

实验二 功能测试实验（8 学时）

一、实验目的

- 1、了解自动化测试技术
- 2、熟悉捕获/回放功能测试工具的测试脚本开发流程
- 3、掌握黑盒测试设计技术
- 4、熟悉统一功能测试（UFT）工具的使用

二、基本知识

- 1、测试用例设计技术
- 2、UFT 工具的工作原理及使用

三、实验环境

- ① Windows 操作系统+IE 浏览器
- ② UFT 11.5 应用软件

四、实验内容

使用 UFT 进行测试的过程包括 6 个主要步骤：

● 准备录制

打开你要对其进行测试的应用程序，并检查 UFT 中的各项设置是否适合当前的要求。

● 进行录制

打开 UFT 的录制功能，按测试用例中的描述，操作被测试应用程序。

● 编辑测试脚本

通过加入检查点、参数化测试，以及添加分支、循环等控制语句，来增强测试脚本的功能，使将来的回归测试真正能够自动化。

● 调试脚本

调试脚本，检查脚本是否存在错误。

● 在回归测试中运行测试

在对应用程序的回归测试中，通过 UFT 回放对应用程序的操作，检验软件正确性，实现测试的自动化进行。

● 分析结果，报告问题

查看 UFT 记录的运行结果，记录问题，报告测试结果。

(一)工具的学习

1、录制测试脚本

以 UFT 自带的飞机订票系统 Flight 为例（见图 1）。先熟悉该系统（打开方式见下图），再录制测试脚本。操作过程为：登录系统、预订从 London 到 Denver 的机票、退出系统。

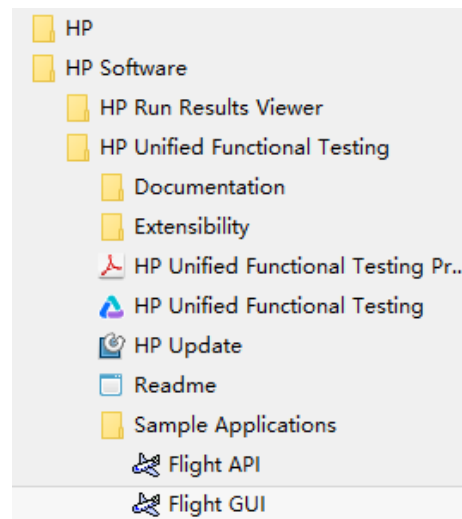


图 1 Flight 系统启动

1) 启动 UFT，选择合适的 Add-in (插件)。Add-in 选择是为了能成功识别对应 Add-in 的测试对象控件，也就是说与被测控件有关（见图 2）。

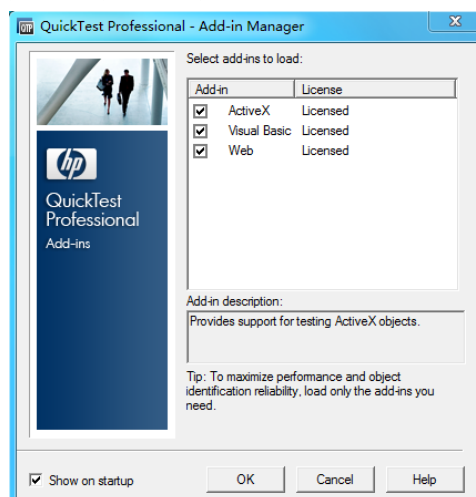


图 2 插件管理器

注意：若在执行 UFT 时未开启【Add-in Manager】


请点选【Tools】>【Options】，在【General】页勾选【Display Add-in Manager on startup】

2) 点击  按钮，如图 3 设置所测试的 Window 应用为 Flight。

3) 输入用户名密码登录。用户名是任意的四位以上字符，密码是 mercury。

4) 选择航班（从 London 到 Denver）

注意：日期格式是月/日/年，且年不能小于当前年，不能大于 2038。

- 5) 输入必要字段，点击 InsertOrder，退出系统。
- 6) 点击  停止录制，保存测试脚本为“Recording”。

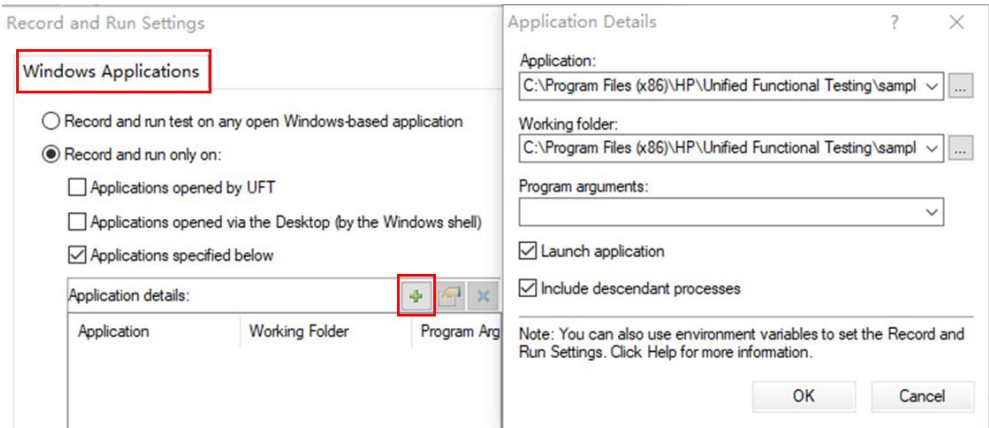


图 3 设置所测试的 Window 应用

2、查看测试脚本

- 1) 录制测试脚本时，UFT 将每个操作都录制下来，在 Keyword View 中以类似 Excel 工作表的方式显示，如图 4 所示。

Item	Operation	Value	Documentation
▼ Action1			
▼ Login			
Agent Name:	Set	"admin"	Enter "admin" in the "Agent Name:" edit box.
Password:	SetSecure	"5e58eb706787d1b0491fd27..."	Enter the encrypted password in the "Password:" edit box.
OK	Click		Click the "OK" button.
▼ Flight Reservation			
Menu	Select	"File;New Order"	Select item "File;New Order" from the "Menu" menu.
MaskedTextBox	Type	"121220"	Type "121220" in the "MaskedTextBox" ActiveX object.
Fly From:	Select	"London"	Select the "London" item from the "Fly From:" list.
Fly To:	Select	"Denver"	Select the "Denver" item from the "Fly To:" list.
FLIGHT	Click		Click the "FLIGHT" button.
▶ Flights Table			
Name:	Set	"Li"	Enter "Li" in the "Name:" edit box.
Insert Order	Click		Click the "Insert Order" button.
Menu	Select	"File;Exit"	Select item "File;Exit" from the "Menu" menu.

图 4 关键字视图

【Item】：操作步骤作用到的组件，包括测试对象（test object）、工具对象（utility object）、函数调用（function call）或脚本（statement）。

【Operation】：执行的动作，如 Click、Select。

【Value】：执行动作的参数值，例如当鼠标点选一张图片时是用左键还是右键。

【Documentation】：自动产生用来描述此操作步骤的文字说明。

可以点选 **【View】 > 【Expend All】** 查看测试脚本的每一个步骤。

- 2) 在 Expert View 中以源码的形式显示测试步骤，如图 5 所示。

```

1 Dialog("Login").WinEdit("Agent Name:").Set "admin"
2 Dialog("Login").WinEdit("Password:").SetSecure "5e58eb706787d1b0491fd27a4b8dbf05afe697a4"
3 Dialog("Login").WinButton("OK").Click
4
5 Window("Flight Reservation").WinMenu("Menu").Select "File;New Order"
6 Window("Flight Reservation").ActiveX("MaskedTextBox").Type "121220"
7 Window("Flight Reservation").WinComboBox("Fly From:").Select "London"
8 Window("Flight Reservation").WinComboBox("Fly To:").Select "Denver"
9 Window("Flight Reservation").WinButton("FLIGHT").Click
10 Window("Flight Reservation").Dialog("Flights Table").WinButton("OK").Click
11 Window("Flight Reservation").WinEdit("Name:").Set "Li"
12 Window("Flight Reservation").WinButton("Insert Order").Click
13
14 Window("Flight Reservation").WinMenu("Menu").Select "File;Exit"

