目 录

自动化测试数据之可不可复用

作者：刘毅

摘要：我认为测试数据不仅是自动化测试的核心要素，也是所有类型的测试所共有的一个基本元素，离开测试数据的测试基本上就丧失了它的可靠性与意义。当然，对于自动化测试来说，它对测试数据的使用和管理又有着它自己的特殊要求：可反复使用、能够自动生成或更新，可轻松迁移……可能有些人对一个系统拥有三、四套乃至更多的测试环境，同时存在多个版本的并行测试没有概念，那么这些人在自动化测试的其他方面要有着更高的追求，但是对于测试数据来说，却从不曾为之忧心和烦恼。

关键字：自动化测试；测试数据；回归测试；数据复用；存储过程

需求背景

我们的应用是基于 Weblogic 和 Oracle 的 B/S 结构保险业务系统，应该属于标准的 J2EE 架构。自动化测试力量现阶段集中在 UI 层验证测试，主要应用于系统测试中的冒烟和回归测试。常规版本周期从半个月到一个月不等，紧急需求或生产缺陷引发的非常规版本测试周期从一个工作日到一个月不等——想必比起互联网应用的发布速度，我们算是比较宽松的。虽然比较宽松，但是由于系统耦合度极高，业务逻辑极其复杂且不具有通用的规则引擎，数据种类千变万化，系统内外关联影响很大，我们任何一个版本的任意一个小小的改动点都有可能引发生产 L-1 级别的程序缺陷乃至重大故障。考虑到关联影响的分析太过依赖个人能力，不具备高度的可复制性和易推广性，所以我们对回归测试和的要求很高，也是因为对回归测试的要求很高，我们对冒烟测试的要求也很高，并且要求能够快速完成。

逆向思考

最初我们纠结于可用的数据如何从数据库抓出来，抓出来之后是直接使用还是先做本地存储然后定期更新。后来我们发现无论如何操作，从一个 4TB 数据库中抓取可用的数据似乎又是一个漫长的过程，尤其是部分同事对 SQL 效率优化没有太多概念的时候，测试脚本常常因此运行太长时间而被自动化测试管理平台的超时控制机制强行终止。即便是获取数据的速度不慢，而有些特殊业务的数

据量也不能支持长久的测试运行和测试环境的切换，依赖上游数据的产生根本不具备及时性和稳定性。

似乎我们大部分人反对使用固定数据进行测试，因为这个观念从自动化测试入门开始就被灌输到我们的脑子里了，绝大多数接触过自动化测试的人都知道测试脚本需要参数化，并且实现测试数据和脚本操作的分离。参数化和数据分离是正确的，只不过数据分离并不代表测试数据不可以固定使用，如果一条数据可以进行千万次的对系统的功能验证，只要它是有效的，我们是否非要去不停地更新掉这条数据呢？我觉得未必！其实我们所说的固定数据更多的是指，在一个功能点中存在多种数据处理方式，也就是存在多个代码分支的情况下，只使用一条或几条相同的数据去覆盖其中的一个分支。那么如果我们能够去分析一下这些代码分支，针对不同的逻辑分支去构造数据进行测试，那么每一个分支上的测试数据即便是固定的，测试遗漏也不会比我们的正交覆盖更多。所以，只要我们努力理清代码的逻辑分支，并且实现测试数据的使用完回退，那么测试数据的固化对于自动化回归测试来说并没有什么问题。

因此我觉得，只要参照被测功能的代码，在数据库层面书写被测功能的逆向数据状态转换过程，在测试运行之前进行调用，那么每一次测试执行都可以使用同一组数据去进行。而且，这种方法的一个最大优点是，我们可以很轻易地在代码中找到每一个终止条件，我们就可以编写对应的判断，让我们的数据依据被测程序代码逻辑来构造。而后在本地的数据源或者文本中存储这些数据，运行时引用并且有针对性（按照指定的数据修改比全库查询更加快速）的进行修改，使其满足运行的需要，运行过程中出现的任何异常都可以交给每天的定时任务去清理，完成对数据场景的恢复和还原。那么这种解决方案到底如何实施并且对我们的测试能起多大作用，我们下面通过一个简单的养老险保全项目来看一下。

应用实例

我们需要进行分红险种的红利领取操作，那么首先我们选择的保单必须是分红险保单，并且保单、所要操作的分单状态正常、分单对应的红利账户状态正常、账户余额大于零等，这些将都是被测程序所要检查的关键点。更复杂的操作如团险保单拆分等，分支虽多，但其每一个操作分支皆与此类同。

- 首先，我们要确保我们的数据中不可能被其他任何程序改动的字段要符合我们的程序逻辑控制，例如：保单的险种必须是分红险种，这一点是系统内任

何程序的运行都无法改变的，所以我们无需担心这些字段被改动而导致自动化测试的时候程序无法正确给出响应。既然像这样的基础的、不可变的信息都是不会改变的，那么我们的逆向数据修改过程中自然无需对其兼容，所以我们必须首先查询或构造出满足这些基础条件的数据，再对其余字段信息进行修改，以缩减我们的代码维护量。

- 依据我们的基础业务知识，构造出任意不满足程序逻辑的提示信息，依据这些提示信息关键字，从我们被测应用的代码库中找到程序所引用的 JAVA 代码方法和数据库 SQL 代码，并且走读代码、理清代码分支，弄清楚本操作最后所修改的是什么表的什么字段。例如我们所说的红利领取，它的工作流程与所有其他保全项目一致，它自身保全确认之后将修改（减少）账户主档表中的红利账户余额，并且产生财务应付记录。那么如果我们判断出红利账户余额不足的时候可以通过我们自己编写的过程去无限放大它，使其满足我们的测试执行需求。同此逻辑，逐个找出所有让程序退出的数据状态点，集合所有数据构造或回滚操作，就变成了我们所要编写的逆向数据修改过程了，数据库代码样例如下表。

红利领取保全项目数据生成与修改过程（鉴于篇幅有限，下面省略代码若干）

```
procedure data_for_903000(p_sub_polno in varchar2) is
  --该保全项目控制关注分单、单状态是否正常，保单、分单是否已暂停，分单对应的红利账户状态是否正常，红利账户余额是否大于0
  cursor c_account_info is
    select * from account_info where sub_polno = p_sub_polno and account_category = '2';
  v_account_info   c_account_info%rowtype;
begin
  --将所有不符合本项目操作的表字段进行修改
  open c_account_info;
  fetch c_account_info
    into v_account_info;
  close c_account_info;
  if v_account_info.account_balance < 100 then
    update account_info Set account_balance = 99999999.00 where sub_polno = p_sub_polno and
account_category = '2';
    commit;
  end if;
  --将所有与当前保单所要发起的保全申请互斥的已有申请撤销
  cancel_alltask_mutexing(v_certinfo.policy_no, '903000');
end;
```

➢ 对于同一保单下操作的互斥性，我们使用通用的过程去处理，其代码可参见下表。而对于任何异常引发的运行失败或中断，考虑到我们可能需要及时重新运行，那么我们也可以把异常数据场景的恢复纳入这么一个过程中，把所有已经写入临时表但是尚未保全确认的数据全部清理掉。作为所有保全项目的公用代码，下面这个“按申请号撤销指定的保全申请”这个过程代码样例虽然简单，但实际上包含数据的修改删除是非常多的，这里不再赘述。

撤销当前保单下所有与将要发起的任务互斥的申请（鉴于篇幅有限，下面省略代码若干）

```
procedure cancel_alltask_mutexing(p_policy_no        in varchar2,
                                  p_endo_type in varchar2) is
  cursor c_get_mutex is
    select distinct apply_no
     from pos_base_info
    where policy_no= p_ policy_no
      and pos_item_type in
        (select distinct mutex_pos_item_code
          from pos_mutex_setting
         where current_pos_item_code = p_endo_type)
          and pos_status not in ('0', '2', '8');
    --查询出所有与当前保单所要发起申请互斥的未完成申请任务
    v_apply_no pos_base_info.apply_no%type;
  begin
    open c_get_mutex;
    loop
      fetch c_get_mutex
        into v_apply_no;
      exit when c_get_mutex%Notfound;
    cancel_appointed_task(v_apply_no);
    commit;
    end loop;
    close c_get_mutex;
  end;
```

按申请号撤销指定的保全申请（鉴于篇幅有限，下面省略代码若干）

```
procedure cancel_appointed_task(p_apply_no in varchar2) is
  --修改保全申请状态和工作流状态控制表
  update workflow_data_table set current_step = '90', current_processor = null where apply_no = p_apply_no;
  --作为一个通用的过程，这里需要删除和修改数十张临时表
  delete from processing_xxx_temp where apply_no = p_apply_no;
  delete from processing_xxx_detail where apply_no = p_apply_no;
  --修改保全申请互斥控制数据
  update task_mutex_info Set isdel = 'Y' where apply_no = p_apply_no;
```

```
    --删除已经生成的财务数据关联
    delete from finance_collection where apply_no = p_apply_no;
end;
```

- 同样考虑会存在异常而引发运行失败或中断，我们还可以建立每天定时数据清理的 JOB，按照自动化测试运行的频率和最晚结束时间来设置 JOB 运行的开始时间和频率，让工作时间段内的手工测试执行不受自动化测试带来的数据损坏的影响。这里需要指出的是，JOB 运行是自动的，在没有人工干预的情况下，它所处理的数据范围如何限定是一个很大的问题。当然，我们可以专门为自动化测试分配独占的操作用户，为已经涉及到的数据上打上特殊的标记，这样就比较便于我们识别到底哪些数据是需要我们的定时任务去清理的。如下面的 cancel_daily_fail_task 过程中 cursor c_get_all_item 所定义的数据查找逻辑也就是我平时所用的保全申请影像条形码的组成规则。

建立每天上午 10 点钟的定时清理 JOB

```
begin
  sys.dbms_job.submit(job => :job,
                      what => 'automation_testdata_gbs.cancel_daily_fail_task;',
                      next_date => to_date('19-09-2011 10:00:00', 'dd-mm-yyyy hh24:mi:ss'),
                      interval => 'trunc(sysdate +1) + 08/24');
  commit;
end;
```

查询所有未完成的自动化测试申请（鉴于篇幅有限，下面省略代码若干）

```
procedure cancel_daily_fail_task is
    cursor c_get_all_item is
      select pos_item_type_code,
             decode(substr(pos_item_type_code, 1, 2),'00',substr(pos_item_type_code, 3,
4),substr(pos_item_type_code, 1, 4))
        from pos_item_type_table;
    v_ all_item c_get_all_item.%rowtype;
  begin
    open c_get_all_item;
    loop
      fetch c_get_all_item
        into v_ all_item;
      exit when c_get_all_item%notfound;
      ......
      cancel_appointed_task(v_apply_no);
    end loop;
    close c_get_all_item;
```

```
end;
```

- 最后，那就是我们在自动化测试流程运行开始的时候去调用这些我们定义好的过程了，以 QTP 为例，使用 VBS 调用存储过程的代码如下。我们的每个不同保全项目的初审发起操作中可以去调用这些过程，在运行之初构造好我们所需要的满足自动化运行的数据，让我们的自动化测试运行得更加自动化一些。

调用数据库存储过程的 VBS 函数

```
Function RunProcedure(procName,dbUserName,dbHostAddress,dbServerPort,dbSid,dbPassWord)
        Set DBCNN = CreateObject("ADODB.Connection")
        Set DBCMD = CreateObject("ADODB.Command")
        DBCNN.Open = "Provider=""MSDAORA.Oracle"";User ID="&_
                        dbUserName&";Data Source=""(description =(address = (protocol = tcp)(host = "&_
                        dbHostAddress&")(port = "& dbServerPort&"))(connect_data =(sid = "&_
                        dbSid&")))"";Password="& dbPassWord&""
        DBCMD.ActiveConnection = DBCNN
        DBCMD.CommandType = 4
        DBCMD.CommandText = procName
        DBCMD.Execute
        DBCNN.Close
        Set DBCMD = Nothing
        Set DBCNN =Nothing
End Function
```

```
'调用举例：
subPolNo = DataTable("分单号","保全初审")
RunProcedure "automation_testdata_gbs.data_for_903000(''' & subPolNo &
''')", "tUser", "192.168.1.135", "1521", "testAnn", "XXX"
```

总结

我知道用我这种这种方法去准备自动化测试数据在自动化开发上是非常耗功夫的，但是我反复对自己说：如果你不想在今后的测试运行中挖东补西而疲于奔命的话，还是早点把这活给做了吧，有句话怎么说来着：早死早超生！我说服我自己照着自己的想法去做了，效果还不错。

没有谁会比别人更见多识广或者更权威——在自动化测试数据使用和管理方面，我觉得可能有部分人所面临的情形与我现在所面临的状况相同，但是每个人所使用的解决办法可能也不尽相同。个中美或不美，一如我见青山、青山见我，难以尽道；周末无事，闲敲键盘，以慰时光而已。

我的 GUI 自动化测试框架发展历程

作者：文青山

摘要：本文总结了我在学习和使用 QTP 做功能自动化测试过程中的经验，着重介绍了以前搭建的一些不成功的基于 QTP 的框架，分析了这些框架为什么失败，这些框架有哪些地方是可取的，哪些地方是不可取的，理想中的框架应该是什么样子等内容。在最后我还介绍了一个基于类的思想来构建的 QTP 框架 qwenTest，此框架是在这些失败的框架的基础上一步步地改进而成的，并且此框架还不断地在改进和完善中。本文粗略的介绍，希望能给各位正在学习和使用 QTP 的朋友们带来帮助。另外，本文更像是一个框架又一个框架的总结，所以有很多不妥之处还望各位朋友谅解。

关键词：QTP；框架

从接触自动化测试到现在快二年了，这二年由新手变成 old 手，由 QTP 到 WatiN 再到 White 的了解，经历过很多，也走过很多的弯路。下面这篇总结非常长，不知您有耐心看完并理解我犯过的错误不，但本着与君共析错误的原则，请原谅我是如此的啰嗦。

下面的内容主要总结了这二年来构建的几个 QTP 框架，并着重介绍了其中的二个失败的框架，这二个框架是我学习时构建的，然后再简单地介绍了现在的测试框架 qwenTest。希望通过文章中的介绍，借助 51Testing 的平台可以与君相识相知相析自动化测试。废话就不多说，直接上内容了。

第一部分：第一个失败的框架（供冒烟测试用的）：

文档目录的组织方式：

模块：

模块名 1 ：在此目录中又划分为：

- 业务函数
- 截图日志
- 描述对象库
- 脚本内容

模块名 2

模块名 3

模块名 4

所谓的模块名为脚本文件，一个模块一个脚本文件。

Case：使用的是冒烟测试用例

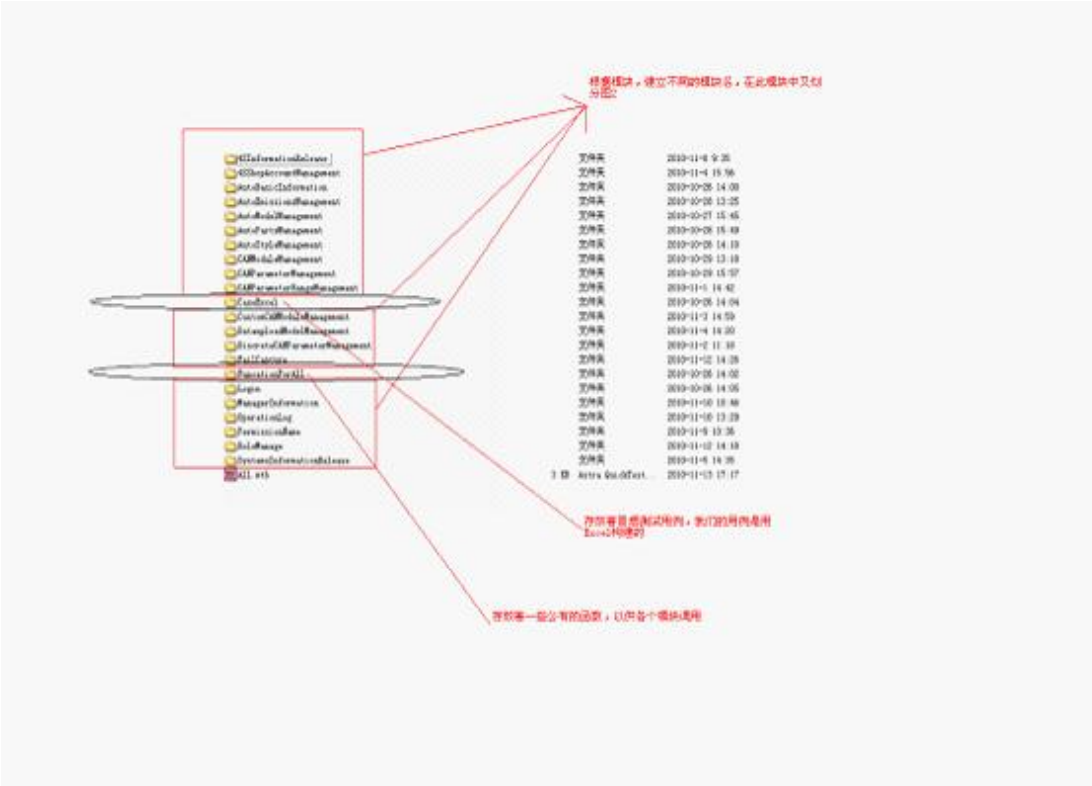
BasicFunction：供 QTP 调用的基本函数

数据：写死到脚本之中

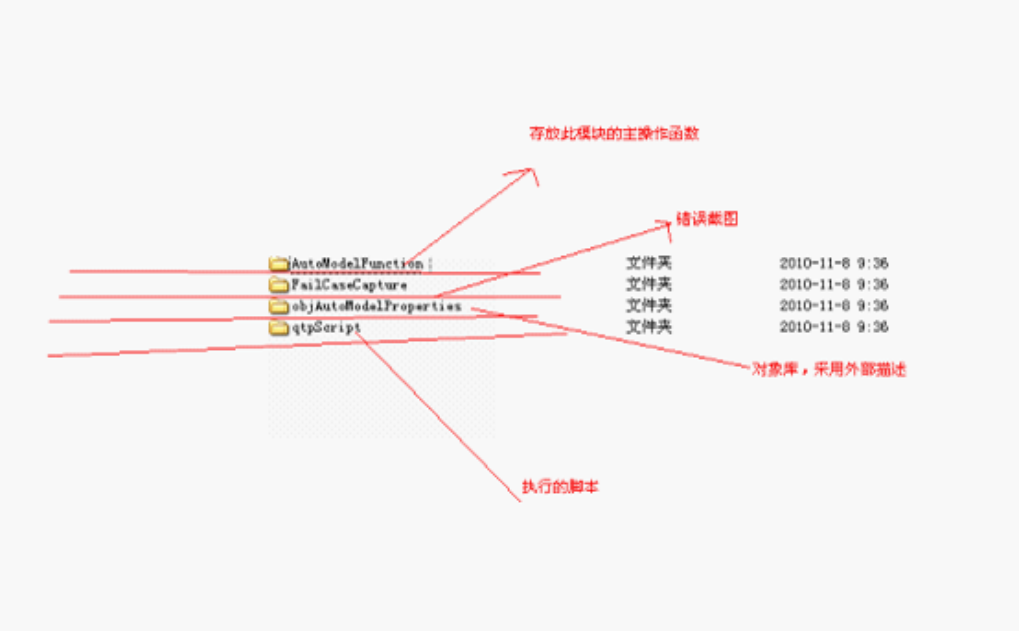
运行方式：Test Batch Runner 批量运行

其文件目录组成为：

外层主目录：



内层目录：



脚本的组织方式：

外部描述对象库后另存为 vbs 文件，并在脚本中使用 ExecuteFile 导入如：

```
Set AutoModelPage=Browser("i-CARE 后台管理").Page("i-CARE 后台管理")
Set AutoModelFrame=AutoModelPage.Frame("html id:=rightframecontent")
Set AutoModelTable=AutoModelFrame.WebTable("column names:=序号;标题;发布城市;发布人;审核情况;发布时间;查看")
```

业务操作函数：

封装其用例的主要过程在外部 vbs 中，并在 QTP 中使用 ExecuteFile 导入如：

```
' 标题查找
Function Search_Title(objFrame,setTitle)
    objFrame.WebEdit("html id:=title").set setTitle
    objFrame.Link("innertext:=查找","class:=a_search").Click
    wait(1)
    Rowc=AutoModelTable.RowCount
    if Rowc>1 then
        for i=2 to Rowc 'RowCount
            setT=trim(AutoModelTable.GetCellData(i,2))
```

```
                if RegExpTest(setTitle,setT)<>"" then
                    Search_Title=1
                else
                    Search_Title=0
                    i=Rowc
                end if
            next
        else
            Search_Title=-1
        end if
    end Function
```

根据业务操作函数写结果：

```
'****************定义相关参数
*************************************************************************
***start
FailCaseCapture="\\10.9.146.182\IBookBackgroundQtpScript\FailCapture"
'根据不同的模块设置不同的地址

CaseExcel="\\10.9.146.182\IBookBackgroundQtpScript\CaseExcel\IBookBackGroundPretest.xlsx"
FuncationForAll1="\\10.9.146.182\IBookBackgroundQtpScript\FuncationForAll\fucations.vbs"
ExecuteFile objProperties        '汽车基本信息界面的对象库
ExecuteFile FuncationForAll1     'qtp 公共函数
'*************************************************************************
**********************end
'********调用初始方法
*************************************************************************
****start
QTP_Small()                '让 QTP 最小化运行
```

```
'**********************************************************************
**********end
Url "http://10.9.10.22:8801/Login.aspx"                    '导航入登录页面
loginIBookF "2008","123456","1234"
'_____________________________________________________start
Browser("i-CARE 后台管理").Page("i-CARE 后台管理").Link("信息发布").Click
Browser("i-CARE 后台管理").Page("i-CARE 后台管理").Link("4S 店信息审核
").Click
'_____________________________________________________start
If LinkFsInfo(AutoModelPage,AutoModelFrame)=1 then
    writePass 155
else
    writeFail 155,"链接失败"
end If
'_____________________________________________________end
```

想法的来源：

1、网上许多人说，应尽量采用描述性编程来实现脚本，未充分理解对象层与应用层隔离的思想

2、《QTP 自动化测试进阶》中的框架搭建思想的影响

出发点：

1、在外部文件中能够更好更及时的管理对象库、方法，减少打开 QTP 去维护对象库的麻烦

2、尝试 QTP 框架的搭建

该方法主要特点：

1、将模块分类别进行管理

2、将对象从 QTP 仓库对象中抽取出来，用外部描述的方式生成

3、将测试主过程方法的脚本与最终生成测试结果的脚本进行分离

实际编写过程中出现的困难：

1、文件目录呈 4 级分布，且日志截图位于第 4 级之中，查看不同模块的截图需要进入不同模块的 3 级目录中的文件夹中查看，导致文件目录点击操作过多，手有时候都会点痛，而且容易迷路。

2、对象库维护反而十分不方便。不仅有 1 中文件目录操作的问题。另外，在实际脚本中采用 ExceteFile 导入对象库，当实现某个测试点后，再实现另外一个测试点，有时候需要再次 ExcuteFile 对象库，否则 QTP 会报错说找不到对象库。

