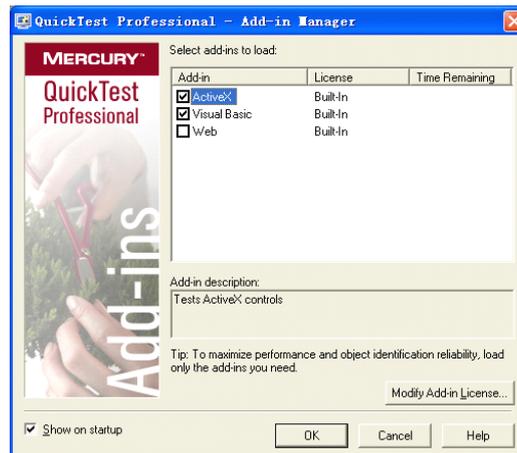


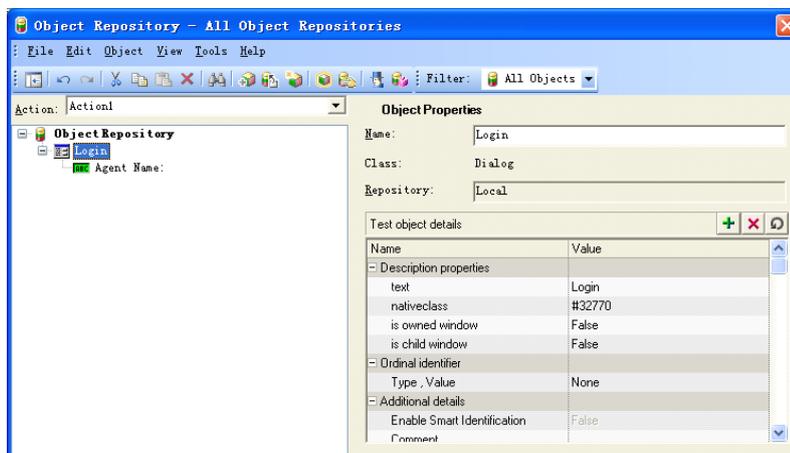
1.1 QTP 的具体讲解

1.1.1 QTP 的设置

- ◆ 启动 QTP 时，会弹出选择加载插件框，用插件的原则：用哪个插件，勾选加载哪个插件。不要都勾上，因为都勾上有时会冲突。如下图所示：



- ◆ Test Object（测试对象库）——用来存放录制解析出的标准对象，存放对象的容器，可以共享，可以维护
- ◆ Test Object Property（测试对象属性）——
 - Property 属性——脚本中用来标识某一个对象只用到部分属性，一般建议用不常变的属性来标识。
 - Method: 时间的方法
 - 测试对象属性：将解析出的对象属性放到测试对象库作为标准对象，提供回放时，测试识别的对象与对象库中的标准对象进行匹配（录制的对象库进行参照匹配）
 - 添加识别对象：直接在对象库中添加。也可以通过录制方式获取。



- 修改对象的名称：直接修改名称
- 可以添加或者修改对象属性
- Ordinal identifier（顺序识别符）——type、value 一般推荐 type 为 index,value 为 0.

1、2、3。如果选择 None，默认值均为 1、

针对很多对象有一样的属性，就要用 index 的顺序值来标识。例如一个界面上有若干个复选框。

其实这是很被动的方法。因索引的最大值会变化，根本无法做到每一行都去捕获，是无穷尽的。

在实际工作中，全部用程序方法去解析它

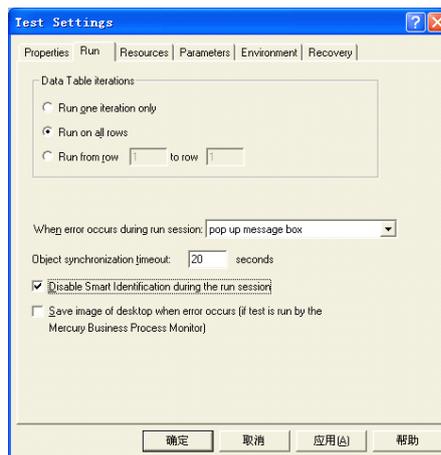
- 虚拟对象——tools——》 virtual object 中选择类型，点击 mark object，将虚拟对象框框起来，放到对象库对应的结构位置，然后重新录制才行，会出现 virtual Button（类型）。虚拟对象可以做操作用，不能做检查用。