```

图 5 专家视图

3、运行测试脚本

1) 测试脚本运行前设置

设置在测试结果中保存哪些图像信息的。

“Tools > Options” 菜单命令，选择 GUI Testing > Screen Capture，如下图所示：

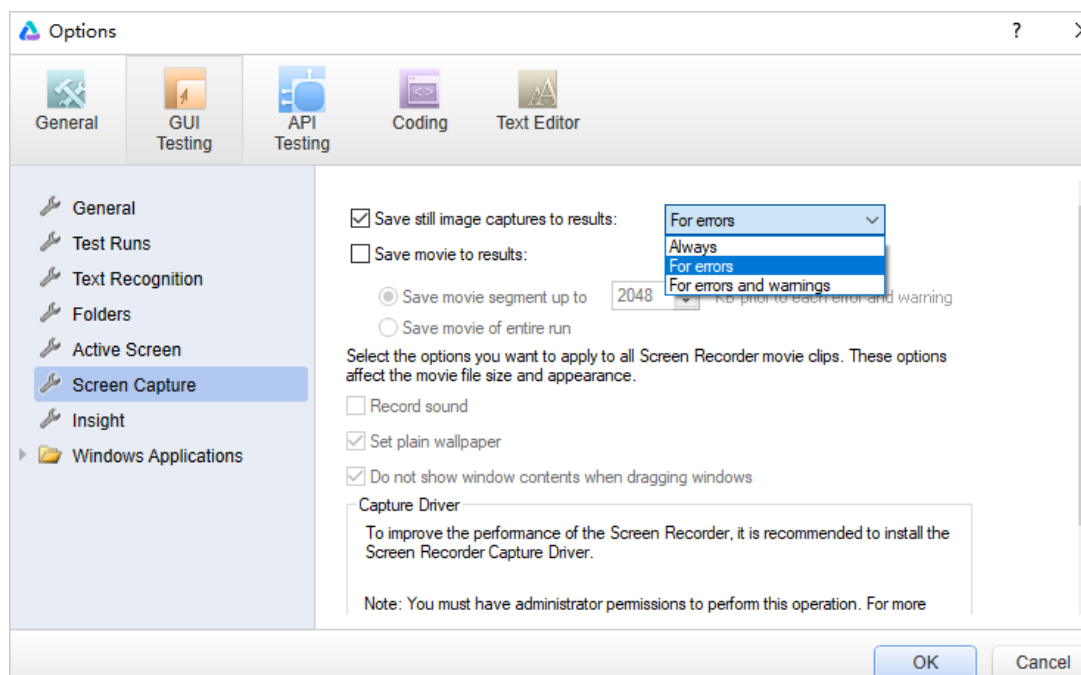


图 6 脚本运行设置

2) 运行脚本并查看运行结果

运行时选择结果位置，一般选择 “Temporary run results folder”。运行无错，如下图。

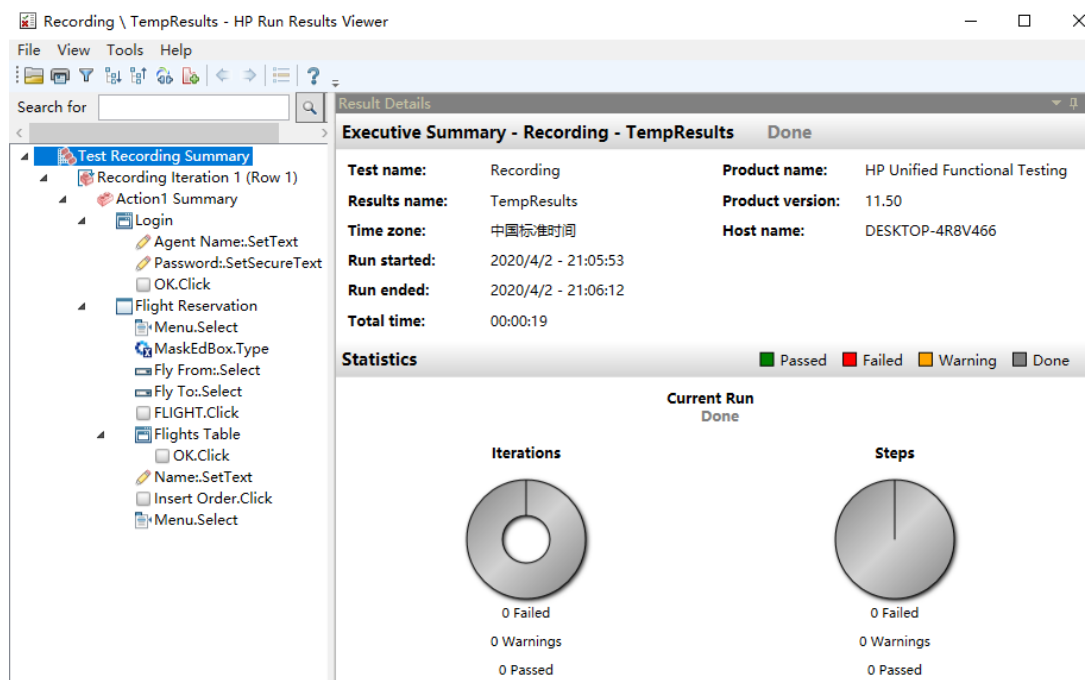


图 7 脚本运行结果无误

若将航班的 Fly From 改为“Londo”，存为“Recording01”。则显示结果出错，见下图。

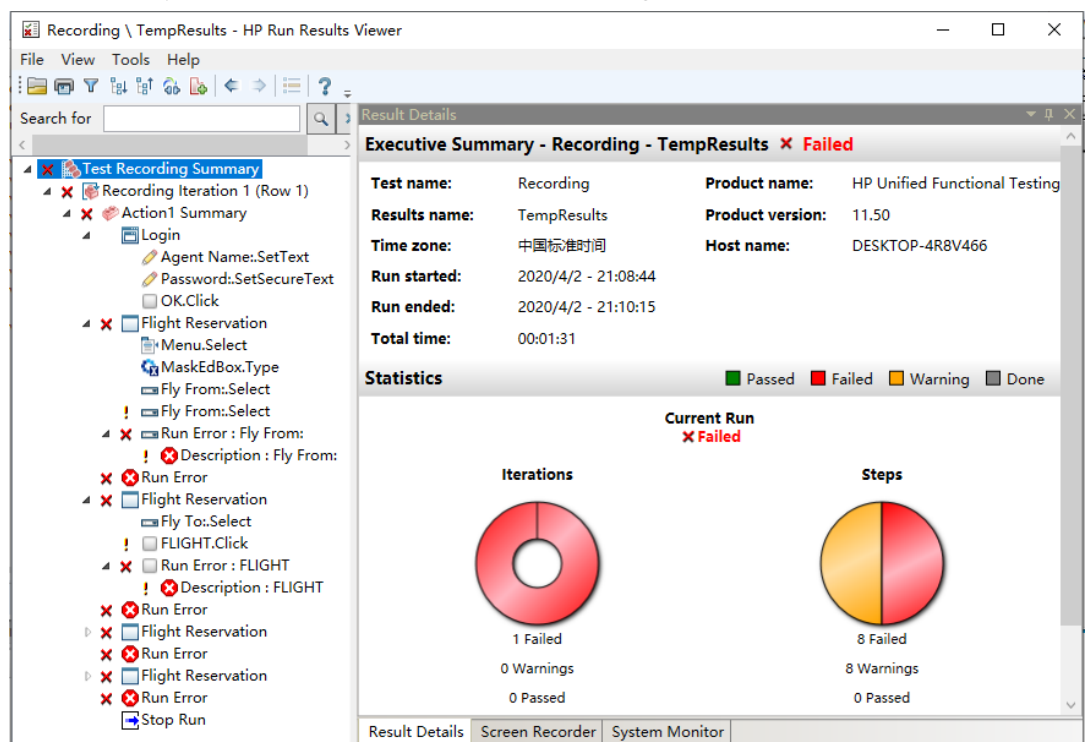


图 8 运行结果有错误

还可以展开左侧的树状视图，定位具体出错步。

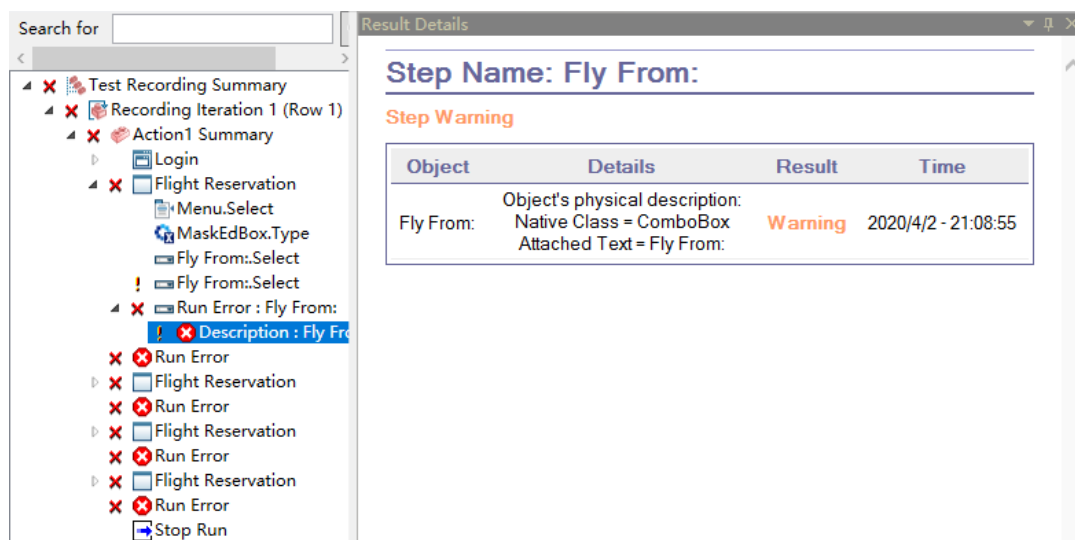


图 9 运行详细错误

4、创建检查点

建立检查点，以验证系统执行结果是正确的。UFT 可以设置如下类型的检查点：

检查点类型	描述	应用举例
Page 检查点	检查 Web 页面的属性。	检查 Web 页面中是否包含“无效”连接，或者检查连接一个 Web 页面所需要的时间。