即脚本的实际形式变成了：

ExcuteFile 对象库

测试点 1

ExceuteFile 对象库

测试点 2

当实际系统存在某几个对象具有相同属性时，通过此种方法无法进行描述，于是又产生了录制生成对象库的方法，最终导致对象库有描述的也有录制的，对象库管理到脚本完成时已经十分麻烦，需要到外部文件和对象仓库中修改，搭建此方法的初衷完全被搞混乱了。

3、冒烟测试用例没有进行归纳

每一条测试用列对应一个外部测试方法，致使大部分具有相同操作过程的测试点，反复进行方法的编写，导致工作量增加。

4、脚本方法采用外部文件引入，当出错时调试十分的不方便，QTP 无法进行准确的错误提示

这一点是 QTP 最令人头痛的，如果你使用的是 ExecuteFile 函数的话。当然即使你在 QTP 里面调试脚本，也会遇到此问题。

5、实际 QTP 运行脚本，对错误结果只有一个简单的截图，有时无法根据这一个步骤的截图分析 Fail 的原因。另外，ExecuteFile 的方式采用的是调用共享的方式，调用共享的方式比较强烈地依据这个 IP，同时在共享中写数据时速度也相当的慢。

6、运行方式采用 Test Batch Runner 批量运行，此工具在批量运行时似乎有些不稳定。

7、没有结合 QTP 的故障恢复机制，当 QTP 出错时无法获知出错于否。

8、数据以及变量的值写死到脚本之中，不利于维护。

框架是什么？框架要做到什么样的效果？

框架就是一种脚本组织和管理的方式 ，能够达到下面的要求：

1、要能方便地维护对象库和相关方法

2、能够将模块进行分类管理

3、要能够方便地管理脚本和数据

4、要能够处理 QTP 故障

5、要能够提高开发脚本效率

6、要适应不同的环境

该方法可取之处在哪儿：

1、将模块分类管理

2、将测试主过程方法的脚本与最终生成测试结果的脚本进行分离

该方法这两种思想的来源是没有错的，基本附合框架所要达到的要求，但是在实施的过程中，由于其目录分得太细，反而影响了工作效率。另外，对于第 2 点在针对冒烟测试的需求时似乎投入的太大（往往冒烟测试规模较小），造成了一定程度上脚本封装上的浪费，因为封装会花不少时间（封装的过程只提供了一次使用）。

那么从上我们可以得出这么一个结论：对于要求快速开发，且规模较小的冒烟测试来说，框架构建得太过灵活反而会耽误其开发的时间；由于其数据和用例具有一次性的待点，所以我们不需要针对其进行特别的封装。那么我认为自动化在做冒烟测试时，应如何呢？我个人建议，想尽一切办法跟老大说不做，借口和理由自己想吧。但是如果说不通，非得做的话，那么您就先整理用例，以便能尽量数据驱动，然后采用录制+检查点+qtp 的 DateTable 管理数据的方法生成脚本（根据实际情况进行少量函数的封装），并且把结果写入到 QTP 的 Result 报告之中。当然以上前提是你的系统，界面元素没有太大的变化，如有变化，可能你得及时的改正之。

第二部分：第二个失败的框架（供系统测试用）

文档目录的组织方式：

模块：

模块名一

模块名二

模块名三

Case：使用的是系统测试用例

BasicFunction：

基本函数 BasalFunction，封装了一些 QTP 常用函数

常用函数 CommonFunction，封装了一些该测试系统的常用函数，即每个模块都可能会用到的函数

业务操作函数 OperationFunction，封装了该测试系统的测试函数

数据：

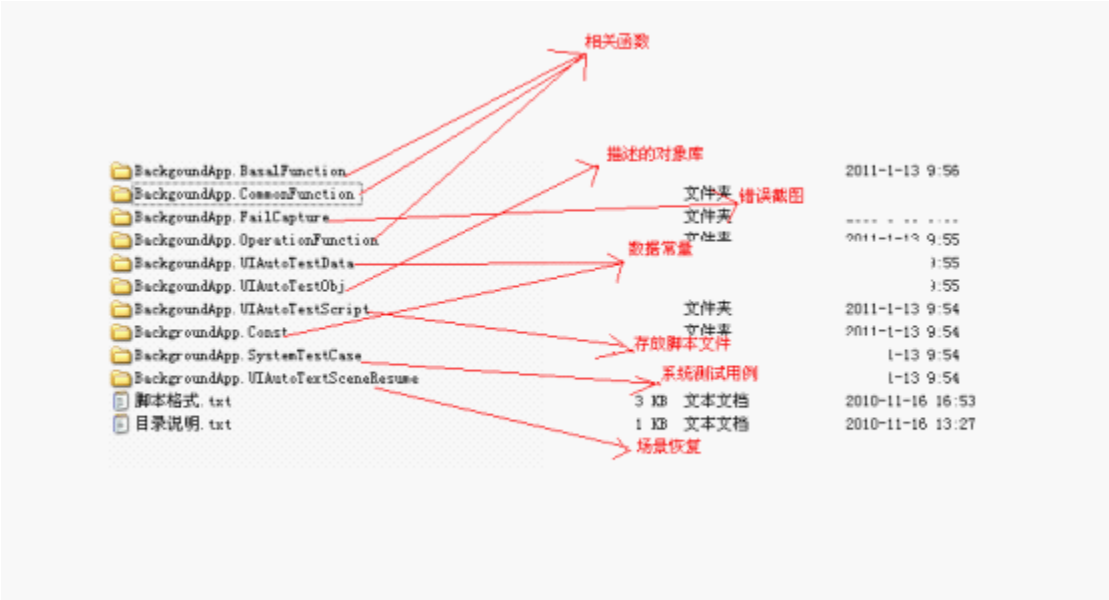
Const：存放配置变量

UIAutoTestData：存放与 Case 对应形成测试数据

运行方式：Test Batch Runner 批量运行（多台电脑）

其文件目录组成为：

外层主目录：



目录简单的说明：

BackgroundApp.SystemTestCase：放置后台系统测试的用例

BackgoundApp.Const：放置定义的常量

BackgoundApp.BasalFunction：放置供业务函数需要调用的函数组件，此模块中的函数可以供 QTP 调用

BackgoundApp.CommonFunction：放置供后台系统调用的公用函数组件，以及常用函数

BackgoundApp.OperationFunction：放置业务操作的函数

BackgoundApp.UIAutoTestScript：放置 QTP 的脚本

BackgoundApp.FailCapture：放置 FAIL 的用例时系统的截图

BackgoundApp.UIAutoTestObj：放置设置的对象

BackgoundApp.UIAutoTestData：放置测试数据

BackgroundApp.UIAutoTextSceneResume：放置场景恢复

内层目录：

内层目录中的内容直接为相应具体文件

脚本的组织方式：

外部描述对象库后另存为 vbs 文件，并在脚本中使用 ExecuteFile 导入：

Set AutoModelPage=Browser("i-CARE 后台管理").Page("i-CARE 后台管理")

Set AutoModelFrame=AutoModelPage.Frame("html id：=rightframecontent")

Set AutoModelTable=AutoModelFrame.WebTable("column names：=序号；标题；发布城市；发布人；审核情况；发布时间；查看")

业务操作函数：

1、抽取该系统可能多处用到的函数，封装到在外部 vbs 中，并在 QTP 中使用 ExecuteFile 导入：

如：

'作者：qwen

'日期：2010-11-16 下午

'点击某链接，并判断链接是否成功

'beforeLinkObj：链接前的对象

'beforeWebElementObj：判断文字前的对象

'LinkID：链接的 html id

’webElementInnerText:文字的内容

Function IBook_Link(beforeLinkObj,beforeWebElementObj,LinkID,webElementInnerText)

beforeLinkObj.Link("html id:="&LinkID).Click

```
        b=
beforeWebElementObj.WebElement("class:=span_topNavNow","innerText:="&web
ElementInnerText).Exist(2)
        if  b   then
            IBook_Link=1
        else
            IBook_Link=0
        end if
    End Function
```

2、封装其用例的主要过程在外部 vbs 中，并在 QTP 中使用 ExecuteFile 导入

如：

```
'根据 Data 设置的 flag 进行相应判断处理
'popMsg 为 Data 中设置的值
Function chk_Login(lgName,pwd,vod,lgFlag,popMsg)
        executeFile   MYEXECUTEFILE_PATH
        if lgFlag=0 then
            ‘调用登录过程
            Login lgName,pwd,vod
            ’如果在登录页存在该提示信息，则返回 1，提示信息从 data
中获得
            if                          lgPage.WebElement("html
id:=LoginMsg","innertext:="&popMsg).Exist(3) then
                chk_Login=1
            else
                ’如果登录成功，且登录成功页存在，则执行注销操作后
并返回 0
                if  Not  lgPage.WebEdit("html  id:=LoginName").Exist(1)
then
                    ‘调用退出过程
                    exit_Login()
```

```
                    end if
                        chk_Login=0
                    end if
                elseif lgFlag=1 then
                    Login lgName,pwd,vod
                ’如果登录成功，且登录成功页存在，则返回 1,否则返回 0
                if Not lgPage.WebEdit("html id:=LoginName").Exist(1) then
                    exit_Login()
                    chk_Login=1
                else
                    chk_Login=0
                end if
            else
                MessageBox ″登录数据表中的期望结果有除 0 和 1 以外的值
"
            end if
        End Function
```

3、在 qtp 中根据上面两个函数，最终实现的 QTP 脚本为：

```
'*******************************************************脚本基本
信息 start
'       脚本名称：系统测试登录模块
'       脚本目的：自动执行  I-Book 后台管理系统预测试用例.xlsx  中的
用例
'       时间：2010-11-15 至 2010-11-16，其中 2010-11-15 档建测试架构，
2010--11-16 号早上完成此模块的脚本
'       作者：文青山
'*******************************************************end
'*************************************************************
公共调用的信息 start
'此目录中放入所有函数的路径等概况了一些函数
```

```
MYEXECUTEFILE_PATH="\\10.9.146.118\testing\IBookBackgroundApplication\BackgroundApp.Const\LgExecuteFile.vbs"
'QTP 自动保存的目录
RES_PATH="\\10.9.146.118\testing\IBookBackgroundApplication\BackgoundApp.UIAutoTestScript\login\Res*"
'导入
executeFile    MYEXECUTEFILE_PATH
SmallQTP()
'*********************************************************************公共调用的信息 end
'————————————————————————————————各种登录数据的测试 start
For i=6 to 50
' 读取操作步骤
getLgChkPoint=readLgData(i,2)
' 读取用户名
getLgName=readLgData(i,4)
' 读取密码
getLgPwd=readLgData(i,5)
' 读取验证码
getLgVod=readLgData(i,6)
' 读取 flag 标志值
getLgFlag=readLgData(i,7)
' 读取提示信息
getPopMsg=readLgData(i,8)
' 调用登录检查函数
  lgType=chk_Login(getLgName,getLgPwd,getLgVod,getLgFlag,getPopMsg)
' 根据返回的信息在 excel 中写入相关信息
j=cint(i+17)
If lgType=1 Then
```

```
            writeLgPass    j
    else
            writeLgFail      j,getLgChkPoint&″ 失 败 !"&" 用 户 名 ：
"&getLgName&"密码："&getLgPwd&"."
    End If
Next
CloseProIE()
’————————————————————————————————————————各种登录数据的测试 end
```

想法的来源：

1、仍未摆脱“应尽量采用描述性编程来实现脚本”的思想，未充分理解对象层与应用层隔离的思想。

2、对失败的框架 1 进行总结，着重解决脚本目录组织的问题以及常量文件配置的方法。

3、理解并试着采用数据驱动的思想，试着建立系统测试用例与测试数据建立的关系。

4、试着使用脚本分离的方法进行脚本的重构。

出发点：

1、解决失败的框架 1 中出现的部分问题

2、在外部文件中能够更好更及时的管理对象库、方法，减少打开 QTP 去维护对象库的麻烦

3、尝试使用数据驱动的方法来进行测试

4、尝试建立测试用例与测试数据建立依赖关系

该方法主要特点：

1、将模块分类别进行管理

2、将对象从 QTP 仓库对象中抽取出来，用外部描述的方式生成

3、将测试主过程方法的脚本与最终生成测试结果的脚本进行分离

4、解决了失败的框架 1 中的文件目录组织的问题

5、将配置文件进行了单独管理

6、将系统测试用例与数据建立了一一对应的关系

7、使用了 QTP 的场景恢复技术，考虑了 QTP 出现异常时的情况

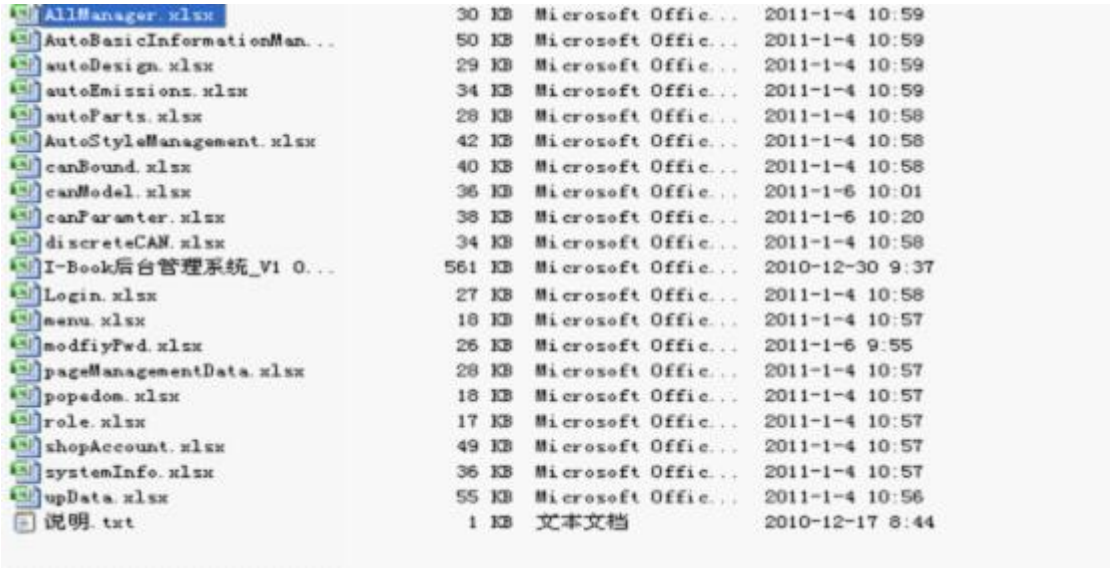
实际编写过程中出现的困难：

1、拥有失败的框架 1 中的（2、4、5、6）的问题

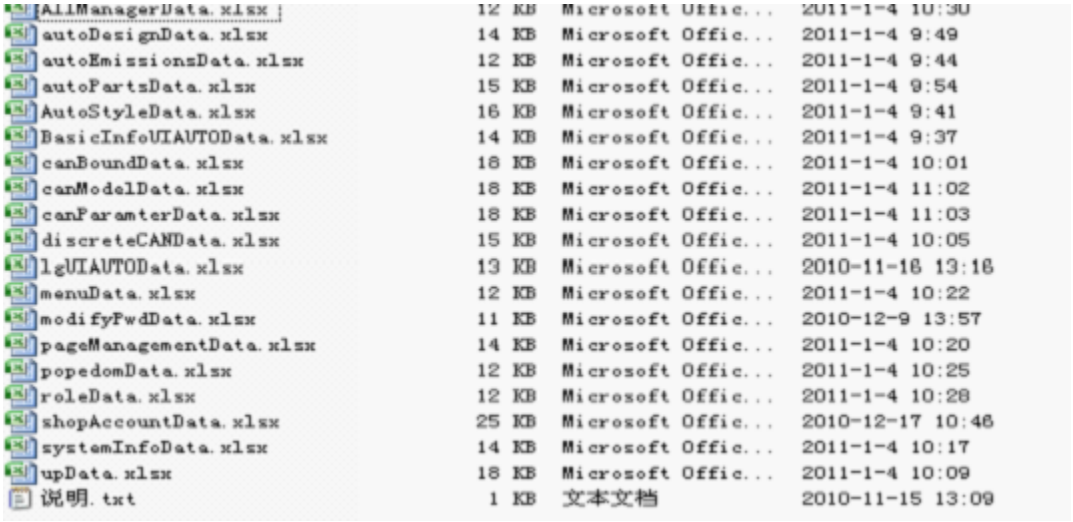
2、建立数据与测试用例一一对应关系，耗去了不少时间，并且这些数据只能够在此系统使用一次。更糟糕的是系统测试用例有些用例是概括性的，而我们在整理用例时不得不把这些用例更加详细的细化，而此又用去了不少时间。

3、由于使用了 Write_ExcelCell 函数，使得我们的测试用例必须分多个 Excel 进行管理，并且每个需要根据每个模块的用例重新构建一个写 Excel 函数。另外，采用此种方法在长时间的测试过程中，excel2007 有可能会遇到一些异常问题，导致 excel.exe 进程不能及时关掉，重而导致 QTP 脚本在下一次写结果的时候无法写入，造成 excel 中的结果为空。

下图示举了重新分割的测试用例文件：



下面示举了根据上图的测试用例生成的一一对应的测试数据：



AllManagerData.xlsx	12 KB	Microsoft Offic...	2011-1-4 10:30
autoDesignData.xlsx	14 KB	Microsoft Offic...	2011-1-4 9:49
autoEmissionsData.xlsx	12 KB	Microsoft Offic...	2011-1-4 9:44
autoPartsData.xlsx	15 KB	Microsoft Offic...	2011-1-4 9:54
AutoStyleData.xlsx	16 KB	Microsoft Offic...	2011-1-4 9:41
BasicInfoUIAUTOData.xlsx	14 KB	Microsoft Offic...	2011-1-4 9:37
canBoundData.xlsx	18 KB	Microsoft Offic...	2011-1-4 10:01
canModelData.xlsx	18 KB	Microsoft Offic...	2011-1-4 11:02
canParamterData.xlsx	18 KB	Microsoft Offic...	2011-1-4 11:03
discreteCANData.xlsx	15 KB	Microsoft Offic...	2011-1-4 10:05
lgUIAUTOData.xlsx	13 KB	Microsoft Offic...	2010-11-16 13:16
menuData.xlsx	12 KB	Microsoft Offic...	2011-1-4 10:22
modifyFwdData.xlsx	11 KB	Microsoft Offic...	2010-12-9 13:57
pageManagementData.xlsx	14 KB	Microsoft Offic...	2011-1-4 10:20
popedomData.xlsx	12 KB	Microsoft Offic...	2011-1-4 10:25
roleData.xlsx	12 KB	Microsoft Offic...	2011-1-4 10:28
shopAccountData.xlsx	25 KB	Microsoft Offic...	2010-12-17 10:46
systemInfoData.xlsx	14 KB	Microsoft Offic...	2011-1-4 10:17
upData.xlsx	18 KB	Microsoft Offic...	2011-1-4 10:09
说明.txt	1 KB	文本文档	2010-11-15 13:09

下图示举了上图测试用例与数据一一对应中的内容：

	A	B	C	D	E	F	G	H	I	J
1	功能描述	操作步骤	预期结果	数据上限	数据下限	小数点位数	标志位	提示语		
2	添加/修改	添加/修改数据上限								
3	特殊字符的处理	1、添加->数据上限为特殊字符（如：*等）->确定->观察	1、提示相关信息，您输入的信息中包含非法字符（\,<,>,&）！	*	随机	随机	0	数据上限、数据下限、小数点位数必须为数字！		
4	特殊字符的处理	1、添加->数据上限为特殊字符（如：%）->确定->观察	1、提示相关信息，您输入的信息中包含非法字符（\,<,>,&）！	%	随机	随机	0	数据上限、数据下限、小数点位数必须为数字！		
5	特殊字符的处理	1、添加->数据上限为特殊字符（如：'）->确定->观察	1、提示相关信息，您输入的信息中包含非法字符（\,<,>,&）！	'	随机	随机	0	数据上限、数据下限、小数点位数必须为数字！		
6	特殊字符的处理	1、添加->数据上限为特殊字符（如：&）->确定->观察	1、提示相关信息，您输入的信息中包含非法字符（\,<,>,&）！	&	随机	随机	0	数据上限、数据下限、小数点位数必须为数字！		
7	特殊字符的处理	1、添加->数据上限为特殊字符（如：+）->确定->观察	1、提示相关信息，您输入的信息中包含非法字符（\,<,>,&）！	+	随机	随机	0	数据上限、数据下限、小数点位数必须为数字！		
8	特殊字符的处理	1、添加->数据上限为特殊字符（如：[]）->确定->观察	1、提示相关信息，您输入的信息中包含非法字符（\,<,>,&）！	[]	随机	随机	0	数据上限、数据下限、小数点位数必须为数字！		
9	特殊字符的处理	1、添加->数据上限为特殊字符（如：\）->确定->观察	1、提示相关信息，您输入的信息中包含非法字符（\,<,>,&）！	\	随机	随机	0	数据上限、数据下限、小数点位数必须为数字！		
10	特殊字符的处理	1、添加->数据上限为特殊字符（如：|）->确定->观察	1、提示相关信息，您输入的信息中包含非法字符（\,<,>,&）！	|	随机	随机	0	数据上限、数据下限、小数点位数必须为数字！		
11	特殊字符的处理	1、添加->数据上限为特殊字符（如：'）->确定->观察	1、提示相关信息，您输入的信息中包含非法字符（\,<,>,&）！	'	随机	随机	0	数据上限、数据下限、小数点位数必须为数字！		
12	特殊字符的处理	1、添加->数据上限为特殊字符（如：''）->确定->观察	1、提示相关信息，您输入的信息中包含非法字符（\,<,>,&）！	''	随机	随机	0	数据上限、数据下限、小数点位数必须为数字！		
13	特殊字符的处理	1、添加->数据上限为特殊字符（如：/）->确定->观察	1、提示相关信息，您输入的信息中包含非法字符（\,<,>,&）！	/	随机	随机	0	数据上限、数据下限、小数点位数必须为数字！		
14	特殊字符的处理	1、添加->数据上限为特殊字符（如：?）->确定->观察	1、提示相关信息，您输入的信息中包含非法字符（\,<,>,&）！	?	随机	随机	0	数据上限、数据下限、小数点位数必须为数字！		
15	特殊字符的处理	1、添加->数据上限为特殊字符（如：>）->确定->观察	1、提示相关信息，您输入的信息中包含非法字符（\,<,>,&）！	>	随机	随机	0	数据上限、数据下限、小数点位数必须为数字！		
16	特殊字符的处理	1、添加->数据上限为特殊字符（如：<）->确定->观察	1、提示相关信息，您输入的信息中包含非法字符（\,<,>,&）！	<	随机	随机	0	数据上限、数据下限、小数点位数必须为数字！		
17	特殊字符的处理	1、添加->数据上限为特殊字符（如：@）->确定->观察	1、提示相关信息，您输入的信息中包含非法字符（\,<,>,&）！	@	随机	随机	0	数据上限、数据下限、小数点位数必须为数字！		
18	特殊字符的处理	1、添加->数据上限为特殊字符（如：#）->确定->观察	1、提示相关信息，您输入的信息中包含非法字符（\,<,>,&）！	#	随机	随机	0	数据上限、数据下限、小数点位数必须为数字！		
19	特殊字符的处理	1、添加->数据上限为特殊字符（如：$）->确定->观察	1、提示相关信息，您输入的信息中包含非法字符（\,<,>,&）！	$	随机	随机	0	数据上限、数据下限、小数点位数必须为数字！		