- 识别对象原则：用固定的属性来描述识别一个对象，可以通过 **Highlight** 来进行验证，成功会在界面上闪烁（前提是打开界面），否则识别对象失败。

例如：用一个绝对坐标（与界面整个屏幕的左上角 00 为起点）的绝对值，随着窗口的拖动绝对坐标就会发生变化。相对坐标（是当前对象与当前窗口左上角的 00 为起点），界面稍微变化，相对坐标值也会变化，两者一般都不用。

- ◆ .Test Setting（测试集）——里面所有的设置均是默认的对整个脚本起作用。

点击 File——》 setting 菜单， **run** 界面如下：



- Disable Smart identification during the run session：对相似的智能识别失效。只要做 web 测试，建议都选上。选上后再抓对象。

- Data Table Iterations：针对 Globle 数据表的循环行数，控制整个脚本执行次数，建议选择第一项：

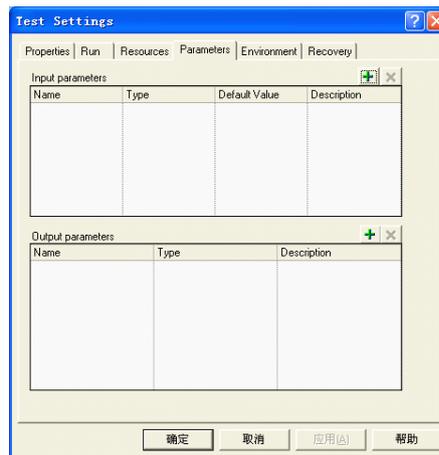
Run one interation only：仅仅执行数据表中一行

Run on all rows：执行数据表中所有行

Run from row to row：执行数据表中具体设置的行

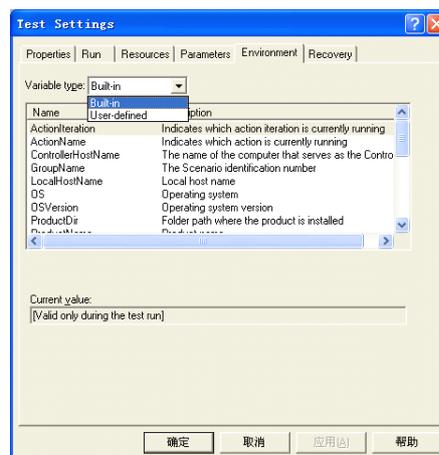
- When error occurs during run session：报告中当错误出现时，出现的错误提示类型。
- Object synchronization time out second：设置整个测试脚本的同步点等待时间。
- Save images。。。。：是否保存出错时的图片到桌面上。

Parameters 页面：



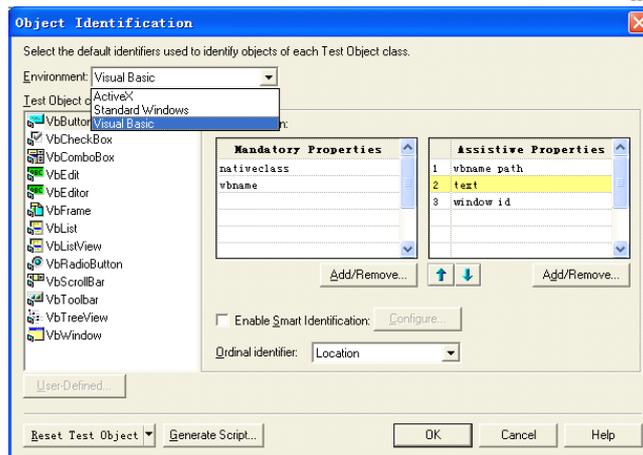
- Input parameters（输入参数）：从 QC 的用例中调用的参数，只有当执行整个脚本才能调用。
- Output parameters（输出参数）：将执行的结果，从 QTP 输出给 QC

Environment 页面：

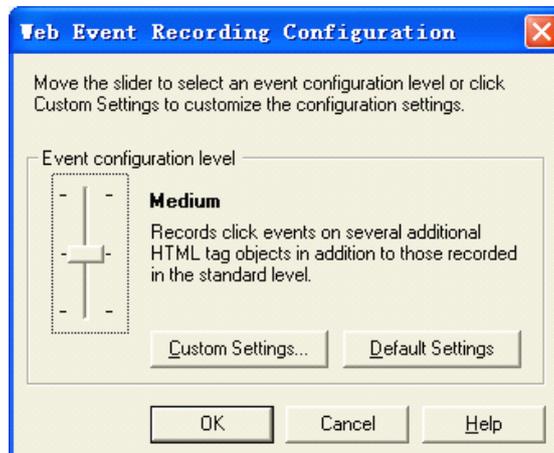


- Built-in: 内建环境，指 QTP 自身的环境变量，可以随意调用。
 - User-defined: 用户自定义的环境，只有当前的脚本才可以调用。
- 在系统开始——》运行中输入 conf，可以调出 netmeeting（郁闷，就这个功能不知道，导致我一天半看不到老师的笔记本界面，坐的很偏，投影仪看的太累了）
- System util Run.Environment.value（"FlightPath"）其中"FlightPath"设置成环境变量