Text / Text 区域检查点	检查在窗口或 Web 页面的指定位置是否显示了内容正确的文本。	检查在一个对话框中的指定位置，是否显示了内容正确的文本。 基于 Web 的应用不支持文本区域检查点
Standard 检查点	也叫做 Object 检查点，用于检查标准界面元素的状态。	检查一个编辑框的值是否正确，或者是检查一个复选框是否被选中。
Image 检查点	检查界面上图片的正确性。	检查 Web 页面上图片的正确性。
Table 检查点	检查表格中内容的正确性。	检查表格中的内容是否正确。
Database 检查点	检查 Windows 应用或 Web 应用访问数据库时，数据内容的正确性。	检查数据库查询的正确性。
XML 检查点	检查 XML 文档内容的正确性	XML 检查点有 2 种：XML 文件检查点和 XML 应用检查点。XML 文件检查点用于检查一个 XML 文件；XML 应用检查点用于检查一个 Web 页面的 XML 文档。
Bitmap 检查点	检查从 Web 页面或 Windows 窗口中捕获的某一部分区域。	检查一个 Windows 窗口（或这个窗口的某一部分）显示的正确性。

1) 打开 Recording，将其另存为 Checkpoint。

2) 插入标准检查点。例如：对 Agent Name 进行检查

--可先选择输入用户名步，再在菜单栏选 Design>Checkpoint >Insert Standard Checkpoint；也可直接在 Active Screen 中的 Agent Name 输入框中，右键选择 Insert Standard Checkpoint。