下面为 Write_ExcelCell 方法的详细内容，该函数的原始函数在网络上广泛流传，我相信有一大部分人在刚开始接触 QTP 对 Excel 进行操作时都是使用的该方法。下面的这个函数就是基于网络上流传的那个函数的变种而已。

'pathway ： excel 所处的位置

'sheetName：表名

'write_Row：要写入的行

'write_PassCol：实际结果的列名

'write_RemarkCol：备注的列名

'write_VersionCol：版本名的列名

```
'write_DateCol：日期的列名
'write_PersionCol：测试人员的列名
'result_Pass：写入的实际测试结果
'result_Remark：写入的备注
'result_Version：写入的版本
'result_Date：写入的日期
'result_Persion：写入的测试人员
'*****************************************使用的方法
'按现有模块的格式写入 excel 相关信息
        Public  Function
Write_ExcelCell(pathWay,sheetname,write_IntRow,write_PassCol,write_RemarkCol
,write_VersionCol,write_DateCol,write_PersionCol,result_Pass,result_Remark,result
_Version,result_Date,result_Persion)
            Dim srcData,srcDoc,write_Row
            '转换为整形
            write_Row=cint(write_IntRow)
            set srcData = CreateObject("Excel.Application")
            '可见
            srcData.Visible = false
            '打开文档
            set srcDoc = srcData.Workbooks.Open(pathway)
            srcDoc.Worksheets(sheetname).Activate
            '实际测试结果
    srcDoc.Worksheets(sheetname).Cells(write_Row,write_PassCol).value =
result_Pass
            '备注
    srcDoc.Worksheets(sheetname).Cells(write_Row,write_RemarkCol).value
=result_Remark
```

```
                '版本
    srcDoc.Worksheets(sheetname).Cells(write_Row,write_VersionCol).value
=result_Version
                '日期
    srcDoc.Worksheets(sheetname).Cells(write_Row,write_DateCol).value
=result_Date
                '测试人员
    srcDoc.Worksheets(sheetname).Cells(write_Row,write_PersionCol).value
=result_Persion
                    '保存
                srcData.Workbooks(1).Save
                wait(2)
                srcData.Workbooks.Close
                '退出 Excel 进程
                srcData.Quit()
                set srcData=nothing
                set srcDoc=nothing
                Set write_Row=nothing
            End Function
```

4、使用 QTP 的故障恢复机制，严重依附于对象库，且这些异常需要预先知晓。并且当出现了这些异常干扰了测试运行时，我们并不能及时的得到通知，至使我们不得不每天上午和下午人工去看看出现异常没有。

5、由于本方法大量使用了 vbs 文件，而 QTP 在使用 ExecuteFile 导入文件时，将所有导入的文件都存入了内存之中，又由于它们都是基于函数和过程的，QTP 并不能智能的根据所导入文件名称不同而区别所有函数的名称，所以导致很容易出现命名同名的情况，而 QTP 糟糕的提示信息，往往造成你不知所因。

6、在测试脚本编写的过程中，有不少地方使用了 Vbscript 的 sendKey 方法，在实际的测试过程中发现此似乎非常不稳定，有时并不能良好的 sendKey，从而造成脚本报错。

7、系统中存在很多具有相同属性的对象，当使用描述性编程去处理它们时增加了不少的工作量。

8、日志记录太少，仅仅只有一个错误截图，当出现 Fail 的用例时，并不能根据此截图分析出问题所在。另外，所有截图文件放在一个文件夹中，当 QTP 运行出现异常时，就会导致许多截图文件生成在这个文件中，从上千张的截图文件中获取你要的信息，相当麻烦。

9、相关方法的注释不详尽，在过了一段时间要去修改某段代码时，有时竟然再也看不懂了。

框架是什么？框架要做到什么样的效果？

框架就是一种脚本组织和管理的方式，并附有一系列的规则的要求，除要能够满足“失败的框架 1”中的要求外，似乎还需要以下要求：

1、要能提供详尽的日志，最好是每个执行步骤的日志。

2、要能够很好的处理测试用例与对应数据的依存关系。

3、要能保持适量封装的函数，并且能够使这些函数尽量的多次重用，所谓的多次有可能是针对本系统的也有可能是多个不同系统的。

4、要在 QTP 出现异常时及时的通知到我们，让我们能够及时的知道有此异常。

5、要抛弃那些不稳定、不好用的方法，转而使用能够提供稳定运行、简单好用的方法。

6、要代码不仅当时能够自己看懂，而且过了很久也要能看懂，并且还要能让其它人看懂。

7、要数据能够尽量的重用，最好做到原始数据能多系统共用。

8、要使常量尽量的统一在一起，以便维护。

该方法可取之处在哪儿：

1、文件目录简洁，操作不复杂。

2、将配置常量集中管理。

3、将脚本划分为 QTP 基本函数、测试系统常用函数、测试函数、实现函数。

4、建立测试用例与数据对应的关系，并采用了数据驱动的方法。

5、考虑到了 QTP 运行异常，使用了 QTP 故障恢复机制来尝试解决这些问题。

在构建此方法时，对框架的理解很肤浅，考虑的出发点仅仅是如何建立数据与用例的对应关系，并考虑解决在“失败的框架 1”中出现的目录构架的问题。虽然该方法引用了故障恢复机制，但 QTP 的故障恢复机制确实很不好用，因为几乎所有的操作都要求能基于仓库对象和已知异常。

如何建立测试用例与测试数据的对应关系？在此方法中进行了尝试，某些书上说的建立测试数据与测试用例的一一对应关系，从实际使用的效果来说似乎并不怎么好。同时，由于使用读取 Excel 函数的问题，导致出现了预期外的额外工作。尤其是系统测试用例的某些测试用例有些是具有概括性，在构建这些用例的测试代码时，还得去改造用例。同时我也深刻地领会到在构建脚本的过程中一个坏函数，可能会影响你整个架构的实现，也有可能会影响到你脚本的测试结果。

在编写脚本的过程中，由于注释的缺少引起了后期维护的问题。命名的规则太随意也导致了大量意想不到的问题。通过外部构建测试对象的方法，也是非常二的！至少通过使用 vbs 文件这种方法来构建我是不赞成的。另外，老大要求将测试结果直接写入 Excel 中，并安排这些用例不再经过测试人员进行测试的这种与手工测试策略结合的方法也非常的令人担忧。

第三部分：GUI 自动化测试框架 qwenTest

文档目录的组织方式：

0DocumentLayer：里面包括一些流程管理文件，做自动化测试前需要走完此流程

1ObjectLayer：提供对象库，该对象库可以是 QTP 的对象库，此处采用共享对象库的方法；同时这里也可以是基于 Excel 进行管理的对象库：

2BasicLayer：提供了一些常用的方法类。其主要方法 TestObject 可以从 Excel 中读取对象的属性和属性值来建立对象库：

3ConfigLayer：用于存放配置内容，所有配置文件均在此统一管理，并使用 xml 文件来实现。

4OpertionLayer：用于存放一些本测试项目可能公用的测试方法

5DataLayer：装载可重用数据和私有数据

6MonitorLayer：提供了一个软件来监视 QTP 的运行异常

7RunEngineLayer：用于存放.vbs 的启动测试脚本

8ScriptLayer：用于存放脚本

9ResultLayer：用于存在测试结果或者日志

内层目录：

1ObjectLayer：

automationTestDoc.xls，该 Excel 包括自动化测试申请表，自动化与手工结合的方案，自动化工作进度计划，自动化测试用例，脚本规范，脚本更改申请，自动化测试结果等文档的模板。

自动化测试实施的流程图.doc，该 Word 定义了严格的自动化实施流程，自动化测试项目必须走这个流程。

1ObjectLayer：

shareObject_20100707.tsr，存在共享对象库。脚本的实现采用共享对象库的方式来管理。

lgObject.xls，存在基于 Excel 的对象库管理，管理方式如下图所示：

A	B	C	D	E	F	H	I	J	K	L
操作对象类型	属性1名称	属性2名称	属性3名称	属性4名称	属性5名称	属性1的值	属性2的值	属性3的值	属性4的值	属性5的值
WebEdit	html id					testing				

2BasicLayer：

Class_Basic.qfl 文件，基于类的方法来构造 Class_Basic 类提供 TestObject(Byref testObjType,Byref arrProperty,Byref arrPropertyValue)方法。该方法利用数组的特性，从 Excel 中获取测试对象的参数可以活灵传递。如 TestObject(“WebEdit”,arry(“name”),arry(“userName”))来生成用户名对象，也可以 TestObject(“WebEdit”,arry(“name”,”id”),arry(“userName”,”userName”))等不定参数。使用该方法可以脱离 QTP 对象库来进行获取对象和管理对象，并且该方法会详细记录对象是否存在的详细日志。该方法非常长，本想贴出来的，可看了一下实在是太长了，我都写了两个晚上才完成，所以在此就省略了。

Class_InitTest.qfl，基于类的方法来构造 Class_InitTest 类，提供了很多初始电脑环境的方法。比如结束 IE 进程，将浏览器最大化等函数。

Calss_String.qfl，基于类的方法来构造 Class_String 类，提供了很多字符串处理方法。比如，随机生成身份证号等。

3ConfigLayer：存在配置文件，其它略

4OpertionLayer：一些可能本测试系统将会重用到的封装方法

5DataLayer：

ClassRand.dll，.Net 类，该类中提供了 10 几种随机生成指定长度的字符串的方法。比如，随机生成 0-9 的数字，随机生成 a-z 的大小写单词，随机生成汉字，随机生产与业务规则相符合的字符串等，尽量做到正确数据正确的模拟。由于是.Net 类库，这些方法生成值的速度比在 QTP 中放了很多。另外在此类库中提供了用数据库的方式读写和更新 Excel，相比以前用 vbs 的速度提高了上千倍，并且至今在也没有出现上文中提到过的 EXCEL 异常。

personal_Data\personalData.xls，项目模块独用数据使用。

6MonitorLayer：

MonitorQTP.exe，MonitorQTP 是我专门开发的软件，该软件模拟处理了许多 QTP 测试 Web 的异常，其软件主使用方法可到 www.automationqa.com 或我的博客下载。该软件通过配置，可以实现当一发现相关异常时，就记录详细的日志和截图，并会发邮件通知你 QTP 运行出现了异常。

7RunEngineLayer：调用 QTP 的 Aom 接口，生成相应的执行文件，再配合 Windows 的计划与任务功能，代替 Test Batch Runner 的运行方式。

8ScriptLayer：存放每个脚本的地方。

9ResultLayer：记录测试结果的地方。

脚本的组织方式：脚本的组织方式，采用三层架构的思想来封装和组建。

第一层：对象层

如果是用 TestObject 方法获得对象库，则需要调用相关方法来获得 Excel 中的每一个对象。如果是 QTP 的仓库对象，那么对仓库对象每个对象的命称需要按照规则来定义。

第二层：业务层

封装其功能的主要过程，以供实现层调用。

如下面这个例子：

```
'根据所指定的角色点击控件
Function pOpedom(ByVal intRole)
  On error resume next
      If intRole=0 Then
          SwfWindow("登录").SwfRadioButton("登录_管理人员").Click
          suc =0
```

```
            else
                SwfWindow("登录").SwfRadioButton("登录_读者").Click
                suc =1
            End If
            If err.number<>0    Then
                "记录日志
                msgbox "pOpedom 函数异常信息：
"&err.number&space(2)&err.description
                suc =-1
            End If
        End Function
        "输入用户名密码登陆系统
        Function login(uName,pwd)
            SwfWindow("登录").SwfEdit("登录_图书证号").Set uName
            SwfWindow("登录").SwfEdit("登录_密码").Set pwd
            SwfWindow("登录").SwfButton("登录_确定").Click
        End Function
        "根据权限判断登陆过程是否成功
        function isLogin(ByRef intRole,ByRef uName,ByRef pwd)
          On error resume next
            Dim suc
            suc=pOpedom(intRole)
              If suc =0 then
                    Call login(uName,pwd)
                    If SwfWindow("主界面  - [添加用户]").Exist Then
                        isLogin=1
                    else
                        isLogin=0
                    End If
            elseif suc=1 then
```

```
                If SwfWindow("主界面 - [添加用户]").SwfWindow("借书").Exist
then
                        isLogin=11 "工作人员登陆成功
                else
                    isLogin=10 "工作人员登陆失败
                end if
            elseif suc=-1Then
                Exit function
            End If
            If err.number<>0   Then
                "记录日志
                msgbox "isLogin 函数异常信息：
"&err.number&space(2)&err.description
            End If
        end function
```

第三层：实现层

根据第二层组织成的测试业务判断逻辑，实现其测试如：

```
        For i=0 to Next
            uName=ReadData(“用户名”)
            pwd=ReadData(“密码”)
            blLg=islogin(uName,pwd)
            if blLg=1 then
                ‘’记录日志和结果
            Else
                ‘’记录日志和结果
            End if
        Next
```

想法的来源：

1、站在软件开发流程的管理思绪角度来考虑自动化脚本的开发

2、解决框架 1 和框架 2 中遇到过的所有问题

3、基于类的思想和三层架构的思想来构建框架和脚本

4、充分理解数据驱动的思想

5、充分理解和利用 QTP 仓库对象以及描述性编程的方法来构建对象库

6、达到框架 2（框架要做到什么样的效果）的所有要求。

该框架就暂时简单介绍到这里，此框架在此简单的描述了一下，应该有很多错误之处，还望大家指正。

最后再问一次框架是什么？框架要做到什么样的效果？

我现在的回答：框架是神马！需求才是真谛！

软件测试中的冲突测试

作者：祁超超

摘要：本文介绍了我们公司内部的一种测试方法——冲突测试的含义，并就冲突测试在我司使用范围、冲突测试用例的设计方法等做了简单的介绍。

关键词：冲突测试

1、什么是冲突测试

冲突测试是我们公司内部的一种叫法，可能不同的公司叫法不同。我们公司所谓的冲突测试是指，在运行某一程序的功能时被第三方功能或者软件给干扰的测试。该测试方法模拟的是一种基于软件状态场景的测试。从软件的运行状态来看，我们认为软件状态一般只有开始、挂起、结束，这三种状态。冲突测试即为模拟干扰软件运行“开始”、“挂起”、“结束”状态的测试。

2、冲突测试的应用范围和一些应用场景

冲突测试这种测试方法，常见于手机软件测试、移动通信类嵌入式软件测试等领域。但在一些桌面软件或者 Web 系统测试领域当中也可应用，只是应用的场景并不如移动通信类软件这么广泛。

下面简单介绍一下其不同软件类型中的典型应用场景：

手机软件：

比如在收短信时，来个电话的场景；

在播放视频过程中，插入了 USB；

在通话过程中，收到了一个短信等等。

移动通信类嵌入式软件：

移动通信类嵌入式软件，大多是基于 3G 网络应用的软件。比如一些车载软件，将此类终端装在汽车上，就可以实现类似基于手机功能，并整合互联网娱乐的功能的软件。典型软件类型，如丰田公司的“G-BOOK”、比亚迪的“i”系统等。

其测试场景比如：

在下载导航的过程中，来了一个即时消息；

在播放音乐的过程中，来了一个电话；

在蓝牙连接的过程中，又接收到了一个新的蓝牙连接请求等等。

桌面软件或者 Web 系统：

桌面软件或 Web 系统的冲突测试，常见于多线程程序、具有多系统合作程序、多权限管理的程序。

多线程程序：

比如某查询功能，由于数据量很多，查询后有一个等待加载完成的状态线程。这时我们就可以通过关闭标签或者页面，结束这个等待的状态线程；

有某导出数据的功能，在导出数据等待过程中，关闭标签或者页面，结束这个等待的状态线程。

多系统合作程序：

比如 51testing 的博客。A 系统为管理用户发表的博文，B 系统为用户使用的博客系统。A 系统用户和 B 系统用户同时都处于某个待发表的博文的审核界面，B 系统用户已删除这篇博文，但 A 系统用户由于页面不会自动刷新，A 系统用户发起了审核通过的会话申请，此时这篇博文能通过吗？

多权限管理的程序：

我们还是拿 51testing 博客系统来举例。管理用户发表博文的系统，有 A、B 用户。A、B 用户，同时处于某个博文的审核页面。A 用户，审核不通过某博文。但由于 B 用户页面不会自动刷新，B 用户又发起了审核通过的会话申请，这篇博文能通过吗？

再如，A 权限管理 B 用户。B 用户正在操作系统时，A 权限管理用户禁用了 B 用户的使用，B 用户此时的相关业务流还能继续吗？

3、冲突测试应该在整个测试中的地位

冲突测试在整个测试中的地位应该与边界值测试中的地位相同。在设计测试用例时，也应较多的考虑这类测试。根据以往采用这种测试方法进行测试的经验来看，此类测试往往会引起一些较严重的问题。比如上文中提示到“在播放视频过程中，插入了 USB”，就引起过手机软件黑屏，且不能恢复的 p1 级问题；“在下载导航的过程中，来了一个即时消息”，引起车载导航功能在界面中卡死，只有重启才能恢复软件运行的问题；“查询功能，由于数据量很多，查询后有一个等待加载完成的状态线程。这时我们就通过关闭标签结束这个等待的状态线程”，引起桌面软件彻底崩溃，只有通过进程结束才能恢复的问题；多系统合作程序那个例子，在我们做过的论坛系统中，引起过文章被发表通过的问题。

为什么说冲突测试跟边界值测试法的地位相同呢？

因为说白了其实冲突测试就是边界值法测试方法的一种具体体现。对边界值法有一个小小的误区，无论是书上还是网络上通常说到边界值法的时候，举的例子往往是数量边界，即某个输入框只能为 20 位字符时，我们测试其 21 位字符的情况，这当然也没有错，只是常此以往，很多测试人员认为边界值法就是数据边界了。其实我认为边界值法，包括三种：输入动作的边界，数量的边界，以及状态的边界。所谓输入动作的边界，比如查询有三个输入框，一个框都不输入去查询，这就是一个输入动作的边界。数量边界，即一个值的最小和最大数量。状态边界，即“开始”、“挂起”、“结束”这三种状态。由于本文重点是介绍冲突测试这种方法，关于边界值就不继续禅述了。

4、冲突测试用例的设计方法和执行策略

此类测试用例设计的方法与其它类型的用例设计方法基本上相同。不过，不同的软件类型其用例设计方法的偏重点也略有不同。我们先来看看手机和移动通信类嵌入式系统的用例设计大体方法和执行策略：

A）确定可打断其它功能状态的功能。

比如手机功能：一般通话、蓝牙、短信、彩信、闹钟、USB、充电等功能，都会弹出一个提示信息层。所以冲突测试用例，都必须考虑这种功能打断其它程序时会不会引起一些异常等。

B）遍历所有功能模块

分析功能模块的使用频率，将不同的使用频率的模块按不同的级别划分，以便有利于测试策略的安排。

C）执行策略

此类测试在手机等移动通信软件中的一个难点就是，找准测试时间点。因为在执行某个功能时，发生的速度非常快，而要打断这个状态就得算好时间。这一点，我想做过手机软件测试的同志是非常清楚的。我所采用的方法，一般是在这个功能使用手机秒表功能计算并统计时间后取出平均时间来执行的。另外，此类用例在前期应执行级别最高的用例，到中后期按照多个版本遍历所有级别的原则进行组合策略按排。

桌面软件和 Web 系统：

A）找出多线程和实现时间比较耗时的功能点

在用例设计的前期，可能这些通过界面是看不出来的，所以此时得寻求开发的帮助。当然也可以在系统出来后，快速地了解系统那些涉及到了的这些点，然后再补充之。

B）确定系统与系统之间某些功能的联系

一般去找那些具有多系统，操作相同功能的功能点。这种情况并不多，但在复杂系统中还是存在的。

C）确定此系统的用户权限关系

此种情况，我的测试生涯遇到的情况是比较多的，凡是涉及权限管理的系统，都应在此下一定功夫来设计此类用例。

D）在上面的基础上，按照状态边界的用例设计法写出用例即可。

E）执行策略

此类用例应置为高级别的用例，在每轮测试时都应该测试。

5、冲突测试用例的实例

手机软件，如短信模块的一些用例：

模块	用例概要	测试步骤	期待结果
短信模块	来电话	1. 从主菜单界面进入短信界面过程中，有电话呼入	1. 电话正常呼入，仍处于主菜单界面
短信模块	来电话	1. 进入写短信界面过程中，有电话呼入	1. 电话正常呼入，仍高亮于写短信菜单
短信模块	来电话	1. 在发送短信过程中，有电话呼入	1. 电话正常呼入，该短信发送失败，保留在收件箱内
短信模块	来闹钟	1. 从主菜单界面进入短信界面过程中，闹钟时间到	1. 正常弹出闹钟，关闭闹钟后，仍处于主菜单界面
短信模块	来闹钟	1. 进入写短信界面过程中，闹钟时间到	1. 正常弹出闹钟，关闭闹钟后，仍高亮于写短信菜单
短信模块	来闹钟	1. 在发送短信过程中，闹钟时间到	1. 正常弹出闹钟，关闭闹钟后，该短信发送失败，保留在收件箱内

桌面软件：

如某模块有如下图的功能，此功能类型与 SQL 查询分析器，执行查询后有一个线程等待的状态，并且该模块是受其它管理控制显示的。

设计的其冲突测试用例为：

模块	用例概要	测试步骤	期待结果
SQL 查询分析器	查询大数据量时退出标签窗口	1.sql 查询分析器，sql 内容为 select * from table，并且该 table 中的数据较多 2.在等待的过程中,关闭窗口标签	1.正常退出系统，无异常
SQL 查询分析器	查询大数据量时退出系统	1.sql 查询分析器，sql 内容为 select * from table，并且该 table 中的数据较多 2.在等待的过程中，退出系统	1.正常退出系统，无异常
SQL 查询分析器	更新大数据量时退出标签窗口	1.sql 查询分析器，sql 内容为 update table set colname=value，并且更新的数据较多 2.在等待的过程中,关闭窗口标签	1.正常退出系统，无异常
SQL 查询分析器	更新大数据量时退出系统	1.sql 查询分析器,,sql 内容为 update table set colname=value，并且更新的数据较多 2.在等待的过程中，退出系统	1.正常退出系统，无异常
SQL 查询分析器	查询时该用户被禁用	1.A 用户在使用 sql 查询分析器查询数据的状态中,被管理员禁用了此功能	1.查询数据被停止，提示"无权限"后自动刷新，该功能从系统中消失
SQL 查询分析器	更新时该用户被禁用	1.A 用户在使用 sql 查询分析器更新数据的状态中,被管理员禁用了此功能	1.查询数据被停止，提示"无权限"后自动刷新，该功能从系统中消失

6、总结

本文就我公司冲突测试的定义，以及应用范围和用例设计方法及实例进行了粗略的介绍，并且这种测试方法也只是我公司内的称呼，可能并不具有广义性。本文也只是浅显地谈到了这方面的内容，可能有许多错误之处，还望各位朋友谅解。

基于 Win32 窗口的开源自动化测试工具 White

作者：思齐 译

前几天看到开源自动化测试工具 White。调试了几个例子，就被它强大之处所深深吸引，这么好的工具，但网上介绍的内容却很少，于是偶翻译了原作者的介绍文档，并在此基础上增加了目前的浅显认识，希望通过本文的介绍能够使更多的人认识它、研究它，最后有朋友再把研究所得介绍出来，那就太好了！