◆ Object Identification: 点击 Tools——》Object Identification 菜单



- Environment 中显示的是当前加载的所有插件。选择不同的插件，下面会出现当前插件自带标准的 Test Object Class（测试对象类）。点击 User Defind 可以自定义。
- 选择具体的测试对象类，右面有 Mandatory Property（必须的属性）和 Assistive Property（辅助的属性）。当必须的属性不足以识别一个对象时，可以会加上辅助的属性来识别。
- Enable Smart Identification：是针对某个小的脚本来进行屏蔽相似匹配功能。
- ◆ Web Event Recording Configuration（web 事件录制配置）：点击 Tools——》Web Event Recording Configuration



- 在 Custom Web Event Recording Configuration 中：点击 Custom Setting 按钮。



- Custom Web Event Recording Configuration 窗口中存放对象需要录制的方法，可以对对象的方法进行维护。例如：Event——》add——具体对象方法
- Reset to 可以设置 High、Medium 和 Basic（最基本的标准对象库）
- ◆ Record and Run Settings（录制运行设置）：点击 Automation——》Record and Run Settings 被测试程序怎么启动
 - Web 页面

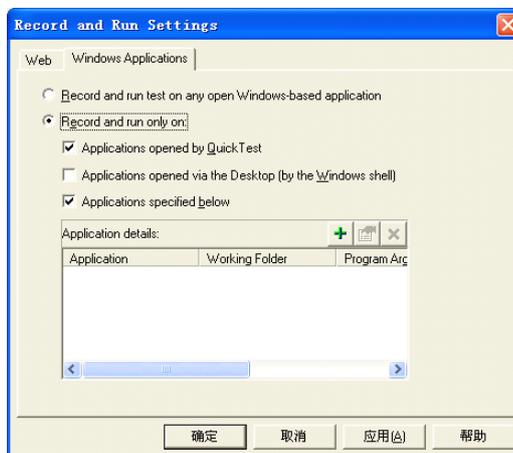


Record and Run Test on any open browser: 录制运行已经打开的浏览器。

Open the following address when a record of run session begins: 输入浏览器的地址。

且下面两个复选框中，第一个复选框是不要录制已经打开的浏览器，第二个复选框是当录制结束自动关闭浏览器。

- Windows Application 页面：应用程序



Record and Run Test on any open Windows_based application: 录制运行已经打开的应用程序。

Record and run only on: 增加应用程序的可执行文件路径，可以对三种选择方式进行录制。

- 总结：
 - 单选框第一项：手动启动，已经打开 WEB 或者应用程序进行录制。
 - 单选项第二行：自动启动，自动按照输入的地址启动一个 WEB 或者应用程序进行录制。
 - 最好的选择就是都选择第一个单选框。

启动应用程序的语法: System utl.Run 具体程序执行路径

如果录制 Windows, 要把 WEB 页面的录制选项选择第一项, 反之一样。

在录制过程中, 只要搞清楚逻辑, 找到对应的代码行, 可以从当前代码行之后的操作继续录制。

1.1.2 录制/回放/验证点

1.1.2.1 验证点原理

- ◆ 录制回放均是基于对象解析
- ◆ **验证点**: 将 GUI 对象属性与预期结值比较的过程, 且生成测试报告。**比较+报告**
- ◆ Insert ——》Checkpoint 只有在录制过程中才可生效。
- ◆ Checkpoint Properties 属性中, 如果对一个按钮插入一个验证点, 则属性中是 QTP 识别 button 的属性中能识别的属性, 在验证点属性中仅仅是 button 所有属性的子集, 仅仅是能识别的属性
- ◆ 在 Checkpoint Properties 窗口中, 对需要验证的属性前面的复选框打勾。
- ◆ Custant 里存放验证的预期值。
- ◆ 也可以选择 Parametre 进行参数化放到表中去。
- ◆ 语法: checkpoint timeout () seconds 定义匹配的超时时间。