--接下来，在打开的检查点属性中选择需要检查的 Agent Name 属性并填入对应的检查值。

用了默认录制脚本时输入的“admin”，见图 10。运行脚本无误，见图 11。

--注意：插入检查点的位置应在输入用户名之后，因此，这里选择了“After current step”。

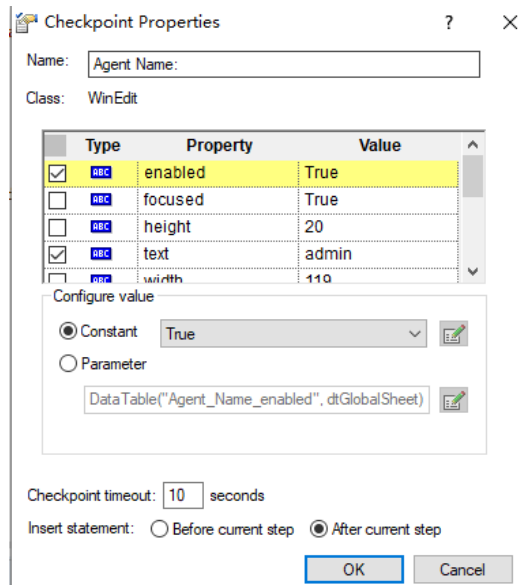


图 10 标准检查点属性

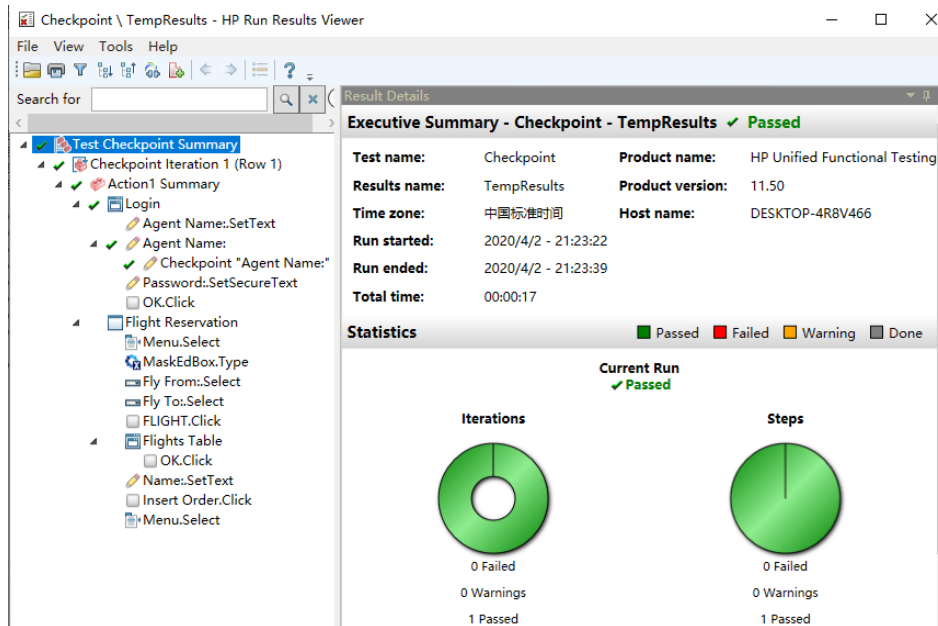


图 11 插入检查点运行结果无误

--这里，可将 text 属性的值 admin 修改，查看运行结果体会检查的含义。下图是将 text 属性值改为 username 后的运行结果。

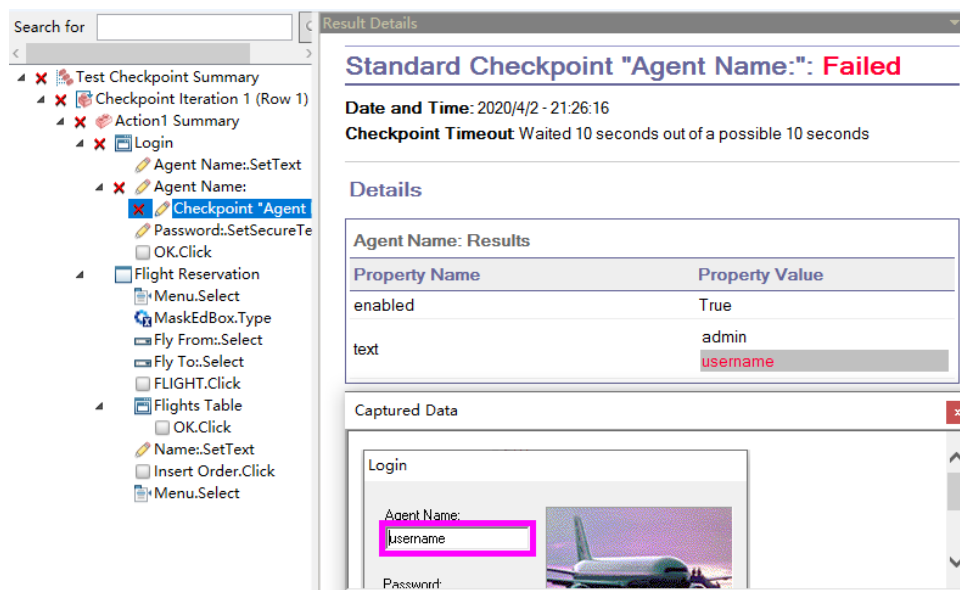


图 12 插入检查点运行结果出错

3) 插入文本检查点

在航班选择窗口中的“From->LON”处插入文本检查点，检查文字是否是“LON”，过程如下图 13。脚本运行结果无误，如下图 14。

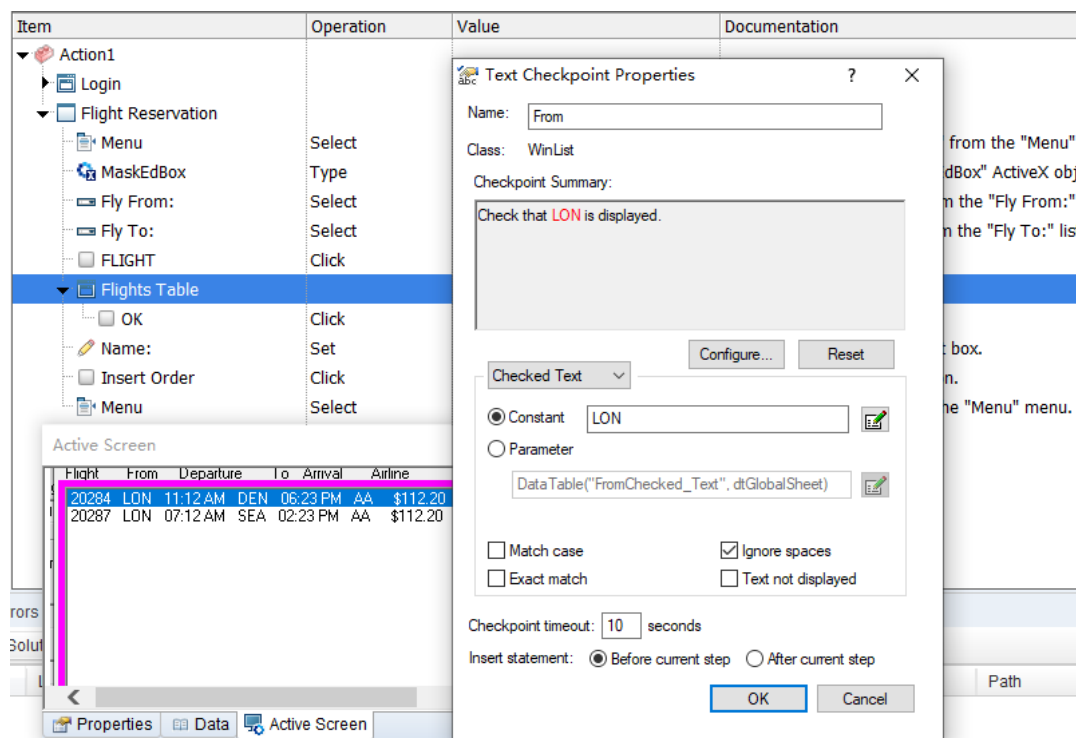


图 13 插入文本检查点

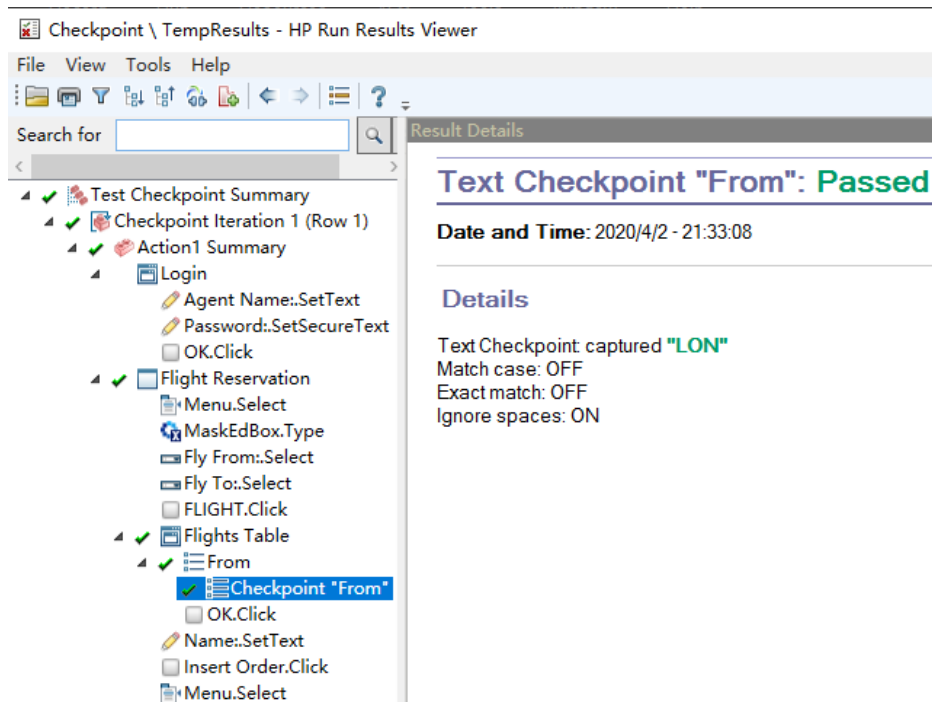


图 14 文本检查点运行结果

4) 插入数据库检查点—默认测试者知道当前系统的数据库情况

例如：Flight 系统用的是 Microsoft Access 数据库，购票操作即向 orders 表增加记录，数据库文件目录为“安装目录 HP\Unified Functional Testing\flight\app\ flight32.mdb”。现在对 orders 表的 Customer_Name 字段进行检查。即将数据库检查点插入 InsertOrder 操作之后。注意：先插入订单，再插入数据库检查点。具体操作如下：