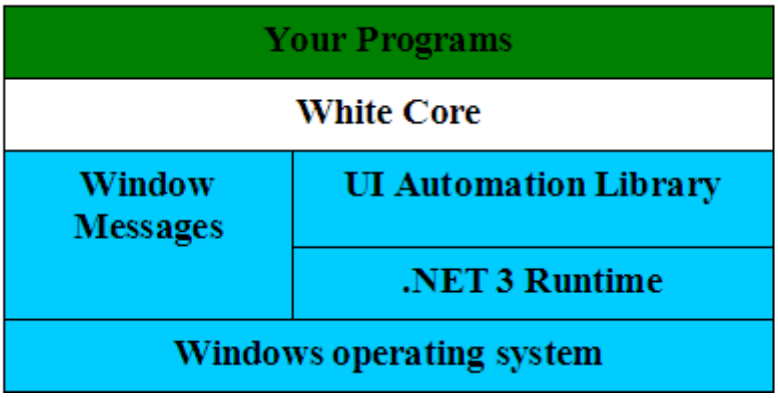
简介：

Microsoft 在.net3.0 中介绍了一些非常有趣的技术，包括 WPF，虽然 WCF 以及 WF 技术有点多余，但是.net3.0 中也介绍了一个非常有用的技术，这个技术就是 UI Autiomation（UIA）接口，一个顺从很多人期望的技术。UIA 是.net 中的一个库，通过这个库你可以极其方便地识别控件的 UI 元素，并获得这些 UI 元素的属性值，通过操作这些 UI 元素以及属性的值就可以实现 Windows 应用程序的人工自动化测试。虽然 Microsoft 用了很大的力气才压缩并生成这些 Windows UI 的接口。但这些 UIA 是基于 Windows 消息 API 接口的，调用它们需要用低级的 C 函数来实现。就像大家所想的那样，功能自动化测试如果可以使用方便的方法，就像调用.net 类库一样的话，是多么舒心的事啊。正如你所希望的那样，White 就是实现这种想法的最好方式。

White 支持 Wind32、WinForm、WPF 以及 SWT 应用程序的功能自动化测试。White 当然也能够处理 UIA 没有提供的一些 Windows 消息。但是使用这些方法取决于用户如何使用 White 提供的 API。White 不支持 Web 应用程序。

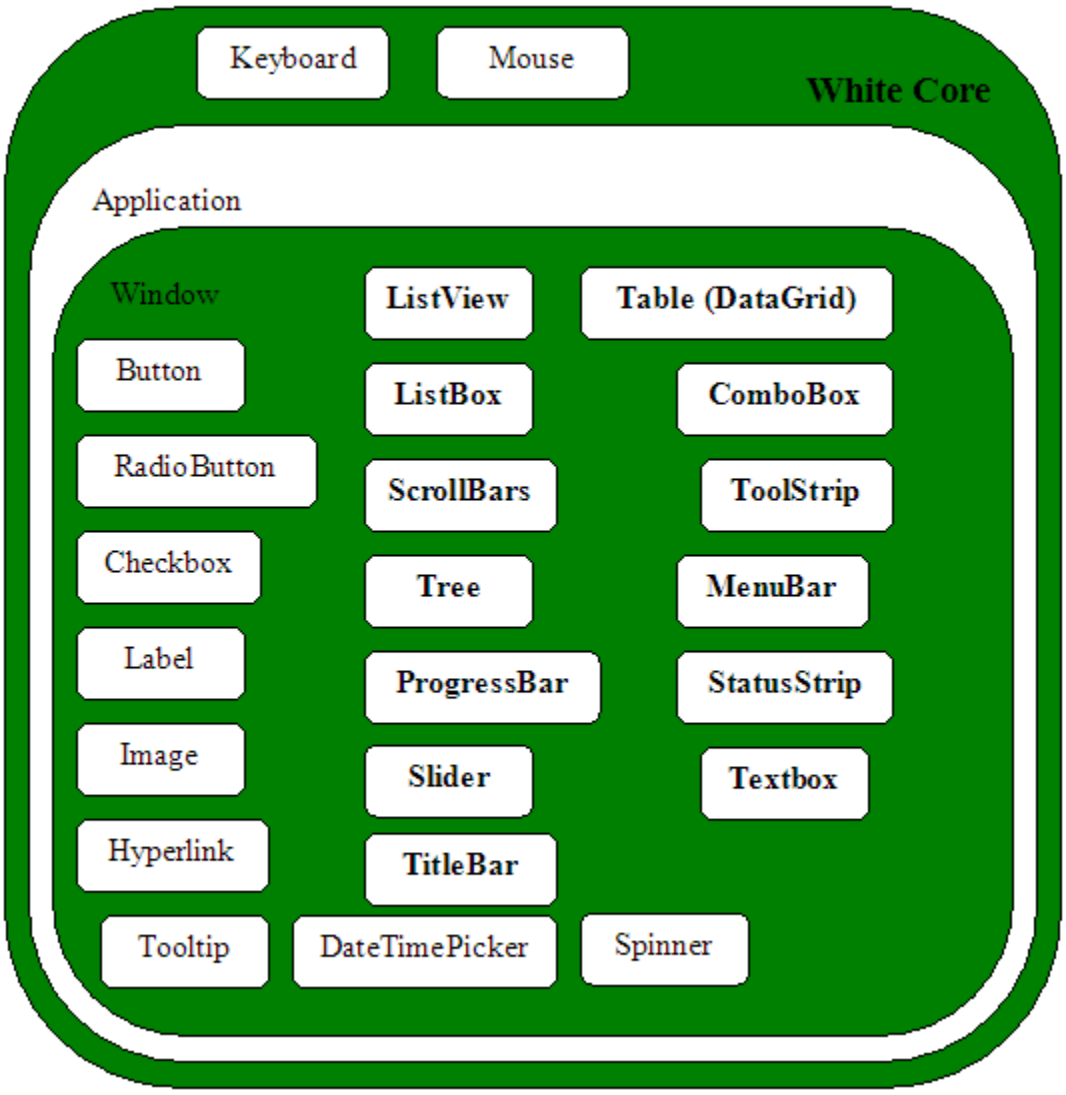
White 开发环境是使用 c#.net。

White 的工作层次：



批注：在你的工程文件中引用 White.Core 下的类，通过这些类的方法和属性的使用，就可以实现功能自动化测试。而 White.Core 封装的又是 Windows Messages 和 UIA 所提供的接口。这些接口是在 Windows Messages 和 Windows 操作系统的基础上才能够运行的。

White 封装的对象

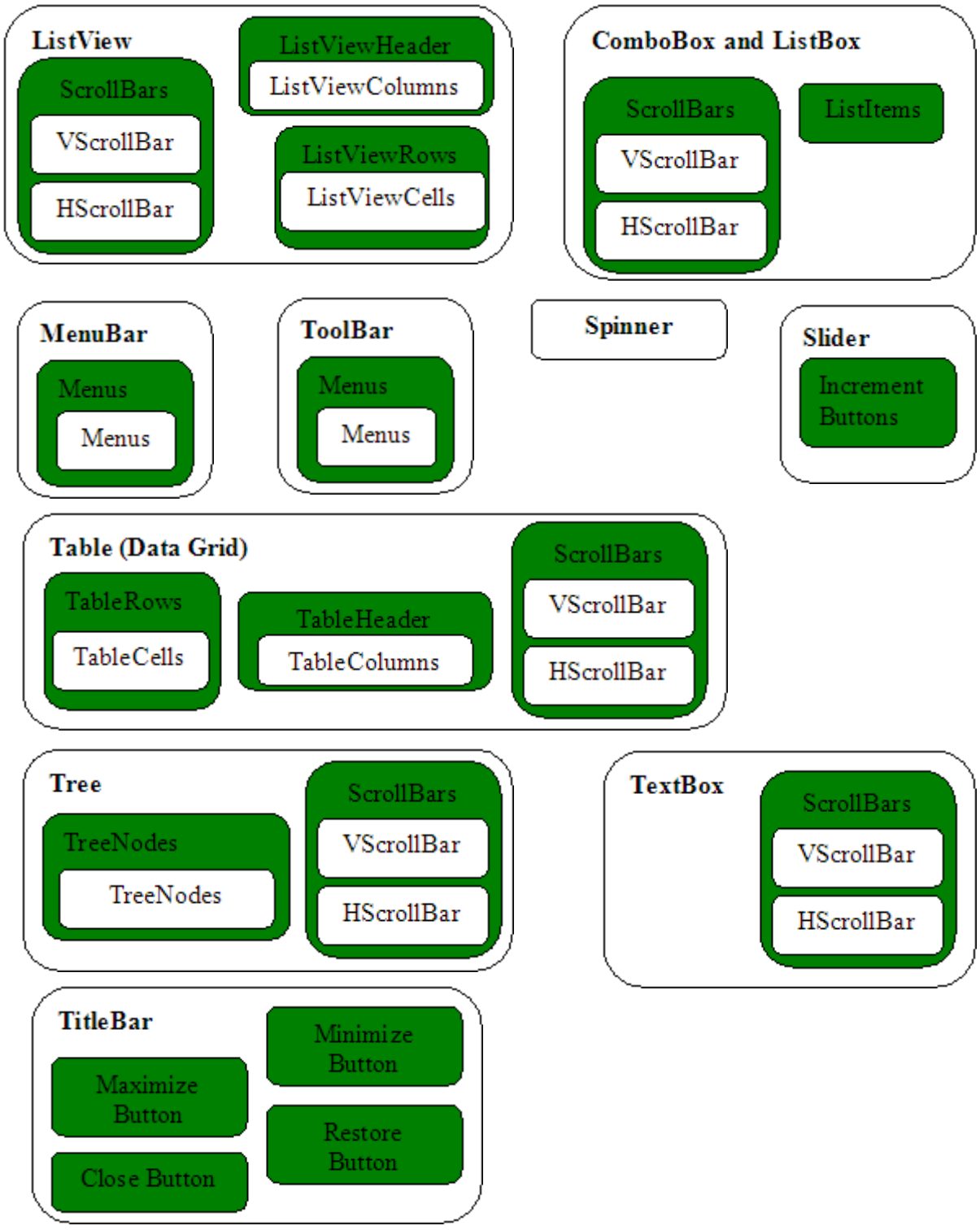


批注：White.Core 提供了 Keyboard 和 Mouse 类的方法，可以模拟鼠标和键盘操作。所以运行使用 White.Core 的程序时，会抢夺你的鼠标和键盘控制权。其实际就类似于 QTP 的回放时，鼠标和键盘的状态。这个图形，其实际告诉了我们 White.Core 使用的层关系。

首先，需要创建 Application 对象，通过 Application 对象再调用 Window 窗口对象，在 Window 窗口对象就可以使用 Button、RadioButton 这些控件对象来

进行测试程序的编写。上面每一个名称对应一个 Window 控件类型，这些控件类型都是标准 Window 窗口控件。要想从应用程序中获得等测试控件是那种类型，见后面的 UISpy.exe 介绍。

对一些对象的详细划分



批注：这是对一些对象中所拥有的子对象更详细的划分。比如 VScrollBar 和 HScrollBar，横纵滚动条等。TableCells 和 TableColumns，单元格和单元列等。在自动化测试时，我们可能需要对一些对象进行更详细的划分，以达到精细的操作。对象的名称需要比较了解，就像 QTP 中的对象一样，这是测试的前提。另外，这里所说的对象其实就是类的概念，熟悉这些类和控件的关系，这是需要加强的。

UIItem 的鉴定

自动化测试的难点之一就是如何获得那些待操纵控件的名称。要想使用 UIA 的 API 来操作这些控件，就必须得这些名称。开发环境中 Name 属性的值在 UIA 接口中是属性 AutomationId 的值。但是不是所有应用程序的控件都会去设置 Name 属性的值的，这时 white 可以使用其 text 属性的值来操作控件。

批注：原作者的意思，大概就是说使用 white 可以通过多个途径来获得这些控件的名称，并操纵这些控件。

使用方法：

环境：.NET 3 framework

获得窗口对象：

```
Application application = Application.Launch("foo.exe");//启动应用程序，注意这里如果不是当前文件夹，就得写全地址了。
Window window = application.GetWindow("bar", InitializeOption.NoCache);//获得这个窗口,”bar”是窗口的标题
```

White 是使用 UIA 的 API 去寻找一个窗口，并控制该窗口的。UIA 与可见的窗口进行 Windows Messages 通话，然后可始枚举这个窗口所有可控制属性，当然这些 UIItems 是非常多的。White 将会最优地从 UIItems 中寻找一个属性来进行控制。InitializeOption.NoCache 的意思是在初始时是否有缓存。

寻找一个 UIItems 并实现一个动作：

```
Button button = window.Get<Button>("save");
button.Click();
```

“save”是这个 button 的 AutomationId 的值。

使用 SearchCriteria 类中的方法寻找 UIItems：

SearchCriteria searchCriteria = SearchCriteria.ByAutomationId("name").AndControlType(typeof(TextBox)).AndIndex(2);//寻找第 2 个名称叫 name 的 TextBox 控件

TextBox textBox = (TextBox) window.Get(searchCriteria);//通过 SearchCriteria 对象来控制这个控件

textBox.Text = "Anil";//设置这个控件的 Text 属性值

通过 SearchCriteria 类中提供的方法可以获得任何 UIItem 属性的值。

a、AutomationId 属性的值取决于开发人员设置。这个属性支持 WinForm 和 WPF 程序，但是不支持 SWT 的应用程序

b、TestControlType 是 White.Core 的属性。

c、ControlType 需要先引用 UIAutomation 接口

d、Text 是有目的额外增加的属性。通过这个属性用来分辨其居有相同属性的控件，这时往往要配合使用 index 方法来达到识别惟一的效果。Index 是从 0 开始的。Index 是从左上，由 X 轴到 Y 轴进行寻找的。

White 如何快速地找到控件：

SearchCriteria 类中有多个方法来寻找 UItems，在寻找 UIItems 的寻找过程中会枚举窗口中所有 UItems，根据你所指定的方法 SearchCriteria 将会快速地、精确地找到这个控件。如何准确地、快速地寻找控件取决于你使用 SearchCriteria 方法中的内容。当你运行你的测试程序的时，White 就可以这些记录的属性快速的寻找到的 UIItems 在窗口中的位置。如果下次你应用程序更改了你使用 SearchCriteria 方法获得的属性值的内容，应用程序就将找不到该控件，当然测试程序也就不能够继续运行了。所以当应该程序的属性或者位置改变时一定要小心，要在测试程序中及时更新其相关内容。

通过这个方法可以处理窗口 Title 发生例外的情况

application.GetWindow("SomeTitle", InitializeOption.NoCache.AndIdentifiedBy("Login"));

这个方法只能使用于不同的窗口操纵相同文件的情况。

配置：

White 的运行是根据 App.config 文件来配置的。如何配置这文件中的节点的值，是需要知道的。下面的例子就演示了如何配置这些值的方法。

```
<configSections>
```

```
        <sectionGroup name="NUnit">
          <section name="TestRunner"
type="System.Configuration.NameValueSectionHandler"/>
          <section name="ProgramMode" type="Debug"/>
        </sectionGroup>
        <sectionGroup name="White">
          <section name="Core"
type="System.Configuration.NameValueSectionHandler"/>
        </sectionGroup>
      </configSections>

      <White>
        <Core>
          <add key="LogActions" value="false" />
          <add key="WorkSessionLocation" value="." />
          <add key="PopupTimeout" value="5000" />
          <add key="MouseDelayForTooltip" value="0" />
          <add key="SuggestionListTimeout" value="3000" />
          <add key="BusyTimeout" value="5000" />
          <add key="WaitBasedOnHouseGlass" value="true" />
          <add key="UIAutomationZeroWindowBugTimeout" value="5000" />
          <add key="TooltipWaitTime" value="3000" />
        </Core>
      </White>
```

所有属性的值都是配置好了的。下面的方法介绍了如何读取或更改这些属性的值。

```
Console.WriteLine(CoreAppXmlConfiguration.Instance.PopupTimeout);
//输出窗口弹出的超时时间
CoreAppXmlConfiguration.Instance.TooltipWaitTime = 100;//更改 ToolTip 弹出的时间为 100 毫秒
```

LogAction=”true”的意思是记录所有有关 UIItems 窗口的日志。

等待处理：

1、white 会自动等待窗口内容加载完全之后才做下一个动作。

2、等待的时候可以通过 WaitBasedOnHourGlass 属性来设置，默认是 true。当 White 遇到下面两种鼠标状态时就会等待其光标样子变成箭头型才继续下一个动作。

 或 

3、上面这状态在等待窗口加载完全是不够的。在测试程序中进行等待操作不同的条件，等待加载完全的方式也不同。BusyTimeOut 属性可以设置最大等待时间。当然还有一些其它属性的设置可以调节不同的等待时间。

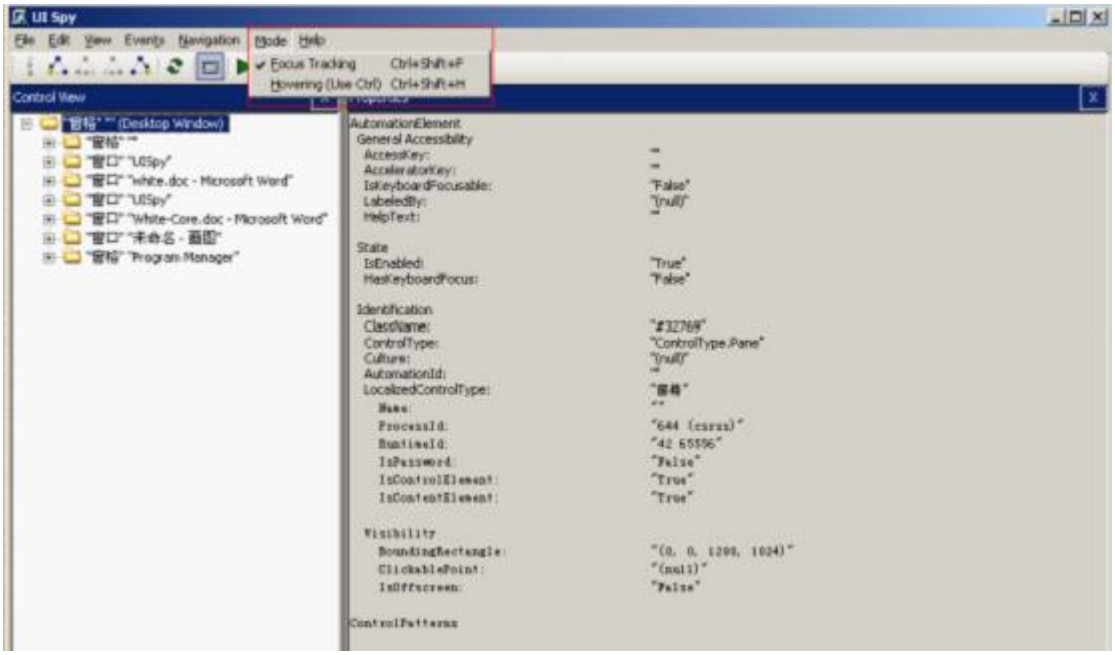
属性	描述
PopupTimeout	找到弹出框的时间
TooltipWaitTime	Tooltip 出现的时间
SuggestionListTimeout	suggestion list 出现的时间
UIAutomationZeroWindowBugTimeout	如果多久没有找到窗口就退出的时间。这里有一个 Bug，UIAutomation 有时可能不能通过进程来找到应用程序。

UISpy：

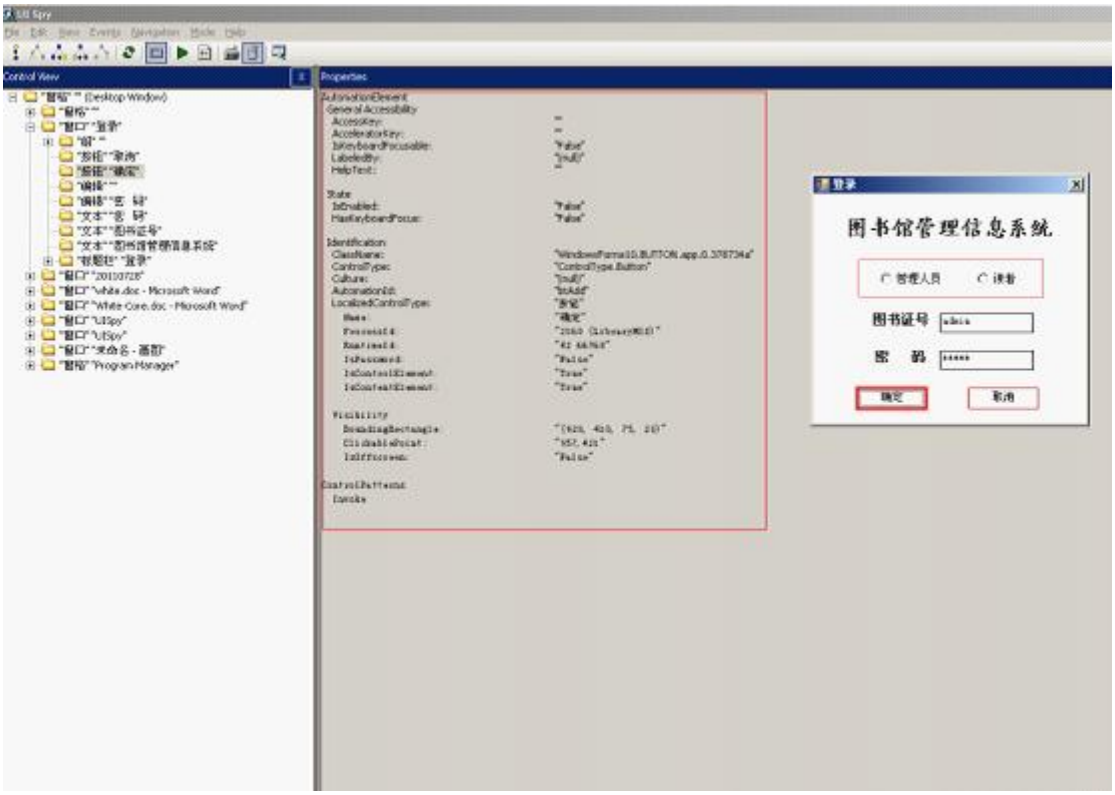
该工具对于从窗口中寻找到 UIItems 非常有用。这个工具从.Net3.0 SDK 中而来。该方法本来是需要在集成.net3.0 SDK 环境下使用的，但这里已经单独生成了一个 UISpy.exe 工具。

批注：原作者没有详细介绍该软件的使用方法。下面简单介绍一下，这个工具的使用方法。

运行后，有两种方式可供选择查找，一种是 focus tracking，即光标定位到那儿，它就找那儿的控件的属性；另一种是 Hovering(Use ctrl)，即按在某个控制上按 ctrl 时所找到的控件属性。