- ◆ QTP 的测试报告
 - 最上面一级是脚本名称
 - 下级是 Run-time DataTab: 运行时使用的数据表
 - 点击报告不同的操作, 生成的保存快照设置: Tools——》Options——》Run——》save still image capture to result always
 - Test Iteration : 迭代次数
 - Action1 Summary: 动作的名称
 - 报告 xml 文件, 可以通过解析 xml 文件来得到高水平的 Testreport
 - 报告: Reporter.reportevent
- ◆ 检查点
 - 标准检查点: 可以检查标准属性的对象
 - 文本检查点: 检查可见外在的文本属性

- 文本区域：检查文本一大片区域
 - 位图检查点：检查图片（以像素绝对匹配为原则，都会影响结果。一般不建议用位图检查点）
 - 数据库检查点：取决于 Windows 数据源配置，可以利用自己书写的数据库检查点替代他，便于迁移。
 - Accessibility 检查点：
 - Xml 检查点
 - 在实际工作中，要弱化 QTP 自带的检查点
 - 自动化的预期值来自用例 Aut 被测应用程序 if a=b 报告 reporter.reportevent
- ◆ 同步点
- 对象 wait.property(精确)——必须有对象属性值发生变化
 - Wait s

1.1.2.2 函数

- ◆ GetRoProperty ——获得运行对象时的属性
- ◆ Set——Winedit 类型要输入值的函数
- ◆ Print——用来调试脚本
- ◆ Trim——空格函数，判断时使用，去空格
- ◆ Exitlase
- ◆ Cstrcl
- ◆ Csntcl
- ◆ Random Number——随机数
- ◆ Split——分割函数
- ◆ Cint——
- ◆ Array——数组
- ◆ Dim——定义
- ◆ Ubound——最大行数
- ◆ Left——左取字符串
- ◆ Right——右取字符串

1.1.2.3 20081101 总结

- ◆ 自动化脚本开发过程：
 - 捕获对象（结合用例涉及的对象）——自动化负责人维护对象库
对象库的对象做简单修改，不要重复，原则保持对象库简洁
对象库里的对象结构与应用程序结构一致，Highlight 成功，就 OK
 - 根据 case 写 code
编写测试步骤
插入验证点：
 - ✓ GetRoProperty

- ✓ SetToProperty
- ✓ Reporter
- 回放
 - 调试——print, msgbox 函数
 - 正式执行
 - 批量执行

1.1.3 20081102 数据驱动

- ◆ 数据驱动——QTP 参数化,包括下面五种:
 - DataTable
 - 随机数 RandomNumber.value
 - 环境变量
 - 参数
 - 其他数据驱动
- ◆ DataTable
 - GetRowCount——获取行数
 - SetNextRow——设置下一行
 - 向 DataTable 里写数据——DataTable("LoginResult".dtLocalSheet) = "pass"
 - 向 DataTable 里写列名——DataTable.LocalSheet.AddParameter"LoginResult"
 - 在 globe 表里若有 4 行,则程序将执行 4 次,建议不要用 globlesheet 做循环用。
 - DataTable 导出导入:
DataTable.Importsheet
- ◆ 其他数据驱动-老师用脚本例子来说明
 - Txt
 - Excel
 - Word
 - Xml
 - Db

Command Type = 1 (SQL 语句, 建议写 1), 2, 3, 4, 5 (存储过程)

Db 数据源: 建立文本, 改后缀名 udl, 双击进行自己数据源配置, 完成后, 选择记事本打开 (不要选择始终用记事本选项), 得到数据源连接串

1.1.4 其他:

- ◆ 参考与调试脚本, 可以用 QTP 自带的脚本编辑器——在 QTP—TOOLS——QuickTest Script Edit 菜单
- ◆ 调试: debug viewer F10
- ◆ 数组调试: for i = 0 to ubound arrayA
Print A[i]
Next
- ◆ Cmd 才会连数据源 adob

- ◆ 在录制过程中，点击模拟录制模式，然后将鼠标点击需要模拟录制的窗体上，点击开始模拟录制即可。如画图需要模拟录制。
- ◆ 网页基本语句：`browser () .Page().webedit("name").set"中国数字化档案"`
- ◆ `Type` 类型是键盘操作

花了整整 5 个小时的时间来回顾 QTP 两天课程笔记，当然脚本一点还没来看，只是将我当时粗略记录整理出来，我想肯定有很多同行也迫切想学习此知识，我也在艰难的学习中，因我不会编程，学习起来很郁闷。特将此提供出来，供同行们分享！希望我们共同进步！

欢迎交流：yuecaining@flyingsoft.cn