--先在“控制面板>管理工具>数据源(ODBC)”里新建“order”数据源，过程如图 15。

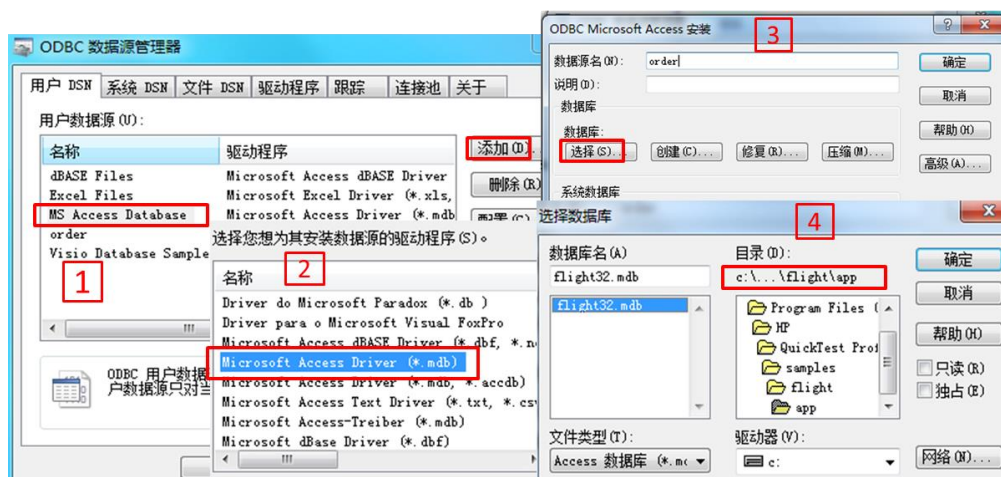


图 15 建立 order 数据源

--系统弹出"Database Query Wizard"对话框，选择"Specify SQL statement manu"单选框。完成后的状态如图 16 所示。

--创建连接字符串：单击"Create..."按钮，选择“order”数据源，即可自动生成连接语句

“DSN=order;DBQ=C:\PROGRAM FILES (X86)\HP\UNIFIED FUNCTIONAL TESTING\SAMPLES\flight\app\flight32.mdb;DriverId=281;FIL=MS Access;MaxBufferSize=2048;PageTimeout=5;”

--输入查询语句 select * from Orders where Customer_Name='admin';

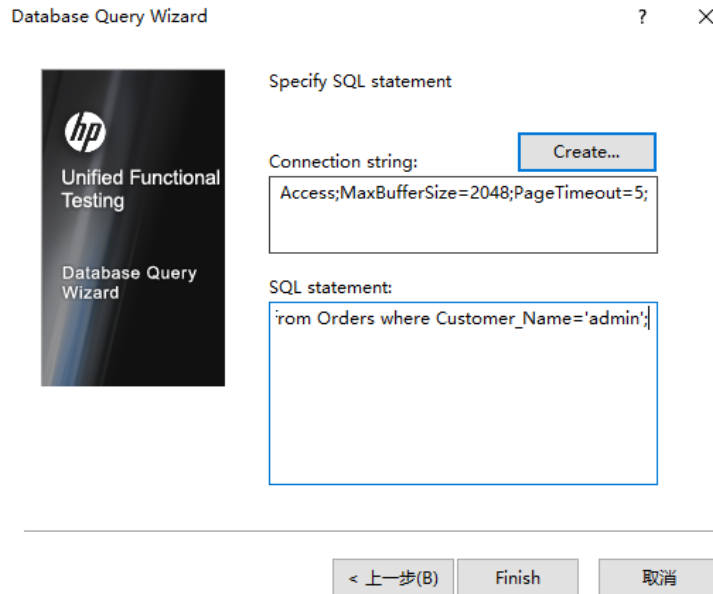


图 16 手工指定查询语句

--系统弹出"Database Checkpoint Properties"对话框，点选"After current step"单选框'，数据库检查点就插入完成了。

注意：对于 64 位的 win7 系统，无法创建连接.mdb 格式的 32 位的 Access 数据库连接
解决：找到 C:\WINDOWS\SysWOW64\odbcad32.exe 打开后,我们就可以加载基于 32 位的驱动了。

5、拆分 Action

Action 即动作，由一系列操作步组成，UFT 中的一个测试脚本可分成多个 Action。分为多个 Action，一方面可用来对测试步进行分组，便于脚本的阅读和维护；另一方面可以把需要重复使用的步录制成单独的 Action，供其它 Action 调用，简化脚本编写过程。

对一个 Action 中的某步进行参数化，当脚本执行时，可仅让该 Action 反复执行，而不是所有的步都反复执行。

Action 的种类：

- 不可复用 Action：只能被当前测试调用的 Action
- 可复用 Action：可以被其它测试多次调用的 Action
- 外部 Action：调用的其它测试中的 Action

1) 可以将登录、预定机票和按订单号查询、退出分成三个 action 进行录制，脚本存为 RecordingSplit。

2) 先通过 Add > call to new action 的方式添加新的 action。

可以对每个 action 建立共享对象库，过程为：Resource>Object Repository

Manager>Object>Navigation and Learn; 过滤识别的对象类型，识别后保存为.tsr 共享对象库；解决方案里 Action 右键选 Associated Repository with Action 选择保存的.tsr 文件。

也可以通过录制脚本的形式识别对象，不过自动识别的对象存储在本地局部对象库中，可通过对象库管理器合并两个局部对象库为新的共享对象库。

接下来可以通过录制的方式，也可以通过添加步骤的方式写测试脚本。

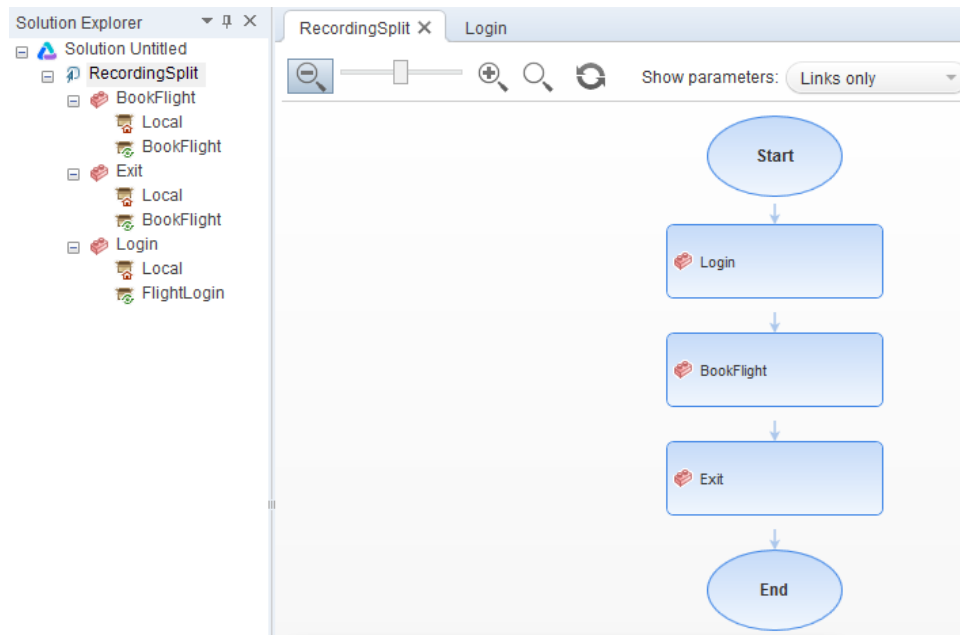


图 17 多个 Action

6、参数化

参数化即用参数替换录入的固定值，扩展基本测试范围的过程。参数化大大提高了测试或组件的功能和灵活性。QTP 有四种类型的参数：

- 测试、操作或组件参数
- 数据表参数
- 环境变量参数
- 随机数字参数

例如：将航班出发地和目的地进行参数化。

- 1) 打开测试脚本 Recording，将其另存为“Parameter”。
- 2) 选择要参数化的数值，这里是 Fly From 的 London 和 Fly To 的 Denver。

--先参数化 Fly From 的值，点选参数化图示 ，如图 18 进行设置。Parameter 选择【Data Table】，命名参数名为 fromPort，数据表作用域为当前 action。