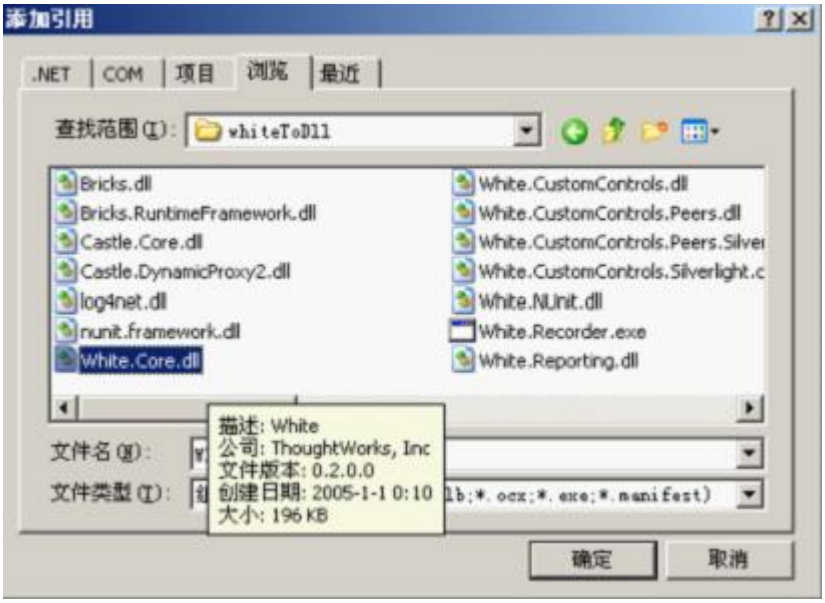


注意 AutomationElement 节点下所有的属性都是可以使用的。但并不是所有属性 White.Core 都提供了方法来进行访问。



举一个简单的实例：

1、新建一个工程后，引用 White.Core，如下图所示



2、引用以下命名空间

using White.Core; //引用 white 组件

using White.Core.UIItems;//GUI 自动化测试对象类型

using White.Core.UIItems.Finders;//GUI 自动化测试对象类型的寻找方法

3、启动应用程序，创建一个对象并将此对象附加到该窗口

Application calCapp = Application.Launch("calc.exe");//启动计算器，这里可以是文件的全部路径

calCapp = Application.Attach("calc");//附加到一个已经存在的窗口，传的是进程名称

Window calcwindow = calCapp.GetWindow("计算器",InitializeOption.NoCache);//创建一个 window 对象，以操纵其子控件

4、获取子对象，实现测试

Button two = calcwindow.Get<Button>(SearchCriteria.ByText("2"));//寻找 2 这个按钮

two.Click();//找到 2 这个按钮之后，点击一下

Button plus = calcwindow.Get<Button>(SearchCriteria.ByText("+"));

plus.Click();

Button three = calcwindow.Get<Button>(SearchCriteria.ByText("3"));

three.Click();

Button equeal = calcwindow.Get<Button>(SearchCriteria.ByText("="));

equeal.Click();

//这个 id 值，是通过 UISpy 获得的，不同的电脑其值不同。

TextBox output = calcwindow.Get<TextBox>(SearchCriteria.ByAutomationId("403"));

string outputValue = "";

outputValue = output.Text;

Console.WriteLine(outputValue);

if (outputValue.Contains("5"))

{

Console.WriteLine("It's right!");

}

Console.ReadLine();

本文是基于作者的源文件中的 White-Core.doc 翻译再加一些注释和小例子而来的。由于我的英文水平有限，原作者的意图可能有些地方没有传达到，还望

大家凉解。White 是一个开源的自动化测试工具，这里翻译的内容也只是大海中的一滴，更多详细的内容还得大家去仔细研究，然后才能很好的使用。

我的测试之路

作者：刘红霞

进入测试已经五个年头了，感觉这个行业还是比较适合自己的，在这个道路上我还有很长的路要走，在此先和大家分享下我的五年测试历程。

职业道路选择------认准目标就前进

我最开始接触测试这行是在 2005 年，还算比较早的，但是那时，我对测试的理解就是要找问题，也不会去深究，对测试没有一个完整的概念，以为测试就是不会写代码的人都可以做的。也没有意识去思考测试项目流程是什么，项目的架构是怎样的，要采用哪种数据库、编程语言，采用什么协议，设计思路是怎样的？

测试了整整一年后，我也只知道按照不是特别规范的测试用例来执行，加上测试进度很紧，当时测了一轮又一轮。那个时期，国内的测试还比较薄弱，公司普遍都还不是很重视测试，我自己也不知道怎样才算把这个工作做好，在网上查阅了相关的资料，但是关于这方面的资料特别少，自己抽空学习也没啥效果。

后来，我觉得测试前景还是不错的，很想对测试这个行业有一个整体的认识，另外，我也想多了解下自动化测试工具，希望这些能让我的测试之路走的更远更好，于是我选择了自我充电，参加了上海一个测试培训。

在这个过程中，因为学习的欲望很强，很多都是我主动想学的，所以我边学边实践。通过学习，我了解了测试的基本概念、基本流程；测试在整个软件周期中的作用；测试用例的编写，方案的编写；数据库基本应用等。现在回想起来，这段时间是一个学习的美好的回忆。我最大的收获就是明确了以后在测试工作中，我应该关注哪些方面、从哪些方面去思考问题、怎样使我的工作做的更好。

心态的作用------一切从工作出发

心态好才能工作好，这句话很对，在测试过程中，可能你会做很多重复的活，但是你怎样才能保证你自己工作积极性一直很高，怎样才能在工作中获取自信呢？在工作当中，我是这样做的：

1、对不理解或困难的工作，我自己去查找资料或问同事，寻求帮助。

2、对自己会做的工作，我在能够完成的同时，我会留出一定的时间，来自己做自由测试，这块很重要，因为很多测试用例覆盖到的地方，基本都测试过，测试是不能穷尽的，可能有些路径或者操作，只有在自由测试中才能找出。

3、如果同事有需要帮助的地方，我会尽力解答，不会的参照第一点，总之，要把自己融入到团队中来。

4、经验的积累。做测试一定要求你平时工作比较心细，最好有比较好的记忆力，然后及时总结，比如你测试过一个地方，有问题，下次你就会很注意，有类似的场景时，你肯定会测试这个点，这样能快速发现问题。

5、站在客户角度去思考。当完成你的基本测试时，要思考下，如果我用这个软件，哪里会觉得不舒服或需要改进的，这样你再和开发沟通，觉得这样做或许更好，特别是有些工具的软件，一定要考虑到用的实际场景和流程。

6、站在测试的角度去思考，这个有些专业知识在里面，你应该思考下设计的思路是否合理，碰到一些异常的问题，软件是否能够处理？越是底层设计的问题，越早发现更好。

总之，如果一个好的测试人员，最后可以做到测试驱动开发，这样你就不会觉得自己的工作重复与烦心了。自己在测试当中，总保持一个怀疑的态度和好奇的心，来对待我要测试的软件，这样才能不断的会有新的问题出来，不过，我们的开发人员可就要郁闷了。认识我的开发人员都很喜欢我来测试他们的东西，因为他们也都希望问题尽量的提早发现，在这一点上，我们能够达成共识，这个也成为了我工作的动力。

专业知识的填充--不断地学习新知识

从测试角度来说，作为一个测试人员，应该掌握比较广泛的知识，比如测试架构：C/S，B/S，测试基本理论，测试的流程，应用的数据库，某些业务知识（如通讯类的，证券，基金等），网络知识，操作系统，测试管理工具，缺陷用例，缺陷报告编写等，嵌入式软件还要熟悉一些硬件的操作以及相关的知识。总之，一句话：书到用时方恨少。

我目前做的基本是手工测试，我比较佩服那些开发人员，因为我不太会写代码，要用的自动化工具，对功能测试 QTP，性能测试 LR 都要熟练，可惜，我一直没有跨进这个门槛。目前工作中有用到这个，正在训练学习中。

对于编程语言的熟悉这一块，以前我们一个测试经理说，一个测试人员，不懂代码就像人残废一样，虽然话有点难听，但是熟悉开发的思路和代码，会让你的测试技术之路走的更宽，更长！

在技术领域里，你知识越渊博，越被人喜欢，因为你就是一个活字典！.

除了专业知识，对软件需求也要深入了解。需求是开发编写软件的源头，不管你是作为普通的测试人员，还是资深的测试人员，都需要对你所测的软件的需求有很好的了解，包括产品需求和测试需求。这个也是一个过程，最开始可以去了解需求的关注项，如：功能，性能，接口，属性，约束条件等。然后由浅入深，理解显性需求和挖掘隐性需求。特别要关注开发实现时，是否考虑到异常输入和输出。

开发与测试关系处理--换位思考

在整个项目中，其实开发和测试是一个团队，团队的目标是一致的，提高软件的质量。但是工作当中因为职责的不一样，往往可能会造成分歧。为了更好的配合开发，测试人员要把握好以下几点：

1、在提交问题时，表达要清楚，重现步骤和预期结果要清楚。

2、如果是概率性出现的问题，最好记录好有用的日志并保持现场，这样能帮助开发更快解决问题，必要时，要协助开发重现问题。

3、在提交问题单时，可以先把严重的问题现象，步骤告诉开发，然后再提单。如果问题较多时，先提严重的，小问题最后再提，因为开发也有绩效考核的，开发修改问题的效率高了，这样开发会很乐意和你合作。

4、有些有争执的问题或可改可不改的问题，和开发讨论没有结果，但是测试觉得实在是改了更好，可以找上一级或者专家协商确定后，再提单，或者告诉开发兄弟，这个问题可以不用马上改，优先级很低，要改了这个软件更好，更能体现开发的能力等。

5、把开发当你朋友。每当我测试到一个很严重的问题时，我会找开发聊，这个问题是怎样产生的，你是怎样解决的？然后会问开发人员，你这样解决之后会不会产生其他的问题？然后会跟开发人员说，以你的能力你肯定能解决这个问题的，相信你！这样会增加开发对你信任，也说明你和他是站在一起的。

6、最后一点是，很简单的，开发人员和测试人员多交流沟通，搞下团队活动，增进合作默契！

写在后面的话：

这个就是我在从事测试这一行业的点滴经验，说实话，虽然在工作中有时候也会遇到困惑，而且掌握的测试技术也还只是皮毛，不过总体来说工作还是很开心的，我也会一直在测试这行走下去，希望我以上的文字对想进入软件测试或刚进入软件测试这行的朋友有些帮助。

使用 Mobi Ready 进行手机软件兼容性测试

作者：王晓芹

移动手机及移动互联网正在日新月异的发展，但是很多网站用手机访问时却往往难以定位，很多时候，我们会发现一些页面内容不可访问。也就是说能用电脑所访问的一些网站却无法支持手机访问。而 Mobi Ready 测试工具则可以帮助用户验证某个站点或页面是否适合在手持设备上显示。

一、工具使用

Mobi Ready 是爱尔兰 DOTMOBILE 公司所开发的一款在线测试工具。打开在线测试页面 http://ready.mobi/launch.jsp?locale=en_EN，只需要在指定的输入框中输入要测试的页面 URL，提交后就可以返回测试报告，这种情况下用户使用的是默认配置。另外，可以通过 More Options 按钮得到自定义测试配置界面，目前配置界面中提供了 6 种选项。

- Choose a device
- SonyEricssonK750i
- Nokia6680
- SAMSUNG-SGH-T809
- Motorola RAZR
- W3C mobileOk DDC，

其中第一个选项即为工具的默认配置；1~4 选项分别为手机类型，第 5 个选项表示按照 W3 C 组织所定义的缺省提交上下文进行配置（关于 W3C mobileOk DDC 详细内容可参见 http://www.w3.org/TR/mobile-bp/#ddc）。

二、工具原理

1、HTTP 参数

这个工具的工作原理是在获取页面时使用移动终端代理字符串进行识别，这可以保证站点可以返回基本符合格式的响应信息。如当在自定义配置"Select a device"一栏中选择"SonyEricssonK750i"发送的参数如下：

User Agent: "SonyEricssonK750i/R1L Browser/SEMC-Browser/4.2 Profile/MIDP-2.0 Configuration/CLDC-1.1"

Accept Types: "application/xhtml+xml, text/css, image/jpeg, image/gif"

2、下载时间：

报告中的下载时间只是一种估计，这种估计基于所支持的每种网络连接方式下的理论带宽以及实际经验，所使用的依据如下：

GPRS：33Kb/s，有 0.5s 的延迟

3G：128kb/s,带有 0.1s 的延迟

WIFI：11000Kb/s，带有 0.1s 的延迟

以上数据不可能代表所有的情况，在某些情况下，数据可能会高，也可能会低。

3、成本

成本估计是基于每个地区移动用户的平均花费。目前各地区移动用户收费都以“每 Kb”为单位进行收费。这个费用会经常改变，费用按照消费水平的不同具有不同的费率，它由基本消费费用和浮动费用构成。费率方式增多后，成本估计变得复杂，为了简化这个过程，尽可能统一使用“每 kb”单位来计算。

三、应用实例

下面以百度页面 http://www.baidu.com 为例，应用 MobiReady 工具进行在线测试，在输入完页面 URL 地址后，工具很快返回一个测试报告。报告共分四大部分。下面对该报告的主要部分进行介绍：

1、总述

报告第一部分主要说明了该网页/网站在手机上的显示的总体情况。对于本测试，报告显示如下：

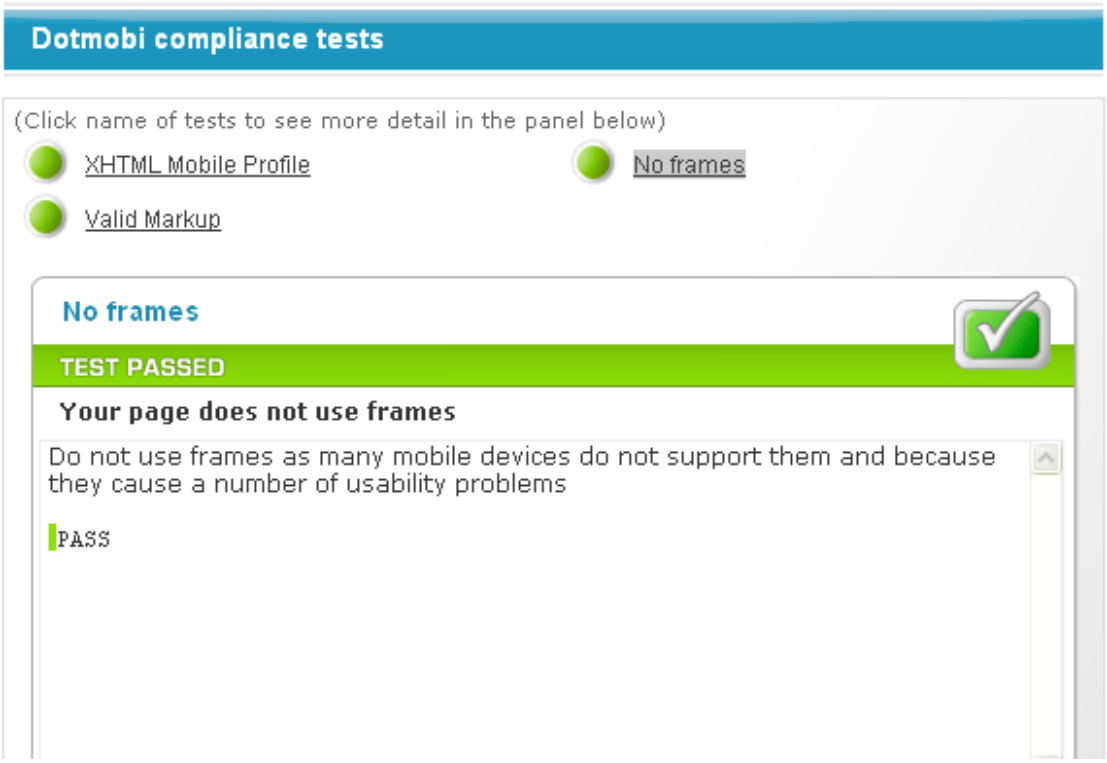
"It will probably display very well on a mobile phone."表明该页面在移动手机上可能会显示很好。

另外，总述部分还提供了柱形图，分别从可读性、页面大小、预期成本、预期速度四个方面展示了测试结果。

图 1 测试结果总体情况

从图示可以看出，本页面的得分为 4.86（满分为 5），分值还是比较高的。页面大小为 5.94K。在所展示的澳大利亚、中国、欧州、美国等几个区域中，中国的预期成本最低。预期速度部分则分别列出了 WIFI，3G 和 GPRS 三种网络连接速度所需时间，其中 WIFI 所需耗时最小，仅 1.01s。

2、符合性测试



符合性测试结果列出了针对 XHTML Mobile profile，valid markup 及 NO frames 三种类型规约的符合性情况，单击以上三种规约名称可以查看更详细的测试结果。比如，单击 NO frames，页面上所展示的是测试页面没有使用 frames，在面板下面的“详细说明”中明确说明了“因为很多移动设备不支持 frame，可能会导致可用性问题，所以不要使用 frames。”该测试项为“通过”。

3、其它测试

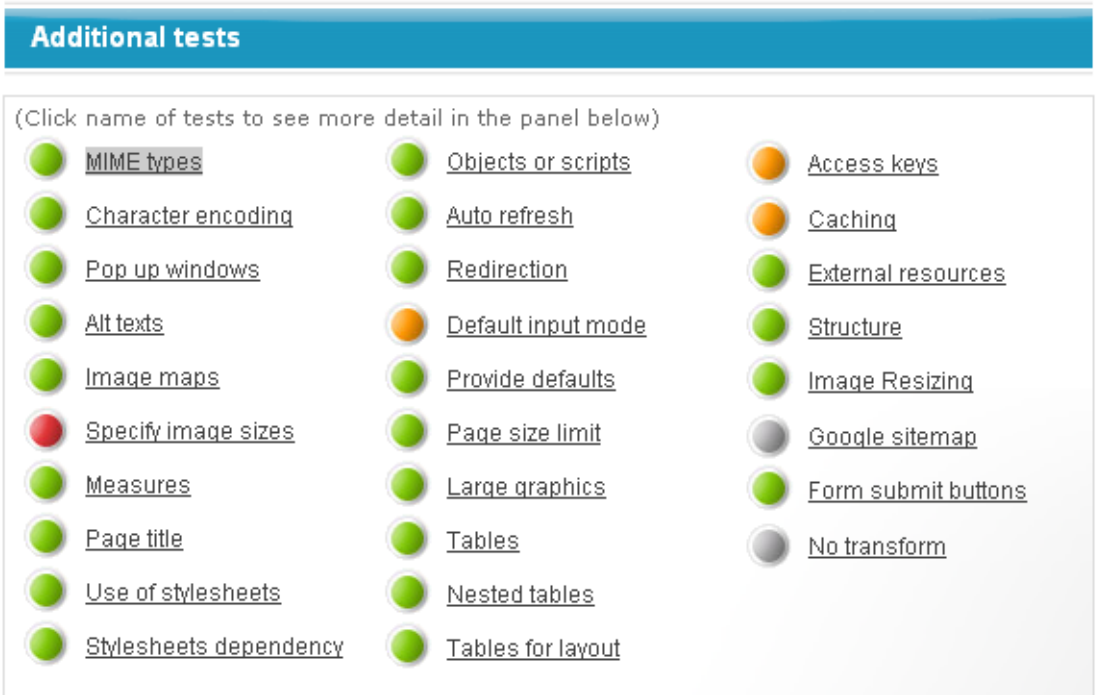


图 2 其它测试名称列表

“其它测试”列出了具体测试项，如图 2 所示，测试名称前面的按钮颜色对应测试结果类型，绿色代表了“通过”，桔黄色代表“警告”，红色代表“失败”，灰色代表“建议”。单击每一个测试项名称可以看到详细内容。

例如单击“Specify image sizes”可以看到提示“At least one image did not specify width or height”。窗口右上角有“help me fix it”字样，单击该文本链接，在弹出的页面中可以看到如下修改建议：

"These attributes must be expressed as pixel (px) values. If they are not, this test will fail:"

错误：

正确：

通过以上提示，就可以如何修改错误的内容了。

4、Page markup 及 Http Reponse Header

报告的最后展示了 Page markup 及 Http Reponse Header 两部分内容，分别为服务器针对移动终端请求返回的内容，有助于软件开发人员对代码进行错误查找及修改。

Mantis 深入学习

作者：鹿鸣

一、mantis 的配置和开发环境

Mantis 我用的 1.2.5 版，建议对 mantis 感兴趣的，都看看 doc 目录下的 administration_guide 和 developers 两个文档，自己试验里面的参数和功能，对 mantis 的理解能加深不少。当然了，即使不深入了解，直接使用 mantis 也不会有什么问题。

二、mantis 结构分析

下面的内容，仅仅代表本人的一些看法，可能有不对的地方，大家可以随时指出，谢谢。

Mantis 的目录和文件很多，根目录下面的 php 文件主要都是功能页面，core 目录中是需要的各种 api 函数文件，因为汉化的缘故，lang 目录的 strings_chinese_simplified.txt 文件也是我们关心的内容。

其实感觉 mantis 的结构安排不尽合理，根目录下面的大部分 php 文件，都应该放置到一个专门的目录，因为都是一些功能页面文件。而用户定义的内容，比如 config_inc.php 等，才应该放置在根目录或者专门的配置目录中，现在的安排显得很混乱，主次不清。

如果看过 mantis 的源码，会发现很多源码文件中都首先引入

```
require_once( 'core.php' );
```

core.php 是我们第一个需要分析的文件，把此文件分析完，应该如何让 mantis 按照自己的心意修改，我们也就了解了。

三、core.php 到底做了什么

下面的分析，我没有解释全部代码，只是挑选主要的说明 mantis 都在后面做了哪些魔法，感兴趣的可以直接去看完整的代码。

1、constant_inc.php

在 core.php 文件第 45 行：

```
require_once( dirname( __FILE__ ).DIRECTORY_SEPARATOR.'core'.DIRECTOR
Y_SEPARATOR.'constant_inc.php' );
```

在 constant_inc.php 文件中定义了 mantis 中使用的各种常量。

2、custom_constants_inc.php

第 50、51 行：

```
if
( file_exists( dirname( __FILE__ ).DIRECTORY_SEPARATOR.'custom_constants_inc.php' ) ) {
    require_once( dirname( __FILE__ ).DIRECTORY_SEPARATOR.'custom_constants_inc.php' );}
```

缺省安装 mantis 后，并没有 custom_constants_inc.php 文件，此文件在需要的时候由用户手工创建，主要是 define 用户自定义的常量值。按照基本的原则，没有绝对必要，不要修改 mantis 源码，mantis 本身已经给你留下了接口文件，此 custom_constants_inc.php 文件就是在 custom_constants_inc.php 中，不能用 define 定义在 constant_inc.php 已经定义过的常量。如果必须要修改，请在 constant_inc.php 中直接修改。

3、config_defaults_inc.php

第 62 行：

```
require_once( dirname( __FILE__ ).DIRECTORY_SEPARATOR.'config_defaults_inc.php' );
```

config_defaults_inc.php 文件是 mantis 系统默认的参数配置文件。此文件和下面 config_inc.php 作用相同，但是 config_defaults_inc.php 是系统预设的，config_inc.php 是用户自己定义的。

4、config_inc.php

第 64-68 行：

```
# config_inc may not be present if this is a new install
if
( file_exists( dirname( __FILE__ ).DIRECTORY_SEPARATOR.'config_inc.php' ) )
{
    require_once( dirname( __FILE__ ).DIRECTORY_SEPARATOR.'config_inc.php' );
    $t_config_inc_found = true;
}
```

在安装或升级 mantis 的时候，mantis 会自动把数据库连接等信息写到此 config_inc.php 文件中。用户需要自定义配置，也都是写入参数变量在此文件中。可以写入的配置变量参考 doc 目录内 administration_guide 的 Chapter 5. Configuration 章节。在根目录的 config_inc.php.sample 文件内也有一些相关的示例。总之，需要用户自定义的参数，就在 config_inc.php 中实现。

