一些条件判断等必须做明确说明，注释特殊处理的原因或理由；
4. 脚本中需要包含一个描述（例如，它是干什么用的）和特别用途（例如，回归测试）的文件头。脚本的文件头应该包括脚本的作者，所有者，创建和修改日期，脚本可以追溯到的需求识别符，脚本所支持的业务范围

Ø 命名规范

1. 脚本命名：遵照手工测试案例命名，结合岗位菜单和功能命名，4 到 15 个汉字
2. Action 命名：按照脚本内容，使用中文命名拆分的 Acition 内容，2 到 10 个汉字
3. 对象库对象命名:遵照页面元素的中文命名进行命名,要求从 Browser、Window、Page、Frame、Dialog、WebElement……等等，一律使用简短的中文命名，要求可编辑域对象名称与参数文件中字段名称一致；
4. 变量命名：采用英文大小写结合的方式例如 barCode、policyApplicationNo 等等；
5. 参数命名：Action 的 Parameter 和 Test 的 Parameter 变量名称需要遵从与脚本变量一致的要求，并且做注释性说明。

6 执行流配置规范

Ø 测试集基础设置

1. 执行流：不允许有 2 个或更多的分支；
2. 失败时：选择如果运行失败，重新运行 1 次，并且结合清理测试进行配置；
3. 测试集运行通知（可不配置）：配置运行结果通知消息、收件人列表、触发邮件发送的条件；
4. 测试集详细信息和测试集附件配置同手工测试案例集。

Ø 执行条件

有严格的先后关系的执行流都必须保证做执行条件设置，因为第一个脚本运行失败之后，后续脚本没有得到停止运行的指令就会一直使用错误的或者不存在的数据进行无谓、无用的运行，给后续其他流程的运行制造了很多麻烦，尤其是时间上浪费了很多资源。

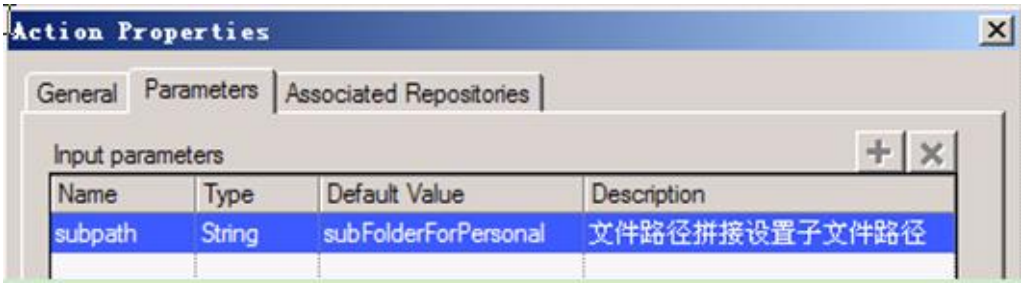
一般的流程里至少必须有一个必须设置的运行条件，那就是登陆系统操作和后续操作之间的联系，道理很简单，没有登陆成功，后续运行操作都是在浪费时间。

而登陆之后很多的查询等类似操作，他们之间并没有直接或者内在的联系，至少不共享测试数据，这样的脚本之间就无需定义执行条件，除非编写脚本时对 Run Failed 做了很精准的判断，断定失败之后环境必定是不可用的，否则客观上减少了测试的覆盖率。

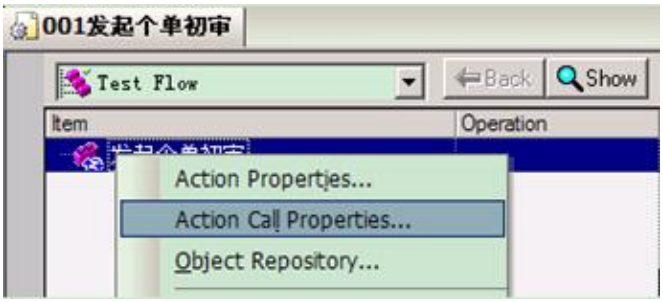
Ø 运行时参数

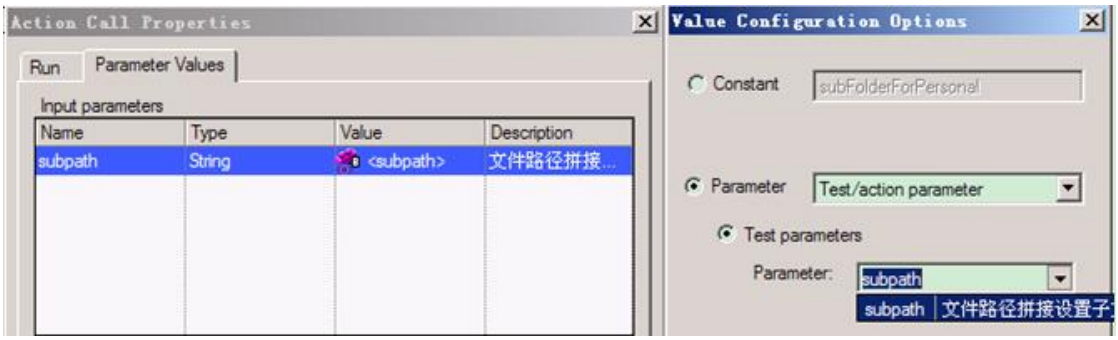
运行时参数的使用是一个灵活处理流程控制的方法，配置要求和方法参见数据管理部分；脚本中对应的配置（必须的）过程如下：

1. 分别在 KeyWordView 中配置 Action 参数，在 Settings-->Parameters 中做相同的参数配置：



2. 在 KeyWordView 中（一定是 Test Flow 中）配置 Action 的 Action Call Properties 为可变的参数，否则运行时仍旧是脚本中定义的常量，而非测试集参数。





3. 在测试实验室中对执行流中的【配置】项修改对应配置的参数运行时值即可：