--以同样的方式参数化 Fly To，命名"toPort"，并在数据表中输入其它值，参数化后的数据表如图 19。

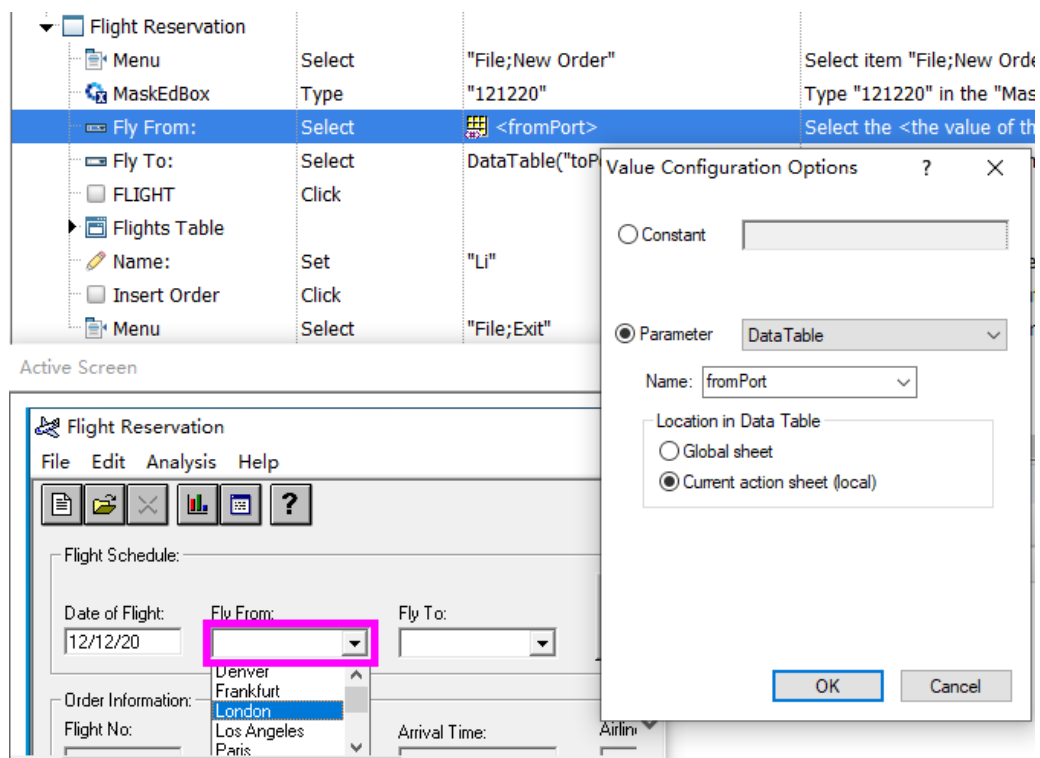


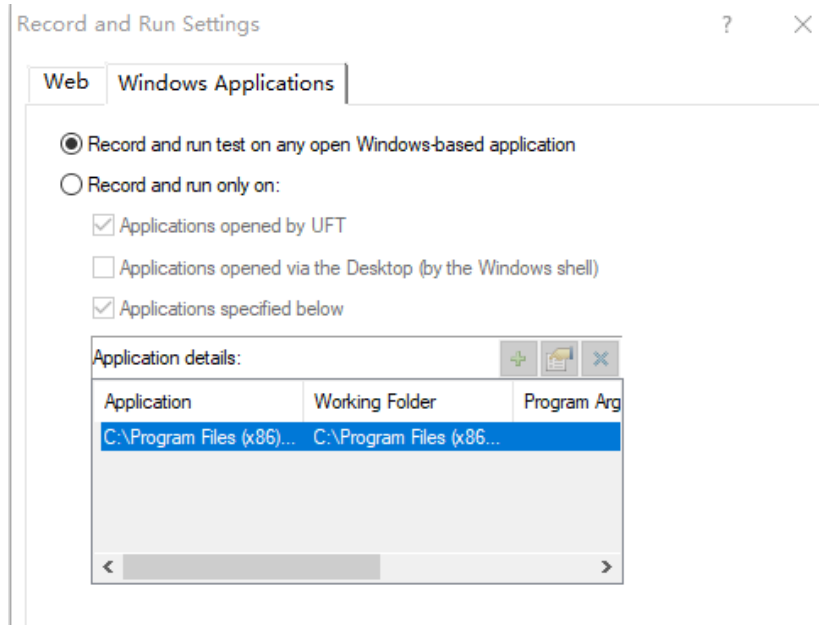
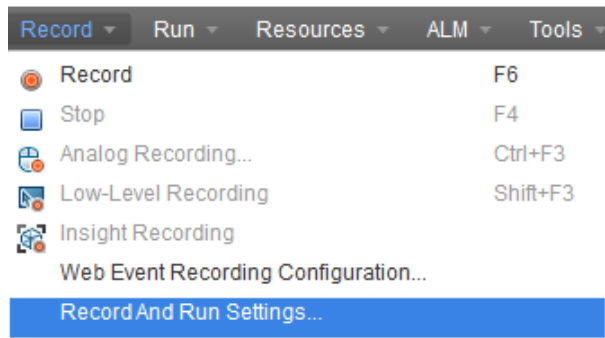
图 18 参数化设置

Fly From:	Select	DataTable("fromPort", dtLocalSheet)
Fly To:	Select	DataTable("toPort", dtLocalSheet)
FLIGHT	Click	

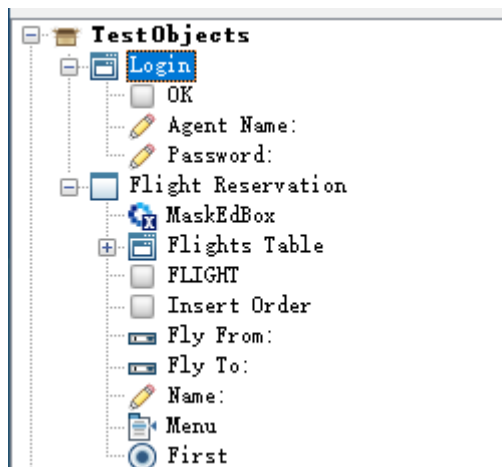
	fromPort	toPort	C	D	E
1	London	Denver			
2	Paris	Frankfurt			
3	Los Angeles	London			
4					

图 19 参数化后的数据表

注意：参数化航班，录制脚本时一定要先在菜单栏选择 File>New Order 因为再次预定航班时，若没有新建订单 Insert Order 按钮灰显的。若忘录此步，可以单独进行录制。单独录制需要对录制设置进行修改，Record > Record And Run Settings 中的 Windows Applications 选项卡，选择第一项 Record and run test on any open Windows-based application，如下图。同时 Flight 应用启动并进入登录后的界面。



确保对象库里面的对象都存在，如下图。



仅在专家视图中添加语句：

Window("Flight Reservation").WinMenu("Menu").Select "File;New Order", 则对象库里没有 Menu 对象，运行会提示有错误。

3) 保存脚本，运行脚本前需设置 Action 调用次数。Action1 右键设置 Action Call Properties 为 “Run on all rows”，见图 20。同时，检查脚本运行设置，如图 21，File>Settings>Run 选

择 “Run on all rows”。这样设置脚本就会执行 3 次，还需要修改脚本，脚本开始加上自动启动 flight 应用程序的语句：“Systemutil.run("C:\Program Files (x86)\HP\Unified Functional Testing\samples\flight\app\flight4b.exe)”，同时 record and run settings 不勾选 Applications specified below。