如果 config_inc.php 和 config_defaults_inc.php 参数有重复的情况，以 config_inc.php 中定义的为准。

5、custom_functions_inc.php

第 245-248 行：

```
    $t_overrides = dirname( __FILE__ ) . DIRECTORY_SEPARATOR .
'custom_functions_inc.php';
    if ( file_exists( $t_overrides ) ) {
        require_once( $t_overrides );
}
```

用户自定义的函数，请写入到此 custom_functions_inc.php 文件中。mantis 安装后缺省没有此文件，需要的时候由用户手工创建。

6、其他

在 core.php 文件里面还定义了很多内容，比如载入 core 目录下面的各种 api 文件，定义时区，出否加载 wiki 等，但是和用户配置密切相关的，主要还是上面的 5 个文件。

7、总结

core.php 文件主要负责加载 mantis 使用的资源和库文件，而且预留了用户的接口，主要就是变量 config_inc.php、常量 custom_constants_inc.php、函数 custom_functions_inc.php，后两个文件在 mantis 默认安装后不存在，需要的时候由用户手工创建。在用户配置或开发的过程中，尽量不要修改 mantis 的原文件，优先使用此 3 个接口文件。

具体的示例，可以参考 doc 目录内 administration_guide 的 Chapter 7. Customizing mantisBT。

四、在 config_inc.php 中自定义配置

上面的是分析，下面开始实践了，分两个部分，第一部分是通过修改 config_inc.php 配置文件，根据实际情况自定义需要的功能；第二部分是通过编写一个代码示例实现需要的功能。

以下内容没有特殊说明，都在 config_inc.php 中定义。

1、国际化

在 config_inc.php 中，如果通过 install.php 安装，默认就有数据库相关的信息。

比如我的就是：

```
$g_hostname = 'localhost';
$g_db_type = 'mysql';
$g_database_name = 'mantis';
$g_db_username = 'mantis';
$g_db_password = 'mantis';
```

接着把 mantis 的默认语言设置为中文：

```
$g_default_language = 'chinese_simplified';
```

$g_default_language 是很关键的参数变量，看源码就能清楚，在 mantis 中显示字段内容的时候，都是用 lang_get()函数，lang_get 函数会根据你设定的默认语言，找 lang 目录下对应的语言文件，比如$g_default_language 设定为'chinese_simplified'后，那么只要遇到 lang_get()，就找 lang/strings_chinese_simplified.txt 文件中对应的变量，界面就相应显示为相应中文。

2、新建角色

不知道为什么，mantis 缺省没有测试人员角色，只有报告员角色，但是报告员角色的权限比开发员低，也就是说，如果测试员设定为报告员角色，开发员因为权限更高，可以处理测试员的缺陷状态，比如直接关闭缺陷，我认为既然 mantis 主要给测试人员管理缺陷用的，那么测试人员的角色权限就应该在开发人员之上。

根据以上说明，创建新的测试人员角色。还有给匿名用户新建一个匿名用户权限。

在 config_inc.php 中，加入

```
$g_access_levels_enum_string = '5:匿名用户,10:复查员,25:报告员,40:修改员,55:开发人员,60:测试人员,70:经理,90:管理员';
```

在 lang 目录的 strings_chinese_simplified.txt 中，查找并修改

```
$s_access_levels_enum_string = '5:匿名用户,10:复查员,25:报告员,40:修改员,55:开发人员,60:测试人员,70:经理,90:管理员';
```

看清楚了，一个是$g_，一个是$s_，变量是不一样的。

在根目录新建 custom_constants_inc.php 文件，里面写入：

```
<?php
    define( 'ANONYMOUS', 5 );
    define( 'TESTER', 60 );
?>
```

custom_constants_inc.php 其实不设定也不影响使用，但是如果需要修改角色相关的代码，最好按照 mantis 的习惯使用定义的常量，而不是常数。

分析一下，其实在 config_defaults_inc.php 中，存在系统预设的 $g_access_levels_enum_string 变量，所以我们在 config_inc.php 中用自定义的 $g_access_levels_enum_string 变量替换系统预设值。这样在使用过程中，就有了新的匿名用户角色和测试人员角色。

但是修改后，其实只相当于添加了 5 和 60 的角色，但是对应的默认语言（这里是中文）角色名称在 lang/strings_chinese_simplified.txt 的 $s_access_levels_enum_string 变量中没有定义，所以在使用过程中，就会出现@5@这样的角色。需要在相应的 lang 目录国际化文件中，对新添加的内容也定义相应的翻译。所有在程序中，遇到类似@60@这样的内容，都是因为变量没有定义相应的国际化语言名称。

另外，mantis 设计的时候，编号不是随意的，高数值的很多情况下包含低数值的权限。比如 10 有提交缺陷的权限，那么 10 以上的其他角色，同样具备此权限，所以设计的时候要小心，一定要按照权限的高低进行排序设置。

3、严重性

Mantis 预设的严重性项条目很多，感觉没有必要，按照我们公司的实际需要，定义四个就足够了。

在 config_inc.php 中定义：

```
$g_severity_enum_string = '30:建议,40:轻微缺陷,50:一般缺陷,60:严重缺陷';
```

在 lang/strings_chinese_simplified.txt 中，修改：

```
$s_severity_enum_string = '30:建议,40:轻微缺陷,50:一般缺陷,60:严重缺陷';
```

原有系统，50 是默认的严重性，如果需要修改，可以在 config_inc.php 中定义：

```
$g_default_bug_severity = 40;
```

这样默认的严重性就是轻微缺陷。

4、缺陷发生频率

Mantis 中默认的缺陷发生频率是 70-REPRODUCIBILITY_HAVENOTTRIED（没有试验），我们把他默认设置为 10-总是出现。

```
$g_default_bug_reproducibility = REPRODUCIBILITY_ALWAYS;
```

其中 REPRODUCIBILITY_ALWAYS 是在 core/constant_inc.php 中定义的常量，其实就是常数 10。

5、缺陷状态

和上面的处理方法类似，在 config_inc.php 中定义：

```
$g_ status_enum_string = '10:新缺陷,30:暂不修改,40:不是缺陷,50:再次出现,80:已经修改,90:关闭';
```

在 lang/strings_chinese_simplified.txt 中修改：

```
$s_status_enum_string = '10:新缺陷,30:暂不修改,40:不是缺陷,50:再次出现,80:已经修改,90:关闭';
```

但是这么修改后，出现了一个问题，缺陷列表没有了缺陷状态对应的颜色，因为在 config_defaults_inc.php 文件内$g_status_colors 数组中定义的键值和自定义的键值不同。

在 config_inc.php 中重新定义颜色键值：

```
$g_status_colors = array( '新缺陷' => '#fcbdbd',  '暂不修改' => '#ffcd85',  '不是缺陷' => '#fff494', '再次出现' => '#c2dfff', '已经修改' => '#d2f5b0', '关闭'  => '#c9ccc4');
```

6、工作流

缺陷状态定义完毕后，定义工作流，可以使用管理员登陆，在管理->配置管理中定义相关的自定义字段、用户权限、工作流等内容。

其实在管理->配置管理->配置报告中，显示的就是已经定义的内容。愿意的话，你也可以直接在 config_inc.php 中定义，效果是相同的。

比如直接在 config_inc.php 中定义工作流：

```
$g_status_enum_workflow    = array (
   10 => '80:已经修改,30:暂不修改,40:不是缺陷',
   30 => '50:再次出现',
   40 => '90:关闭,50:再次出现',
   50 => '80:已经修改,30:暂不修改,40:不是缺陷',
   60 => '90:关闭,50:再次出现',
   80 => '90:关闭,50:再次出现',
   90 => '50:再次出现',
);
```

定义角色对缺陷状态的处理权限：

```
$g_set_status_threshold = array (30 => 55,40 => 55,50 => 60,80 => 55,90 => 60,);
```

还有很多，不详细说了，感兴趣的自己去看吧，反正就是需要，mantis 后台配置的内容也可以在 config_inc.php 中直接定义。

7、其他

下面是其他一些在 config_inc.php 中可以定义的变量。

其实大家未必需要设置这么多的内容，我是因为实验各种参数，很多觉得有用的就直接留下来了。

```
$g_complete_date_format='Y-m-d H:i:s';   //日期格式
$g_delete_bug_threshold = MANAGER;       //  删除缺陷的角色权限
$g_max_file_size = 5000000;   //最大的附件容量，这个同时需要在 php.ini 中设置
$g_view_summary_threshold = VIEWER;         //允许查看统计角色
$g_show_realname = ON;         //在页面显示用户的姓名而不是帐号
$g_preview_attachments_inline_max_size = 2000000;     //支持预览用图片格式最大附件尺寸
$g_allow_anonymous_login = ON;                  //是否支持匿名用户
$g_anonymous_account = 'anonymous';         //匿名用户的帐号，匿名登录后，为此帐号的权限
```

```
$g_show_timer = ON;              //在页面最下方，显示页面的载入时间
$g_default_home_page = 'view_all_bug_page.php'; //登录后默认进入查看问题页面
$g_enable_profiles = OFF;    //关闭平台配置，如果此项为 on，那么提交缺陷的时候，就有一个平台配置供你选择
$g_allow_file_upload    = ON;    //下面的三项都是上传附件相关的，你可以不把附件保存在数据库中，而是保存在本地磁盘目录上
$g_file_upload_method = DISK;
$g_absolute_path_default_upload_folder = 'C://apache//www//mantis//upload//';
$g_show_project_menu_bar = ON;   //在页面上方，显示项目直达链接
```

五、开发版本默认值功能

上一章用户定义的内容，都是通过系统自带的变量接口进行配置。但是一些内容，没有自定义的接口，怎么办？

好在 mantis 是开源软件，自己动手修改吧。

在提交缺陷的时候，有产品版本和目标版本，在处理缺陷的时候，有修正版本，这些版本都没有预设值，都需要自己手工选择版本，我们下一步就是通过自定义开发，给这些版本加上默认值。

1、mantis 中的版本

在后台管理版本的时候，版本有两个选项：已发布、已过期。

而 mantis 中使用版本的地方有三个：产品版本、目标版本、修正版本。

我们需要考虑的是：mantis 为什么这么设定，这么设定有什么意义。

下面是我认为的三个版本的含义：

产品版本：当前测试软件的版本。

目标版本：预期解决缺陷的版本。

修正版本：真正解决缺陷的版本。

其中产品版本一定是已经发布的，目标版本是没有发布的，修正版本有可能发布、也有可能没有发布。

在 mantis 中有两个功能模块：

变更日志（Changelog）：记录项目在各个不同的版本已经处理的缺陷。根据修正版本进行记录，只记录最终状态为已解决和已关闭的缺陷。

路线图（Roadmap）：在预期的目标版本是否真正处理了缺陷，给一个项目已经处理缺陷的百分比。根据目标版本进行分类，最终状态为已解决和已关闭的缺陷才计算为已经处理的缺陷。

变更日志和路线图功能需要使用的时候，都必须设定相应版本。

例如：一个问题，发现在 1.2 版本，目标 1.4 版本，关闭在 1.5 版本。相应的变更日志中，在 1.5 版本就记录一条状态为关闭的记录；在路线图中，1.4 版会记录此问题关闭并且算已解决问题。

所以说，mantis 之所以设计为三个版本，是为了相关的功能需要。

如果大家认为不需要使用变更日志和路线图功能，三种版本没有必要全部填写。

2、通过编程实现版本默认值功能

通过上一节的分析，我们了解了不同版本的差别，那么就可以确定，产品版本，是已经发布的版本，目标版本和修正版本，应该是没有发布的版本（也有可能是已发布版本版本，请手工选择即可，不在自动化提取默认值功能代码考虑范围之内）。根据版本是否发布，我们从版本列表中取得已发布或未发布的第一个版本作为相应版本的默认值。

在根目录创建 custom_functions_inc.php 文件，在里面写入自定义的获取版本函数，$project_id 参数是项目的 id，$t_released 参数确定版本是否已发布，此自定义函数取得版本列表中遇到的已发布或未发布的第一个版本作为函数返回值，代码如下：

```
<?php
    function custom_version($project_id,    $t_released= VERSION_ALL) {
     foreach (version_get_all_rows( $project_id,    $t_released) as $t_version){
        $t_custom_version = $t_version['version'];
        break;
     }
     return $t_custom_version;
    }
?>
```

在根目录的 bug_report_page.php 中，找到缺省的设定版本的地方，产品版本在 355 行，修改为：

```
<?php print_version_option_list( custom_version($t_project_id,
VERSION_RELEASED), $t_project_id, $t_product_version_released_mask ) ?>
```

目标版本在 395 行，修改为：

```
<?php print_version_option_list(custom_version($t_project_id,
VERSION_FUTURE)) ?>
```

根目录的 bug_change_status_page.php 文件 265-276 行修改为：

```
    <!-- Fixed in Version -->
    <?php $t_fixed_in_version = bug_get_field( $f_bug_id, 'fixed_in_version' ); ?>
    <tr <?php echo helper_alternate_class() ?>>
        <td class="category">
            <?php echo lang_get( 'fixed_in_version' ) ?>
        </td>
        <td>
        <select name="fixed_in_version">
            <?php if ($t_fixed_in_version ) {
                print_version_option_list( $t_fixed_in_version,
bug_get_field( $f_bug_id, 'project_id' ), VERSION_ALL );
        } else {
print_version_option_list(custom_version($t_project_id, VERSION_FUTURE),
    bug_get_field( $f_bug_id, 'project_id' ), VERSION_ALL );
            }?>
        </select>
        </td>
</tr>
```

经过以上的修改，在 mantis 中，提交或处理缺陷的时候，相应版本就有默认值了。

六、总结

此篇文章，仅仅是我对使用 mantis 这几个月的学习总结，我看过很多人写的 mantis 安装配置文章，里面都仅仅说如何配置 mantis 完成需要的功能。而我这个系列的文章，希望告诉大家，为什么按照别人说的配置下来，mantis 就能相应改变，大家如果能从知其然到知其所以然，也就不枉费我花费这么多的精力了。

个人的文笔有限，了解 mantis 的人可能会觉得我说的过于繁琐，但是我这个系列是给希望了解 mantis 的初学者看的，高手可以直接看代码，比我文中说的更详细和准确。

其实觉得 mantis 的结构安排不尽合理，很多的内容过于绕。但是 mantis 确实考虑到了很多的情况，看过上文就知道，很多内容不需要大改，mantis 尽可能的把函数参数接口都设计好了，可以根据需要由用户直接调整。

大概就是这些了，文章虽然检查了几遍，但是个人不敢保证里面是否还有问题。

再次感谢有耐心看完全文的的人。

浏览器兼容性测试的自动化预研分析

作者：软件测试在福建

摘要：本文是分析在不减少测试内容的情况下，是否有通过自动化手段减少测试工作量的解决方案。

关健字：兼容性测试；浏览器测试；自动化测试

一、浏览器兼容性测试现状

1、背景

浏览器是指可以显示网页服务器或者文件系统的 HTML 文件内容，并让用户与这些文件交互的一种软件。网页浏览器主要通过 HTTP 协议与网页服务器交互并获取网页，这些网页由 URL 指定，文件格式通常为 HTML。

个人电脑上常见的网页浏览器包括微软的 Internet Explorer、Mozilla 的 Firefox、Apple 的 Safari、Opera、HotBrowser、Google Chrome、GreenBrowser 浏览器、Avant 浏览器、360 安全浏览器、世界之窗、腾讯 TT、搜狗浏览器、傲游浏览器、百度浏览器等。浏览器是最经常使用到的客户端程序。

既然浏览器是用户访问 WEB 软件的主要方式，那么对于发布的 WEB 软件，就要测试用户使用各种浏览器能否正常显示

常见的浏览器兼容性问题，主要表现在如下两方面；

1）页面显示

页面显示的美观性是 Web 应用程序中重要需求，不同浏览器上呈现给用户的同一个 Web 页面可能显示的不一样。这些差异性主要表现在对于页面元素的位置、大小、外观。如果在某款浏览器上显示不美观，就会成为一个问题，需要修改。

2）功能问题

Web 软件中的功能性问题主要是不同浏览器对脚本的执行不一致，功能性问题极大的限制了用户对 Web 界面元素的使用。这类问题通常很难被发现，比如某个按钮可能显示正确但实际它是无法使用的，这个则需要用户真正的去使用它才能被发现。

2、现有解决方法

由于软件尽可能多的支持不同类型浏览器，近年来越来越多的成为一个趋势，因此测试的工作量也越来越大。而更多的方法是通过手工进行测试，如何减少工作量，当然想到的还是自动化。

自动化如何减少测试工作量，以下先通过一个实例来简要看看浏览器兼容性手工测试过程。

测试需求指标：

某款软件支持 IE6.0，IE7.0，Firfox4.3，要求测试该软件在这三款浏览器上的兼容性

手工测试过程如下：

1）手工操作：IE6.0 浏览器下，手工在界面上进行各种操作（常见的操作是点击菜单，按钮，输入框输入文字）；

2）操作结果检查：查看界面在浏览器上的显示是否美观，是否存在功能问题；

3）在 IE7.0 上重复步骤 1，2；

4）在 FireFox4.3 上重复步骤 1，2。

在业内目前通过自动化的方法减少手工测试工作量，主要的过程如下：

1）录制手工操作

将手工操作过程在一个浏览器上通过软件的方式记录下来。记录的主要是鼠标与键盘的动作，同时对每一个操作后的结果，设置一个预期结果。

2）检查测试结果

根据上个步骤录制的脚本，在其他浏览器上进行回放（自动化操作），并比对实际结果与预期结果，如果一样则测试通过，否则测试失败。

上述自动化的主要问题在于：

1）录制时需要设置预期结果，存在额外的工作量。

2）页面变化是 WEB 软件开发中经常发生的，比如变化了界面显示的文字，移动存放的位置，都可能导致录制的脚本需要进行修改，脚本的维护工作量会比较大。

3）对于操作的结果检查，由于各个浏览器对于界面的展现各不相同，（比如字体的显示各不相同），因此结果可能是多种的，就会很容易产生展现的结果也是美观的，但被误报成测试失败的现象。

分析常规的自动化过程后，我们可知对于测试结果检查，由于界面美观性，现有的自动化技术很难准确检查，因此纯自动化的方法不大可行。

而如果去掉测试结果检查的自动化，仅仅采用录制的方法减少手工测试工作量，由于只能一个个的核对操作结果，工作量及测试时间上减少的比较有限。

那是否还有什么自动化办法减少手工操作的工作量呢，本方案就是基于此来解决的。

二、解决方法

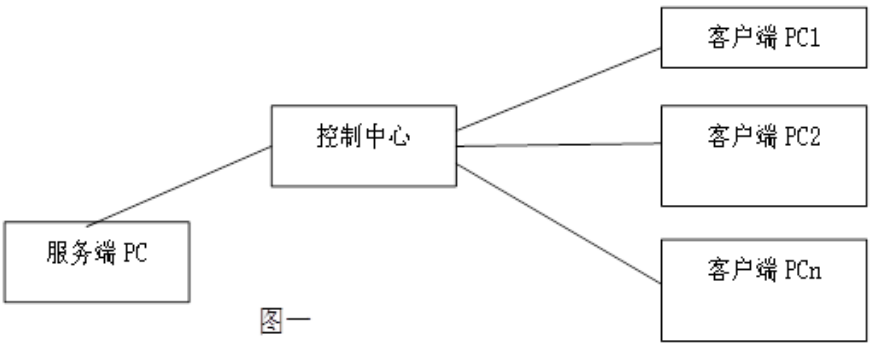
1、方案概述

通过方案中所涉及的各系统及相互协作，来达到自动化进行浏览器兼容性的测试目的。主要包含以下几种组件：

- 控制中心：负责主从间的指令与信息同步
- 服务端 PC：手工测试操作 PC；
- 客户端 PC：自动化操作 PC

每个客户端 PC，安装一款需要测试的浏览器

2、方案的网络拓扑图



图一

3、方案的工作步骤

客户端 PC 开启监听程序，实时获取控制中心下发的操作指令

服务端 PC 上进行手工测试操作：测试人员在服务端 PC 上进行手工测试

服务端 PC 记录操作指令

服务端 PC 记录手工测试操作，并形成操作指令，记录到表中。操作指令表的记录如下表：

指令序列	X 轴	Y 轴	动作	动作主体
1	423,345	21,112	Right Click	鼠标
2	61,245	56,835	Input	键盘
…	…	…	…	…
n	411,257	6,152	Left Click	鼠标

表一

其中：

“指令序列”：唯一标志一条指令

“X 轴”：操作对象所在的 X 轴值

“Y 轴”：操作对象所在的 Y 轴值

“动作主体”：发起该动作的主体，鼠标或者键盘

“动作”：动作主体的动作

1）服务端 PC 的操作指令实时上传到控制中心

每记录一条操作指令，服务端 PC 即将操作记录上传到控制中心

上传的操作指令即为表一中的一行数据

2）控制中心将操作指令下发给各个客户端 PC

控制中心本地保存一张客户端 PC 表，发下所示

客户端 PC	IP 地址
1	192.168.1.2
2	192.168.1.3
…	…
n	192.168.X.Y

表二

其中：

“客户端 PC”：客户端 PC 编号，唯一表示一台客户端 PC

“IP 地址”：该客户端 PC 的 IP 地址

根据客户端 PC 表的地址，控制中心将收到的指令下发给各个客户端 PC

3）客户端 PC 监听程序收到操作指令后，根据收到的操作指令自动化操作

如表三为收到的一条指令，那么客户端将如此操作：

a、从表中获取动作主体及动作：如本例中为鼠标的右击动作

b、从表中获取动作的位置：如本例中为 X 轴坐标 423，345，Y 轴坐标为 21，112

c、根据步骤 a 与 b，进行自动化操作：如本例中，将会在 X 轴坐标 423，345，Y 轴坐标为 21，112，进行鼠标的右击动作

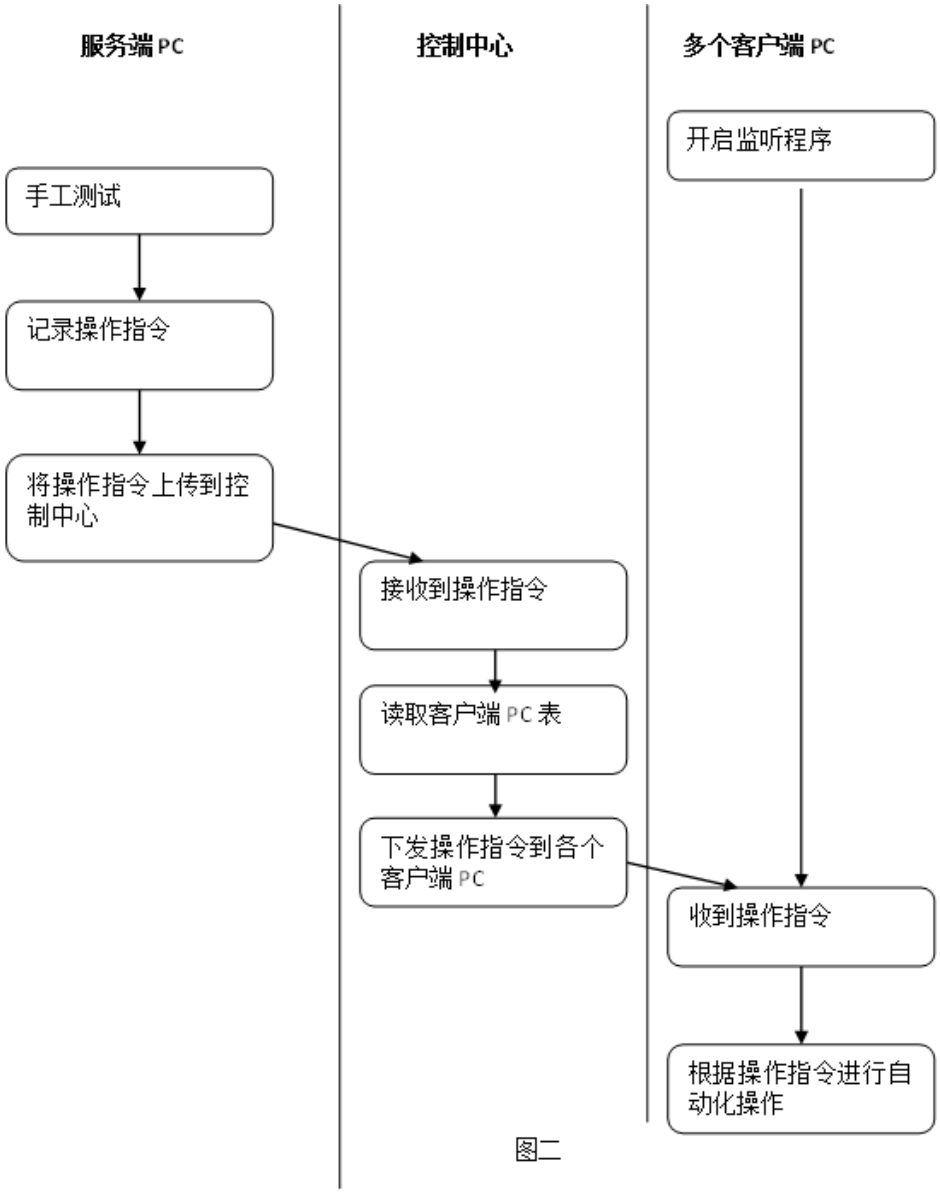
指令序列	X 轴	Y 轴	动作	动作主体
1	423,345	21,112	Right Click	鼠标

表三

4）测试人员对比自动化操作后的浏览器界面

测试人员查看多台 PC 自动化操作后的结果，判断界面美观性，是否存在问题。

5）整个控制过程图如下：



图二

三、总结

减少了手工操作的工作量；通过在一台机器上操作，并同时使多台从机上自动化的同时相同操作，减少手工操作工作量；一台 PC 的操作假设需要 T 分钟，则 N 台 PC（对应 N 个浏览器）手工操作需要耗时 T*N 分钟。通过这种方法，手工只需要操作一台 PC，则可以同一时间查看 N 台 PC 的操作结果，可以减少操作时间（N-1）*T 分钟。

可以同时查看多台测试机对于同一个操作的执行结果，提高了工作效率。而不是像传统意义上的测试，串行一个一个浏览器进行测试及测试结果核对，有效提高了工作效率，减少了测试时间。

深入分析 TestComplete 名称映射

作者：旋次

摘要：通过实例和分析对 TestComplete 名称映射的功能步步推进层层展开，将名称映射。

关键词：TestComplete；NameMapping；Templates；FullName

一、前言

TestComplete 是 AutomatedQA 公司开发的一套支持自动测试软件的工具，近年来发展异常迅速，屡获大奖。支持 VBScript、DelphiScript、JScript、C#Script 等多种脚本语言，支持传统的各类桌面应用程序、WEB 应用、单元测试和分布式测试等等领域，最新版本对 flash 的测试技术发展也非常迅猛，据说可以不用加编译参数重新编译被测程序源码。

对比同领域的其他的自动化测试工具，TestComplete 的特点是简洁而强大。在 TestComplete 的常规应用中，对象识别能力也非常出色，本文重点分析其对象识别的原理和使用方法。

二、TestComplete 名称映射分析

1、FullName 和 MappedName

Windows 系统中应用程序、从属于应用程序的窗体、窗体中的控件都是对象，每个对象都有唯一的标识（通常是十六进制的一个值），这些共同构建出一个庞大的对象目录树，根节点就是 Sys，可以想象一棵挂满了十六进制数树叶的树，每片树叶还有属性方法事件装饰它，如果录制的代码全是十六进制数做对象名称，所有人都会崩溃的，所以首先要对这些对象起个名字，最简单的法子是使用这个对象的类似 caption，index 和 classname 一类的属性来起一个名字，于是 TestComplete 给每个在内存中的对象起了名字，为了清晰表明各个对象的层次关系，使用了 fullname 来标识。Fullname 看起来大概象这个样子：

Sys.Process("IEXPLORE").Page("http://xxx:nn/xxx.html").Panel("ext_comp_1017").Panel("ext_gen15").Panel("ext_gen16").Panel("ext_comp_1009").Panel("ext_gen26").Panel("ext_gen28").Panel("ext_comp_1008").Panel("ext_gen75").Panel("ext_gen76").Panel("ext_comp_1073").Panel("ext_gen161").Panel(0).Panel(0).Panel(0).Panel("ext_gen162").Panel("ext_comp_1075").Table(0).Cell(0,7).Table("ext_comp_1048").Cell(0,1).Button("ext_gen195")

呃…介个…是有点头晕，实在是太长了，不便于书写和阅读，是需要搞个短点的名字才行，于是 TestComplete 搞了个映射名称 MappedName：

NameMapping.Sys.IEXPLORE.pageHttp1213289411581Citymanager.panelExtComp1017.panelExtGen15.panelExtGen16.panelExtComp1009.panelExtGen26.panelExtGen28.panelExtComp1008.panelExtGen75.panelExtGen76.panelExtComp1073.panelExtGen161.panelXPanelMl.panelXPanelMr.panelXPanelMc.panelExtGen162.panelExtComp1075.table.cellExtGen193.tableExtComp1048.cellXBtnCenter.buttonExtGen195

如果看起来还是有些长，可以使用对象引用的法子来截短一些：

Aliases 是 NameMapping.Sys

Set iexplore=Aliases.IEXPLORE

Set page=pageHttp1213289411581Citymanager

Set p1=panelExtComp1017.panelExtGen15.panelExtGen16.panelExtComp1009.panelExtGen26

Set p2=p1. panelExtGen28.panelExtComp1008.panelExtGen75.panelExtGen76

Set p3=p2.panelExtComp1073.panelExtGen161.panelXPanelMl.panelXPanelMr.panelXPanelMc

Set p4=p3.panelExtGen162.panelExtComp1075

Set p5=p4.table.cellExtGen193.tableExtComp1048.cellXBtnCenter.buttonExtGen195

Call p5.click

这下看起来好点了。

2、MappedName 有啥用处

映射名称仅仅是为了书写阅读的方便吗？当然不是！那是为了什么？

试想你录制了一段脚本，录制的时候对象 obj，显示的内容是"aaa07201701"，回放的时候这个显示的内容发生了变化，成了"aaa07201721"，内容 aaa 的后面几位是时间，每次运行的时候这个时间都会相应变化，但是你希望能得到该对象的

引用，加以操作，如果是使用 fullname，名称可能就从 obj("aaa07201701")变成了 obj(" aaa07201721")

问：怎么办？

答：可以使用类似正则匹配的方式来寻找对象，修改对象引用为 obj("aaa*")，这下总行了吧。

问：不见得哟，假设有很多个对象都一样的显示，obj("aaa*")匹配上的就会是很多个名称 aaa 开头的对象。

答：那就使用其他的属性来共同限定，比如这个对象类型是 button，其他显示为 aaa 开头的对象都不是 button，那就可以 obj("aaa*","classname=Button")，要是有多个 button 都是一样显示，那就加上实例索引，第几个 button，例如：obj("aaa*","classname=Button",2) ，这次没问题了吧。

是的，这样的确就已经可以了，不过这个方法有个缺点，对象的引用修改全部是在源代码中，如果代码量很大，非常容易修改错误，有什么好点的法子吗？

答：当然，可以使用 NameMapping。双击开 project 下的 NameMapping（没有？自己创建一个，或者随便录制一段脚本，结束录制后会自动保存一个的），双击目录树上你的对象 obj aaa07201721 进入一个编辑界面，把左面的 text 属性修改为 aaa*，然后添加上右面的 classname=Button，其实和修改 fullname 是一样的效果，但是这种法子有个好处，更直观，并且修改是在 NameMapping 中统一修改的，便于管理，不用跑源码里面去找得那么痛苦，当然现在 TestComplete 高版本的也不会再使用 fullname 来录制脚本了，都是映射名称了。

想起前面的那个问题，现在来回答一下，映射名称还可以使得修改属性更方便管理，就这样啦？还有没有其他的目的呢？

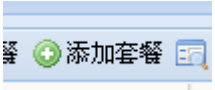
3、MappedName 的本质

问：如果整个项目中这种需要修改属性的情况很多，一个个这么修改不是很崩溃？

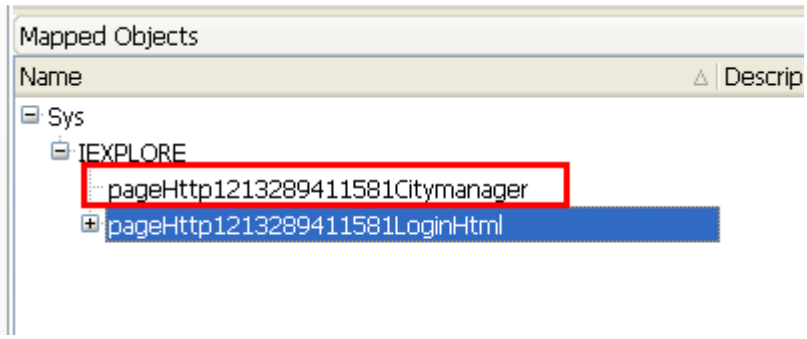
答：不可能吧，哪有那么多意外情况。

那就给你看个例子，这是录制的一个 web 应用：

录制的内容是点击添加套餐：



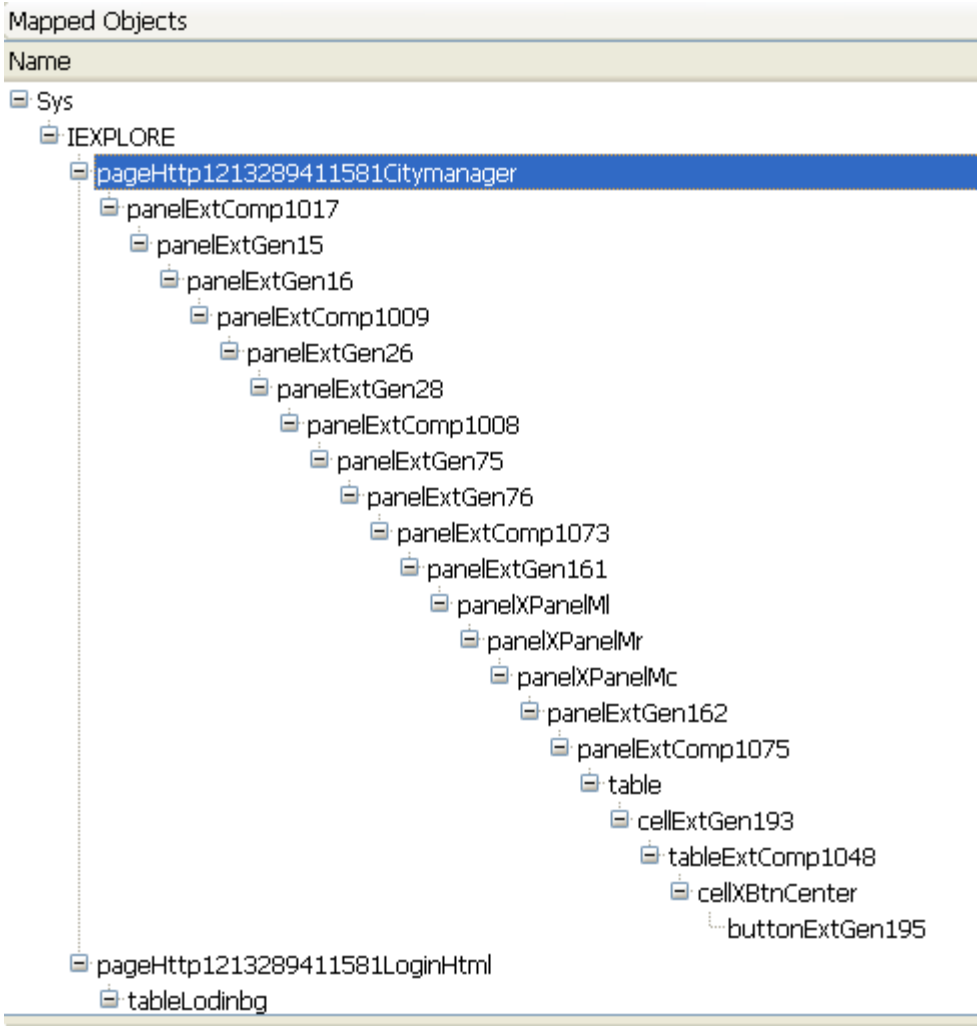
录制前可以看到该映射名称下没有子节点：



录制完成后：

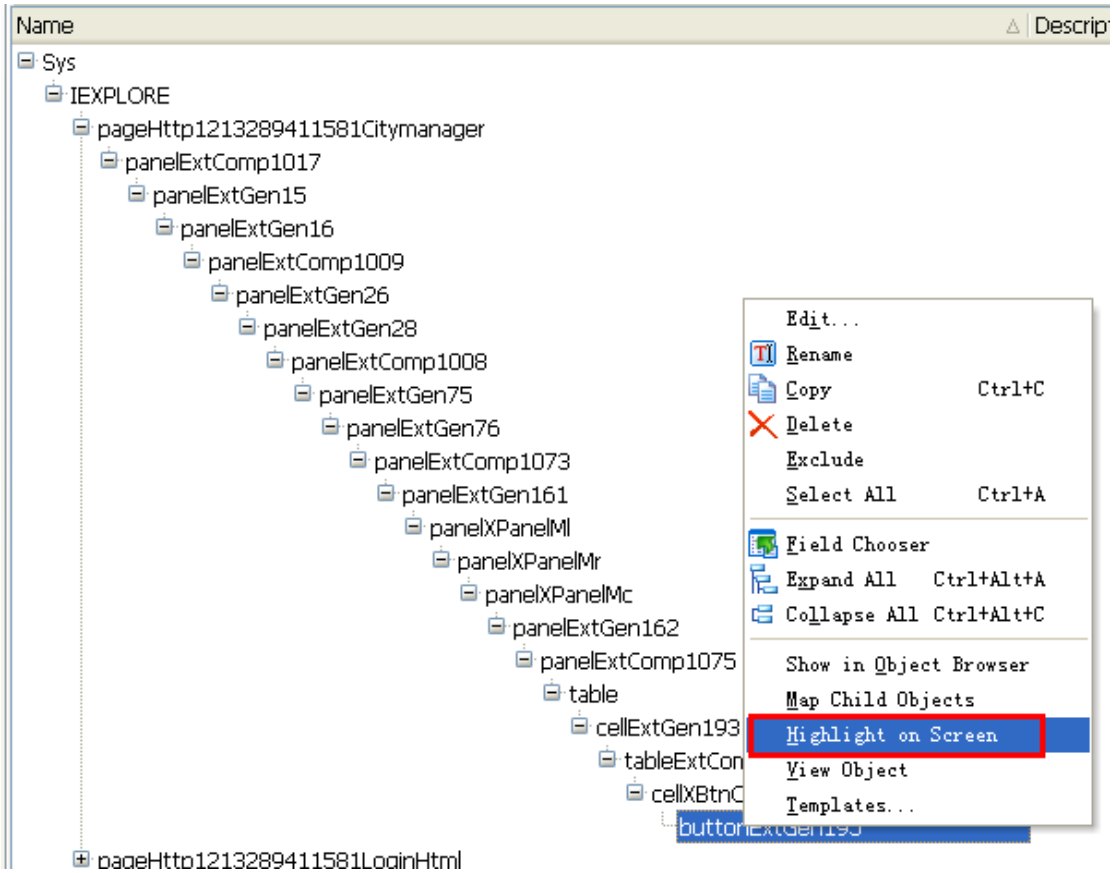
```
Sub Test1
    Dim iexplore
    Set iexplore = Aliases.IEXPLORE
    Call iexplore.ToURL("http://xxx/xxx/cityMainFrame.html")

iexplore.pageHttp1213289411581Citymanager.panelExtComp1017.panelExtGen15.panelExtGen16.panelExtComp1009.panelExtGen26.panelExtGen28.panelExtComp1008.panelExtGen75.panelExtGen76.panelExtComp1073.panelExtGen161.panelXPanelMl.panelXPanelMr.panelXPanelMc.panelExtGen162.panelExtComp1075.table.cellExtGen193.tableExtComp1048.cellXBtnCenter.buttonExtGen195.Click
End Sub
```



关闭 IE，重新登录后，回放脚本，提示找不到对象了。

找到之前录制脚本中的映射名称：

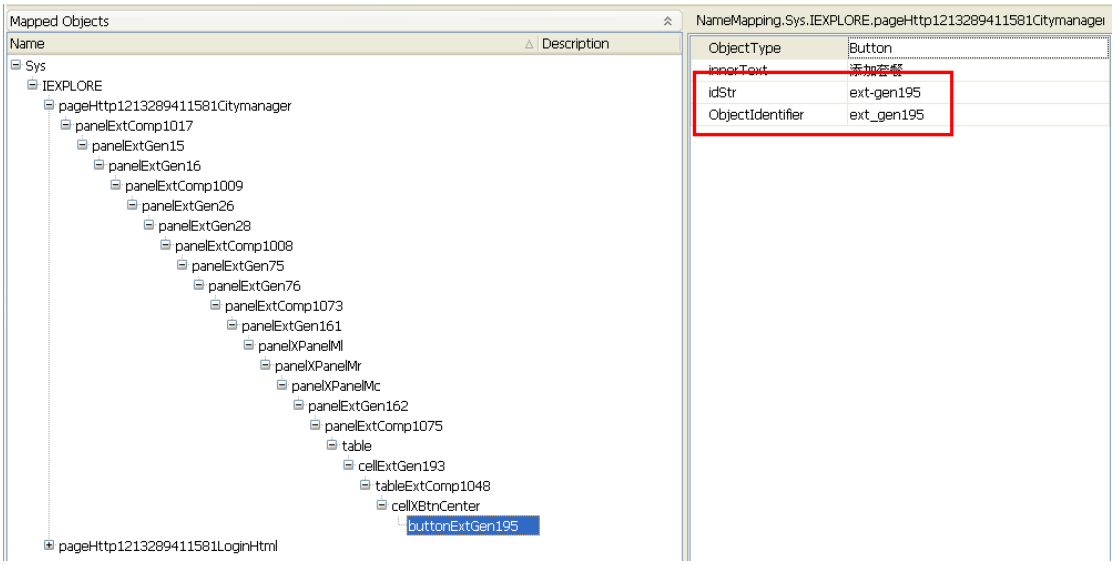


说明之前的映射名称根据预先设置的属性和属性值，查找不到现在的实例了。

让我们来猜猜 TestComplete 的映射名称是怎么根据自己的预设属性和属性值来找到对应的对象实例的呢，我猜大概应该是这样的。

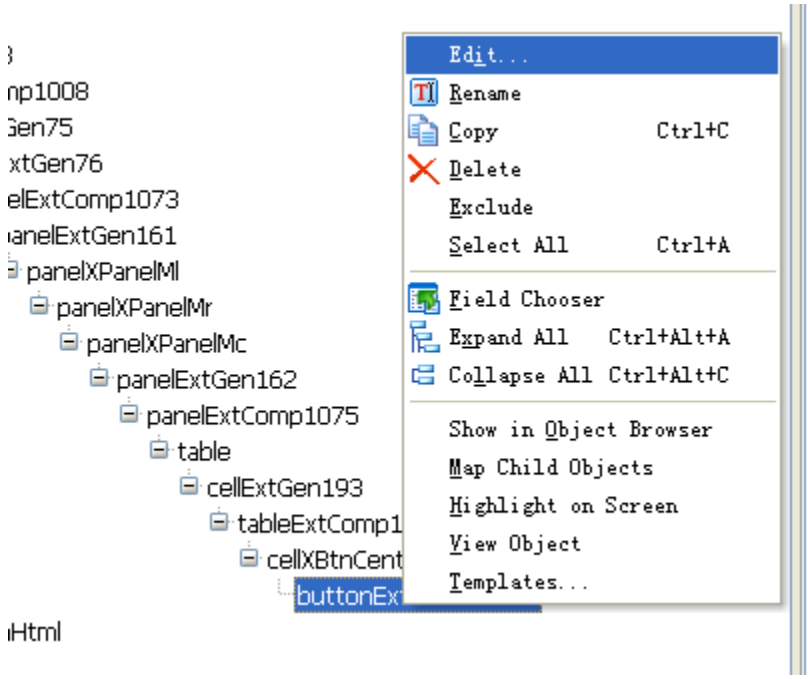
- 首先是 NameMapping 映射上 Sys
- 根据进程名字和实例顺序找到进程 IEXPLORE
- 在 IEXPLORE 的儿子辈对象中，查找出 ObjectType=Page 并且 URL=xxx 的一个对象，这样 iexplore.pageHttp1213289411581Citymanager 就和 Sys.Process("IEXPLORE").Page("http://xxx:nn/xxx.html")映射上了

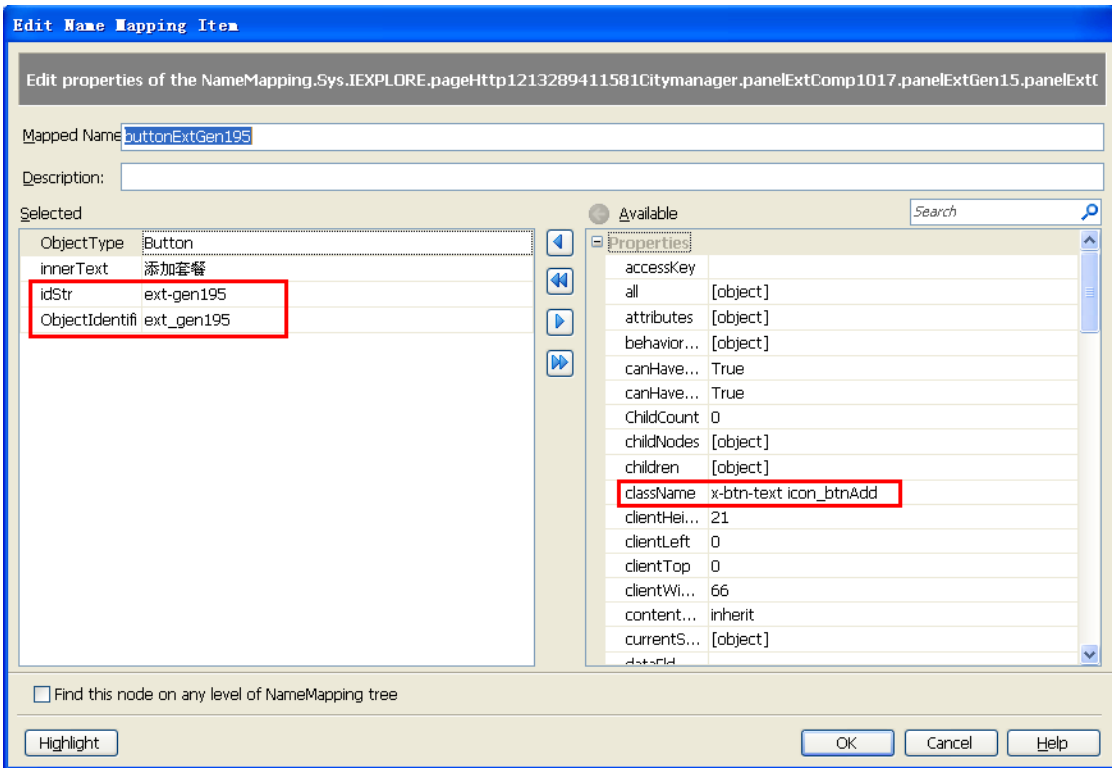
- 后面同一个逻辑，匹配上一个对象实例后，根据映射名称设定的属性和属性值，在该匹配上的对象实例的儿子辈对象中查找相应对象，一直到结束