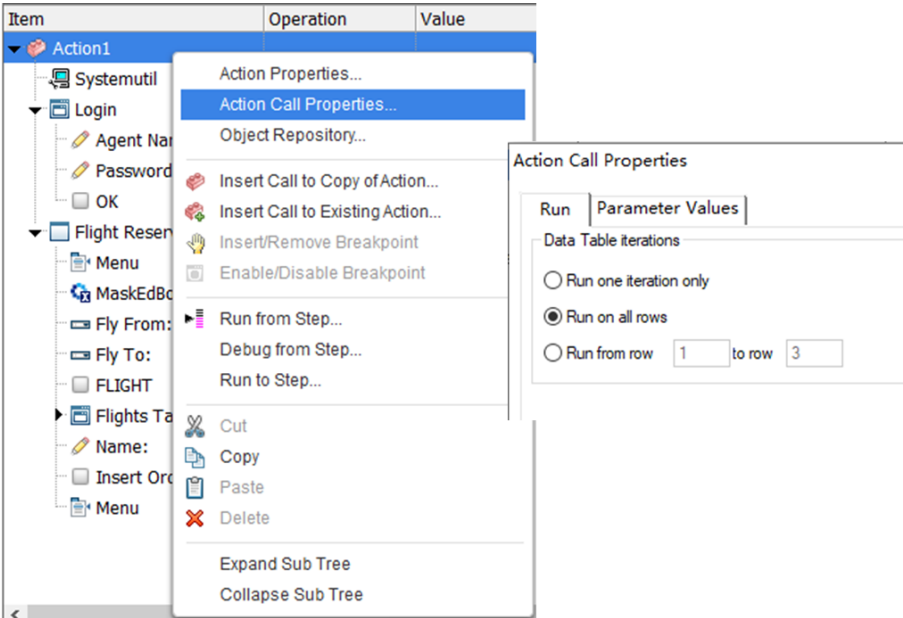


图 20 Action 调用设置

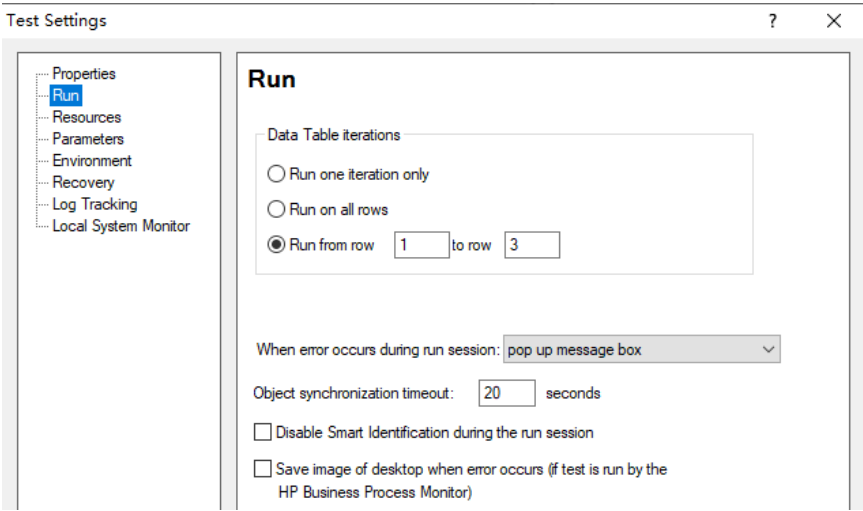
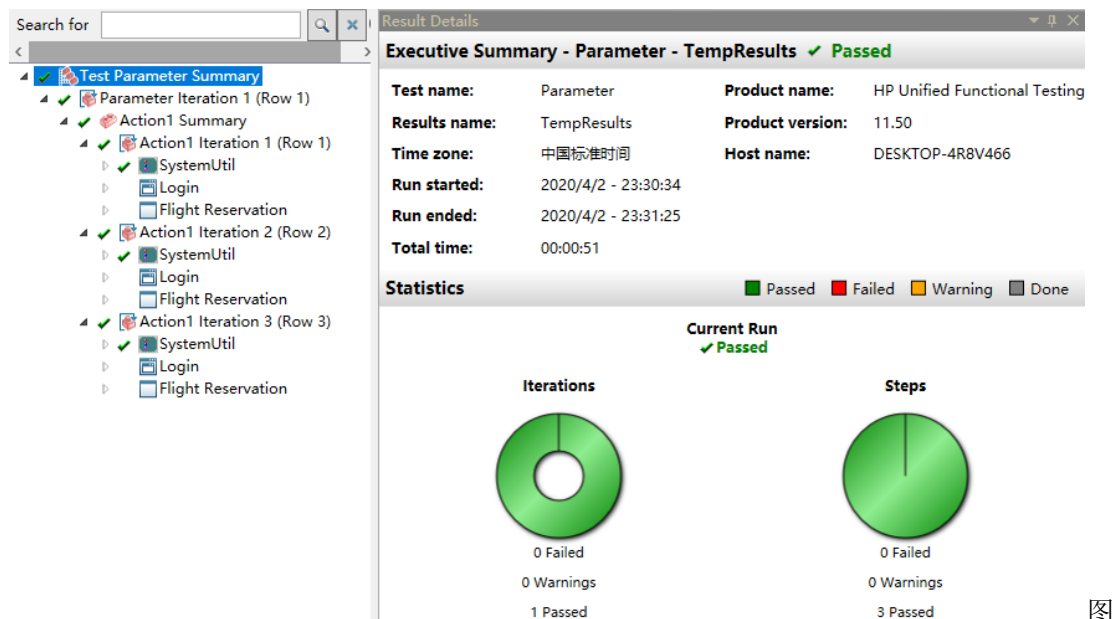


图 21 设置运行迭代次数

运行脚本无误，如图 22。



22 参数化脚本运行结果

参数化部分代码参考如下：

```
Systemutil.run("C:\Program Files (x86)\HP\Unified Functional Testing\samples\flight\app\flight4b.exe")

Dialog("Login").WinEdit("Agent Name:").Set "admin"
Dialog("Login").WinEdit("Password:").SetSecure "5e58eb706787d1b0491fd27a4b8dbf05afe697a4"
Dialog("Login").WinButton("OK").Click

Window("Flight Reservation").WinMenu("Menu").Select "File;New Order"
Window("Flight Reservation").ActiveX("MaskedTextBox").Type "121220"
Window("Flight Reservation").WinComboBox("Fly From:").Select DataTable("fromPort", dtLocalSheet)
Window("Flight Reservation").WinComboBox("Fly To:").Select DataTable("toPort", dtLocalSheet)
Window("Flight Reservation").WinButton("FLIGHT").Click
Window("Flight Reservation").Dialog("Flights Table").WinButton("OK").Click
Window("Flight Reservation").WinEdit("Name:").Set "Li"
Window("Flight Reservation").WinButton("Insert Order").Click

Window("Flight Reservation").WinMenu("Menu").Select "File;Exit"
```

7、建立输出值

参数化可以让测试脚本使用不同的测试数据。UFT 也可以从应用程序输出数据到 DataTable，进而被其它测试脚本调用。

本节在实现在 RecordingSplit 的基础上，对插入的订单编号 Order No 设成输出参数。然后按订单号查询的时候，订单号来自于插入订单的订单号，而非固定取值。

步骤：1）打开测试脚本 RecordingSplit，将其另存为 OutPutOrderNo。

2）在 BookFlight 中设置输出值。在 Order No: 文本框处点击右键选择 Insert Output Value，将文本框 Order No: 的 text 属性值参数化为全局变量“Order_NO_text_out”。注意该输出值应添加在 Insert Order 步骤之后。