其实只要看看过滤的属性就知道了，当页面中的<添加套餐>出现的时候，在其已经匹配上的儿子辈对象实例中，用四个属性一个个的去比对，objecttype=button 的对象，可能有好多个，再在这几个对象中继续搜索，innerText=添加套餐的，刚好就只有一个！这不就成功了，不好意思，还有两个属性呢，在找到的对象中再找找[idStr=ext-gen195]和[OjbectIdentifier=ext_gen195]的对象，这下糗了，因为这两个属性的值发生了变化，不再是 ext-gen195，而是 ext-genXXX 了，具体是多少没有规律，每次运行都不同。

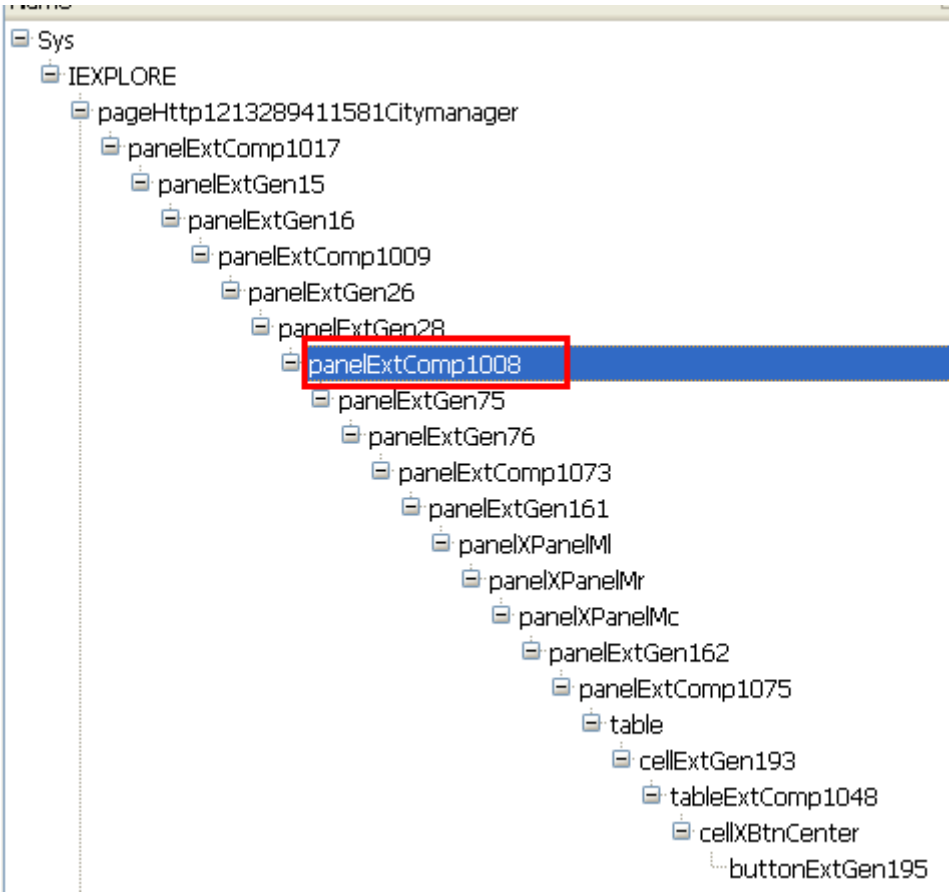
前文也描述过类似情况，那我们去掉那两个属性，留下可以匹配为唯一的属性不就行了，聪明！开工。


np1008
Gen75
xtGen76
elExtComp1073
anelExtGen161
panelXPanelMl
panelXPanelMr
panelXPanelMc
panelExtGen162
panelExtComp1075
table
cellExtGen193
tableExtComp1
cellXBtnCent
buttonExt
iHtml
Edit...
Rename
Copy Ctrl+C
Delete
Exclude
Select All Ctrl+A
Field Chooser
Expand All Ctrl+Alt+A
Collapse All Ctrl+Alt+C
Show in Object Browser
Map Child Objects
Highlight on Screen
View Object
Templates...


Edit Name Mapping Item
Edit properties of the NameMapping.Sys.IEXPLORE.pageHttp1213289411581Citymanager.panelExtComp1017.panelExtGen15.panelExt(
Mapped Name buttonExtGen195
Description:
Selected
Available
Search
ObjectType Button
innerText 添加套餐
idStr ext-gen195
ObjectIdentifi ext_gen195
Properties
accessKey
all [object]
attributes [object]
behavior... [object]
canHave... True
canHave... True
ChildCount 0
childNodes [object]
children [object]
className x-btn-text icon_btnAdd
clientHei... 21
clientLeft 0
clientTop 0
clientWi... 66
content... inherit
currentS... [object]
Find this node on any level of NameMapping tree
Highlight
OK
Cancel
Help


Selected
ObjectType Button
innerText 添加套餐
className x-btn-text icon_btnAdd

再次回放，发现还是不行，郁闷，再仔细一看，明白了



这个目录上面随便一个节点，都是 idStr 和 ObjectIdentifier 属性约束的，下次运行的时候就发生了变化。Oh，卖锅的！一个个 panel 改下去不得累死！这还仅仅是一个点击动作，要整个项目做下来，那比给周扒皮当长工还惨。不管怎样，有个法子总比没有法子好。痛定思痛，发现更痛。因为发现这个法子其实根本就不可行，原因如下：

选择 className 其实还是有风险的，仔细观察会发现 web 中的对象类名在不同的状态下有可能不同。

比如：

className=x-panel-header x-unselectable x-accordion-hd，这次是 unselectable，下次可能就是 selectedtable。

也就是说，除非你修改属性的同时，保证界面在相应的操作状态上才行。

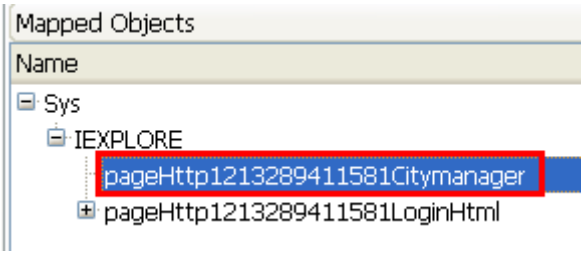
彻底崩溃了，除非有个方法，让 TestComplete 在录制脚本的时候，映射 panel 对象就使用 className，objectType，innerText 三个属性，这样才能记录到运行

当时的值，然后再修改 innerText，去掉一些会变化的内容，替换成星号就可以了，至于怎么修改，你可以在 NameMapping 的目录树上修改，也可以跑到 NameMapping 目录下修改文件 NameMapping.tcNM，不过不建议这种法子，怕改错了。现在问题的关键是怎么提前预设需要的属性和属性值。

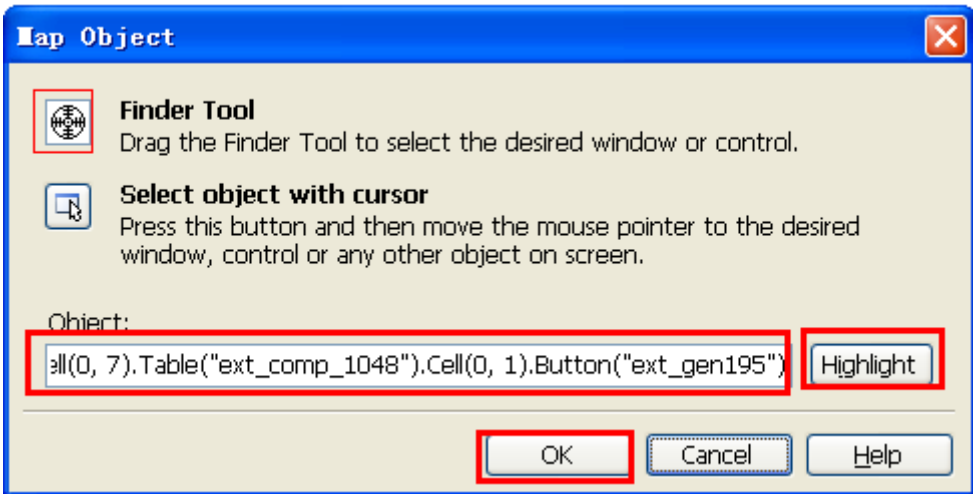
经过摸索，找到了一个方法：

以 panel 为例，需要去掉[idStr 和 ObjectIdentifier]，使用[className，objectType，innerText 三个属性]，在录制的时候就用这三个属性进行 panel 的名称映射。

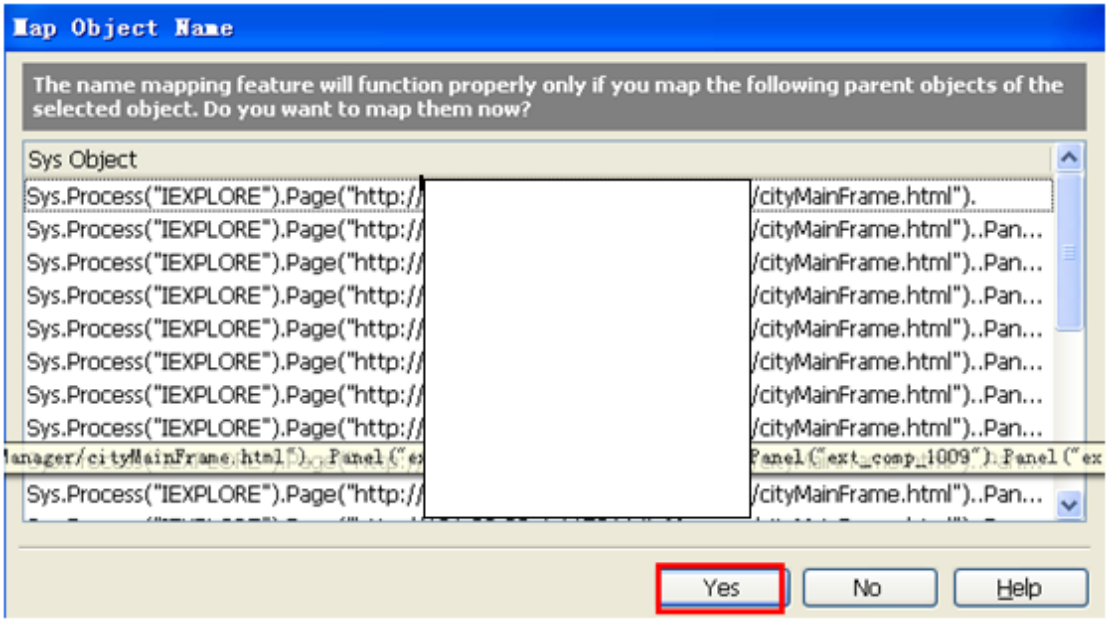
- 在录制脚本之前删除掉不用的 namemapping



- 点击[map object from screen]



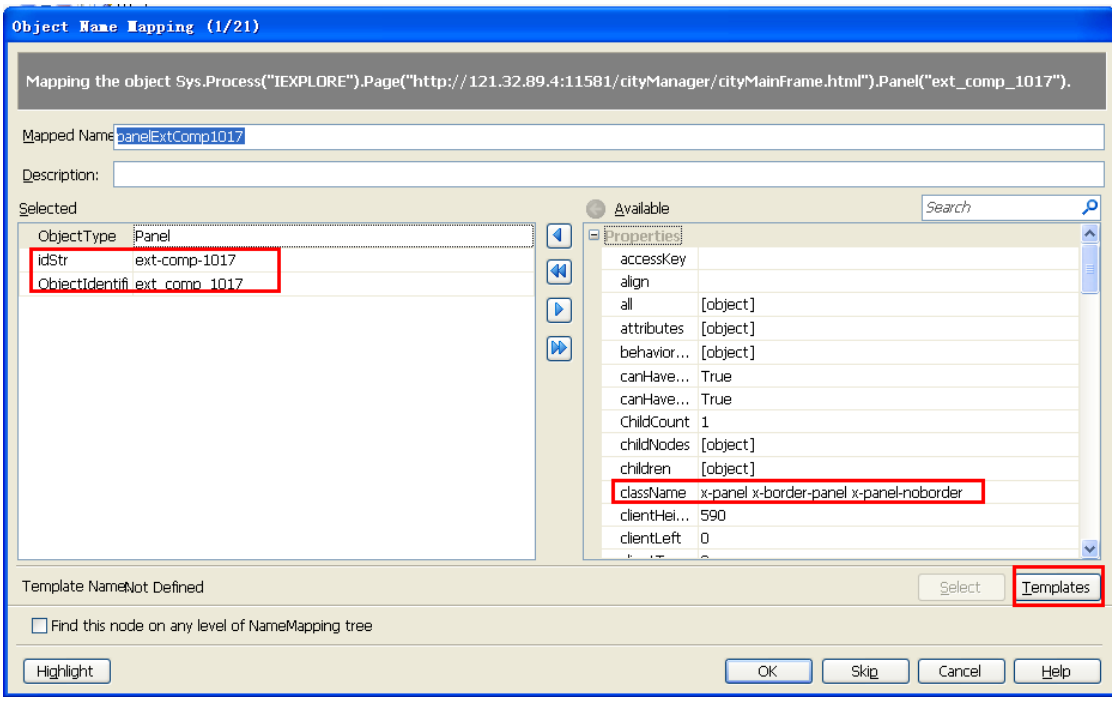
要是怕选择的不对，可以用 highlight 看看有没有选对对象。

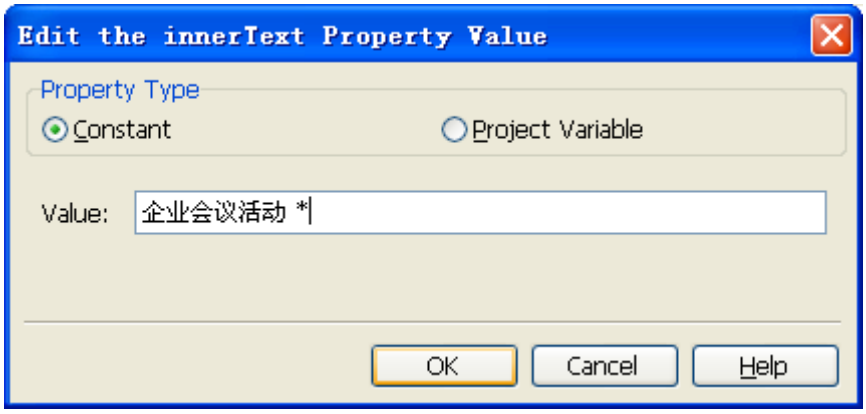


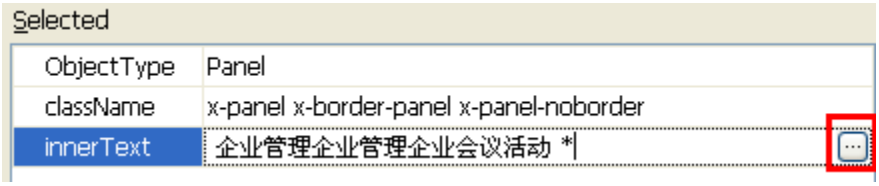
这里是要映射对象了，点击 Yes

如果提示你要映射的对象找不到，刷新一下 ObjectBrowser 就好了

● 去掉[idStr,ObjectIdentifier]，加上[innerText,className]，innerText 的值变化的部分需要修改为星号

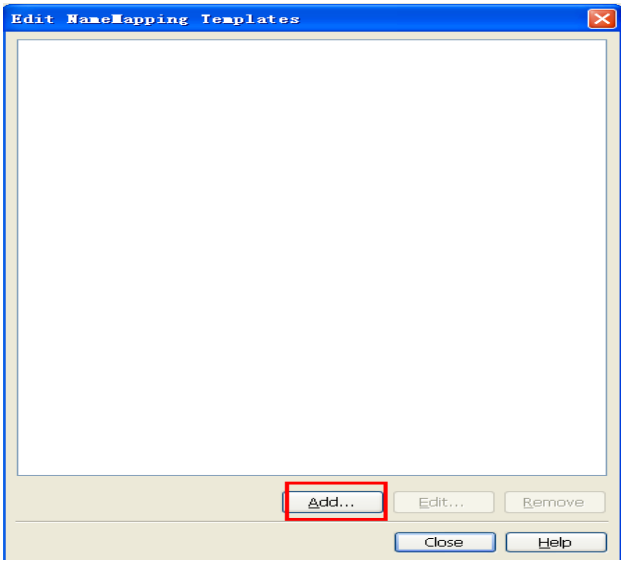


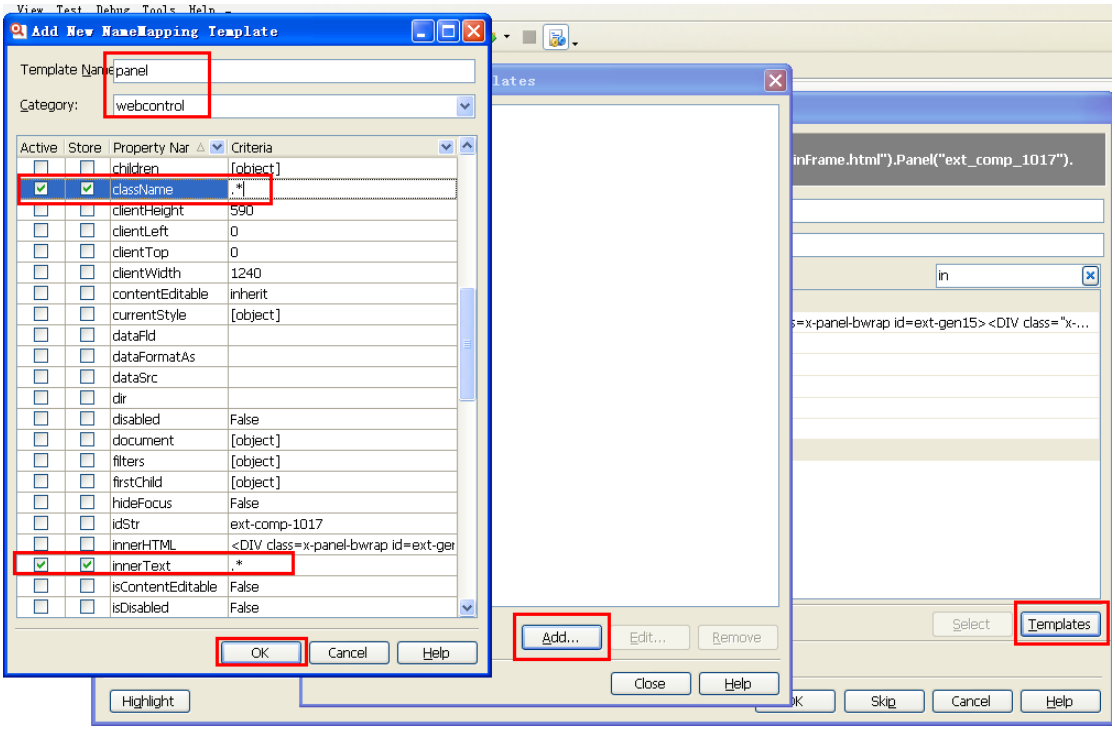




● 这里是关键了，要建立模板，当出现 panel 对象的时候：

只有在这种情况下，点击了 templates 才会出现 add 按钮，非常难找





参看帮助：Edit NameMapping Template Dialog

这里是解释 active 和 store 的：

The properties currently used to match the mapped object with the template have the Active option checked. To modify this set of properties, simply check or uncheck their Active checkbox.

Note: When name mapping, TestComplete automatically chooses one or several templates that match the mapped object. If that object does not contain a property specified by the template or the property's value does not match the value defined by the template, the template is not chosen for the object.

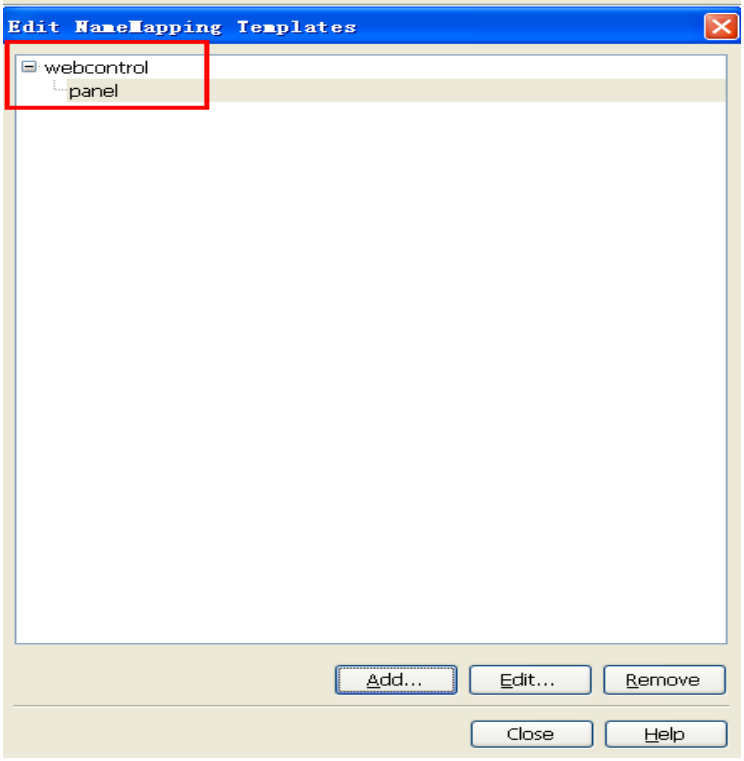
The properties that are added to the list of properties used for object identification, have the Store option checked. To change this property set, check or uncheck the Store checkbox for the desired properties.

Note: Do not use the Name and FullName properties to identify objects. Objects mapped using these properties may be identified incorrectly.

简单的说，就是如果钩选了 active，等你点击确定退出模板窗口，你会发现当前窗口的属性变成你的最新设定了；

只要钩选了 store 识别 panel 的时候，就会启用 className=.*、innerText=.* 和 ObjectType=panel

说了一大堆，其实就这点管用。



模板都建立了，大功告成了，你可以试试再录制脚本，凡是 panel 对象就会启动新设置的三个属性进行映射了。

做到这里，再回头看看之前的那个问题，NameMapping 除了方便书写阅读，还可以使得修改属性更方便管理，其实 NameMapping 最重要的能力是根据设定条件进行名称的映射，对象的高效识别。

TestComplete 看界面很简单，其实内在丰富呀。不单对象展现直观，更重要的是对象的识别简单实用。些许收获，不敢专享，有错漏之处，望不吝赐教。