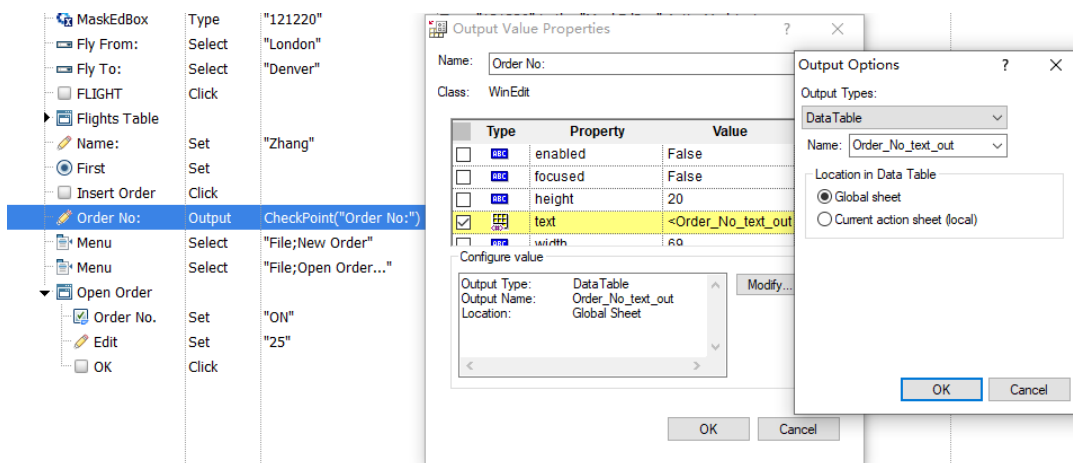


图 23 OrderNo 输出值设置

3) 将按订单号查询的订单号参数化为输出参数 Order_NO_text_out。

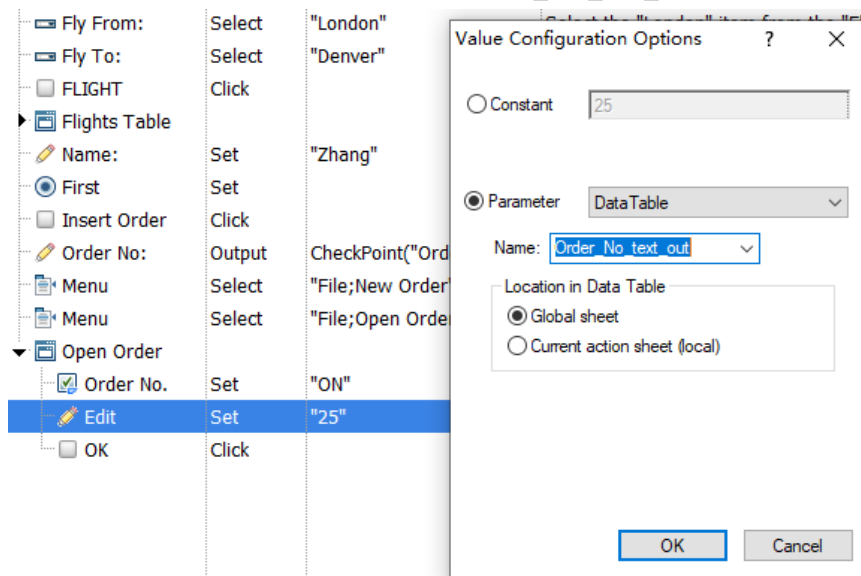


图 24 查询订单号参数化为输出参数

5) 运行脚本，输出参数执行情况如图 25，整个脚本执行成功。

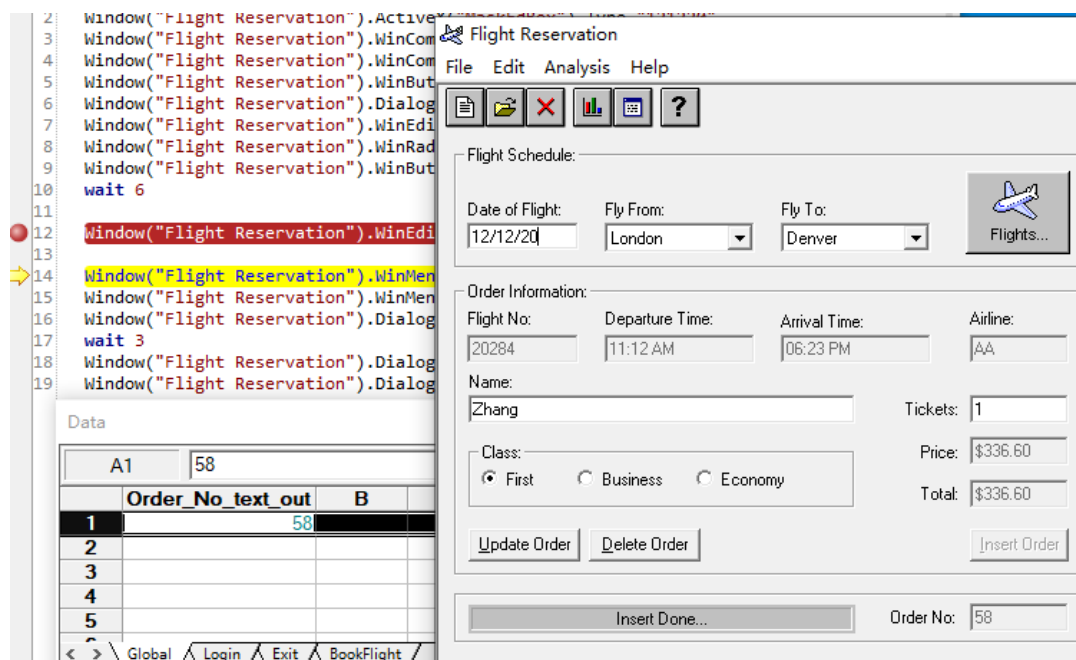


图 25 运行中的输出参数

6) 保存最终的 OutPutOrderNo 脚本，其中 BookFlight 脚本为:

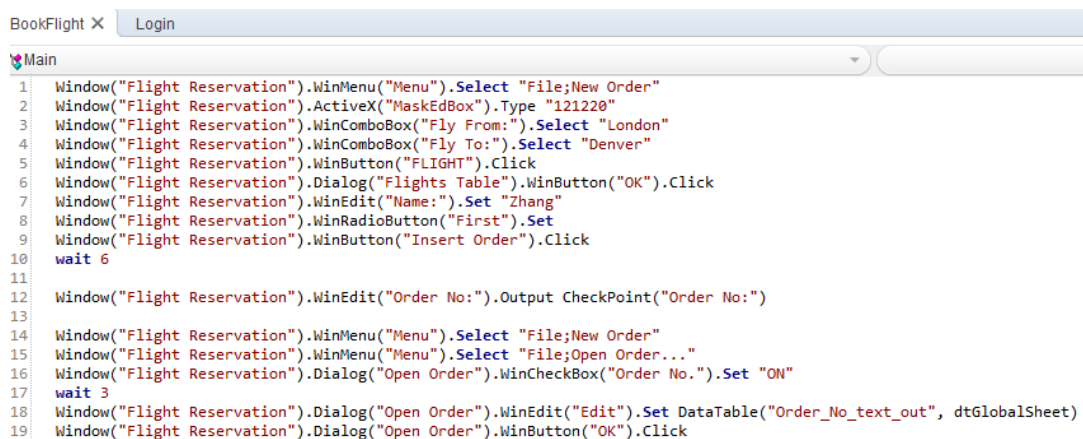


图 26 BookFlight 脚本内容

(二)按要求完成测试脚本的录制和编辑增强

- 1、脚本录制，将脚本分成 3 个部分，分别实现登录、机票预订和查询、退出。
- 2、将预订机票的出发地和目的地参数化。
- 3、检查机票预定时填入的 Name 的文本值。
- 4、插入订单后，再按订单号查询订单。订单号设为输出参数，然后在 File>OpenOrder 页面选择按 OrderNo 查询，填入的 OrderNo 参数化为上步设定的输出参数。
- 5、运行脚本无误后保存为 OutputOrder。

五、实验要求

- 1、按指导书步骤完成实验内容的第一部分工具的学习，保存最终的 Output 脚本。学习过程中记录错误信息并给出排错方法。

2、独立完成实验内容的第二部分，保存最终的 `OutputOrder` 脚本。遇到问题，请仔细回看第一部分，记录错误信息并给出排错方法。

3、写出实验报告，报告包括以下几项：

- ① 实验目的。
- ② 实验内容。着重记录独立完成的脚本实现步骤，记录出错信息并给出排错方法。
- ④ 实验结果。给出脚本运行结果截图
- ⑤ 实验小结。总结整个实验，并给出实验心得。