

# 自动化测试实例报告

(制度报表生成子系统)

——测试部 王攀攀

## 一、概述

### 1.1 测试目的

本次测试通过三个简单实例来描述 TestComplete 完成制度报表生成子系统自动化测试过程。

### 1.2 测试方法

- 测试工具及脚本

应用 TestComplete 工具作为测试工具，对应用系统进行功能测试，并通过脚本进行管理等等。

- 测试方法

通过工具 TestComplete 录制并回放，增加脚本编写，使其回放值与期望值一致。

### 1.3 测试工具介绍

TestComplete——自动测试管理工具，全面支持工程层面上的测试，包括个体单元、性能测试、功能测试、回归测试、分布式测试以及 HTTP 性能测试等。

作为 Aqtest 的后继产品，TestComplete 提供系统化、自动化和结构化的测试功能，支持 Visual Studio .NET, Java, Visual Basic, C++ (Visual C++ and C++Builder), Delphi 和 Web 程序。

## 二、自动化测试实例一

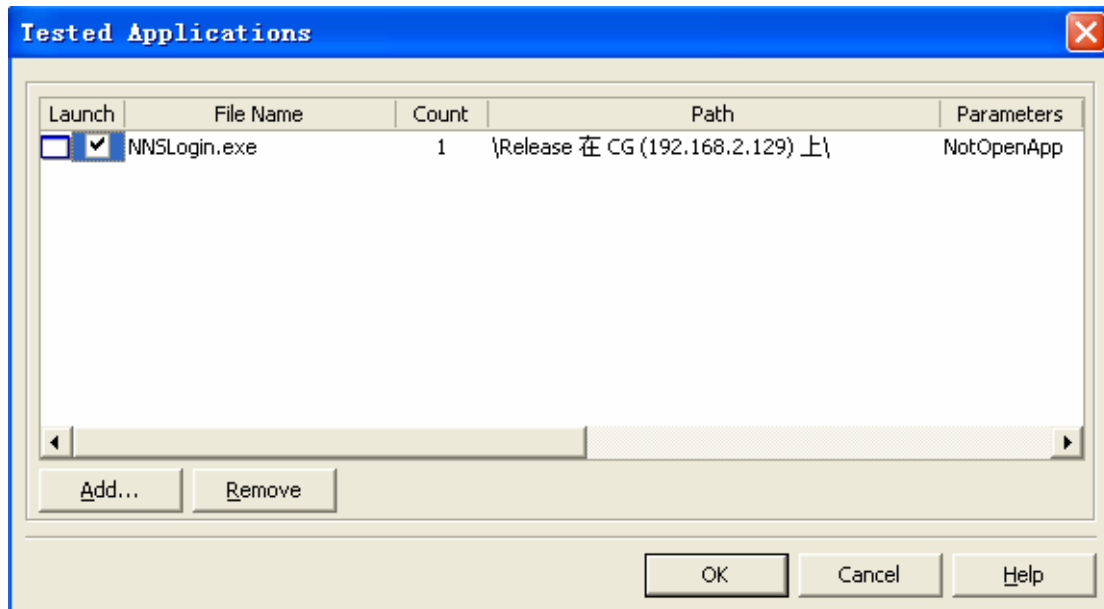
### 2.1 测试实例特征

本测试实例包括，用户登录，通过对指标集的维护进行自动化测试，包括指标的录入，删除，修改及保存功能。并对成功和异常操作分别记录操作日志和异常日志。

## 2.2 测试过程

### 1 装载应用

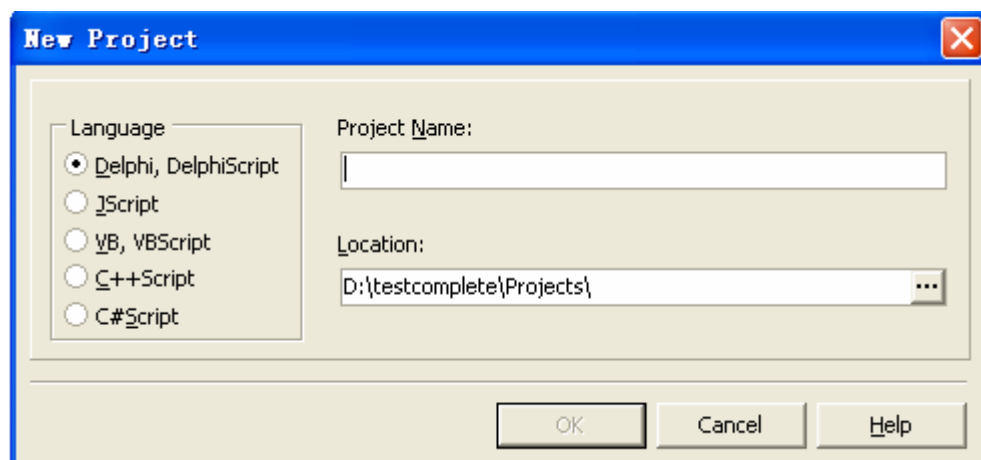
点选菜单 File—Launch Applications,打开如下界面:



点击 Add 按钮加载应用,可加载多个,并可设置参数。本实例只加载单个应用,无参数(默认为 NotOpenApp)。

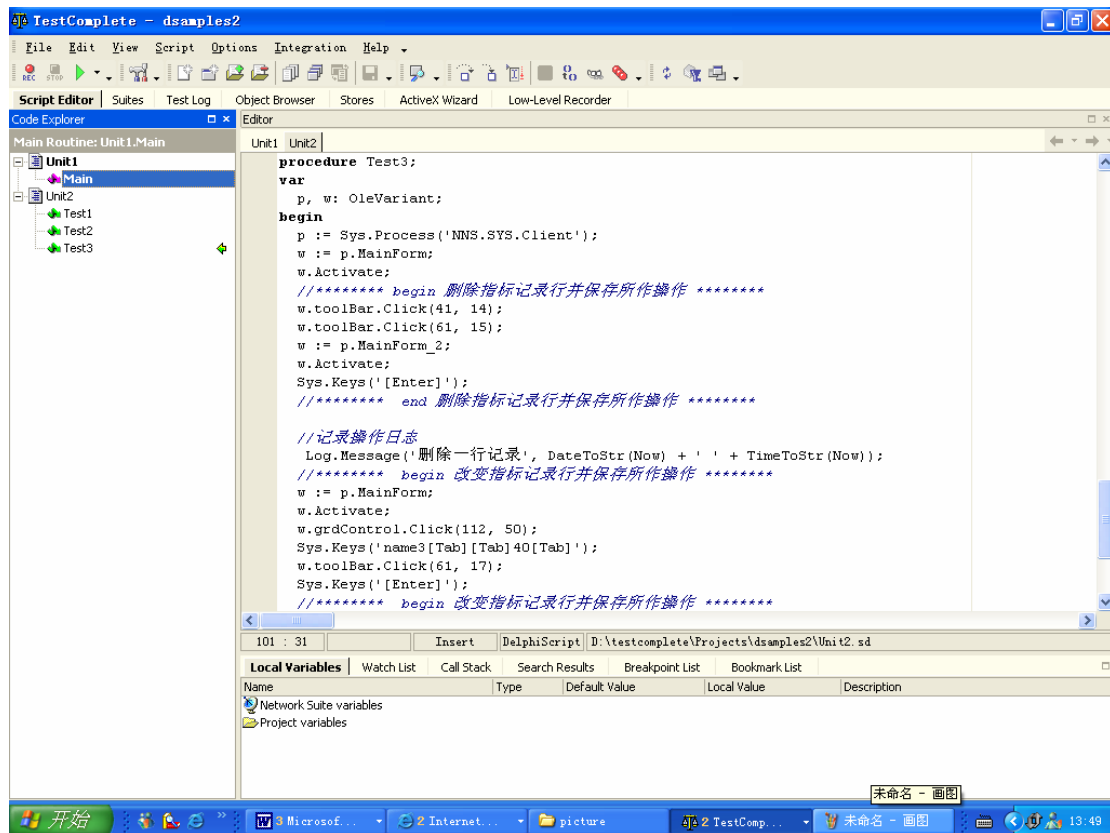
### 2. 创建工程

点击 File—New-Project,出现图示界面:



选取脚本编码语言。

本工程主要包括两个 unit.unit1 中有 main,主要是应用启动脚本编写.unit2 中有三个 test,分别是各功能维护操作录制。各功能罗列如下图所示：



### 3. 启动应用脚本编写

```

uses unit2;

var
  p, app : OleVariant;
procedure Main;
begin
  //***** begin 启动应用 *****
  app := TestedApps.Items[0];
  TestedApps.Items[0].Parameters := 'NotOpenApp';
  //***** end 启动应用 *****

  p := app.Run;

  //***** 执行测试 *****

try
  Test1;
  test2;
  test3;
except
  
```

```

//***** 获取异常并记录日志*****
Log.Error('异常错误', '注意', pmNormal, fmBold + fmItalic, clRed);

end;
app.Close;
//***** begin 延时 1 秒钟至应用关闭 *****
Sys.Delay(1000);
//***** end 延时 1 秒钟至应用关闭 *****
if p.Exists then
    //***** 如果应用仍然启用，强行终止 *****
    app.Terminate;
end;

```

●本应用启动编码较为简单，加载应用后判断是否成功并做不同处理，详见脚本注释。这里涉及两个基本的方法处理：log.error()和 Delay。

- ▲ 通过 Error 事件记录系统异常操作，便于测试脚本的维护
- ▲ 如果测试的应用要进行一些费时的操作，一般用 Delay 事件一来延迟施行，例如：装载数据库或连接到系统计算机

## 4. 录制功能

本实例功能分三个 TEST 录制。

TEST1 录制用户登录及指标集创建部分；

TEST2 录制指标集子项的录入及保存；

TEST3 录制指标项的删除，修改功能维护操作，并对所作操作保存。

### I. TEST1 脚本编写

```

procedure Test1;
var
    p, w, u: OleVariant;
begin
    //***** begin 启动登录进程，打开用户登录窗口 frmLogin *****
    p := Sys.Process('NNSLogin');
    w := p.frmLogin;
    w.Activate;
    Sys.Keys(' admin[Tab]123456'); //键入用户及密码
    w.btnLogin.Click(36, 3);
    //***** end 启动登录进程，打开用户登录窗口 frmLogin *****

    //*****begin 重启进程，打开南宁市政府决策支持与分析系统窗口
    MainLogin*****

```

```


p := Sys.Process('NNSLogin'); ①  

//延时 1 秒钟等待打开窗口  

Sys.Delay(1000);  

w := p.MainLogin('WindowsForms10.Window.8.app3','*');②  

//***** end 重启进程，打开南宁市政府决策支持与分析系统窗口  

MainLogin*****  
  

w.Activate;  

Sys.Keys(' [Enter]');  

//***** begin 启动客户端进程，打开主窗口 MainForm *****  

p := Sys.Process('NNS.SYS.Client');  

w := p.MainForm;  

w.Activate;  

//***** end 启动客户端进程，打开主窗口 MainForm *****  
  

//***** begin 切换 TAB 页至指标页并打开新增指标集弹出窗口 *****  

w.tabStrip1.Click(54, 14);  

w.trvGuidLine.HScroll.Pos := 0;  

w.trvGuidLine.Click(' |指标集');  

w.trvGuidLine.ClickR(' |指标集');  

w.PopupMenu.Click(' 新增指标集');  

//***** end 切换 TAB 页至指标页并打开新增指标集弹出窗口 *****  
  

//***** begin 创建指标集 *****  

p := Sys.Process('NNS.SYS.Client'); //切换 WINDOW 时，重启进程，否则窗口找不到  

到  

w := p.frm_NewItem_2;  

w.Activate;  

w.Object_2.Click(27, 13);  

Sys.Keys(' wp001[Tab]namewpp001');  

w.btnOK.Click(19, 17);  

//***** end 创建指标集 *****  
  

//***** 记录操作日志 *****  

log.message(' 创建指标集成功', DateToStr(Now) + ' ' + TimeToStr(Now));③  

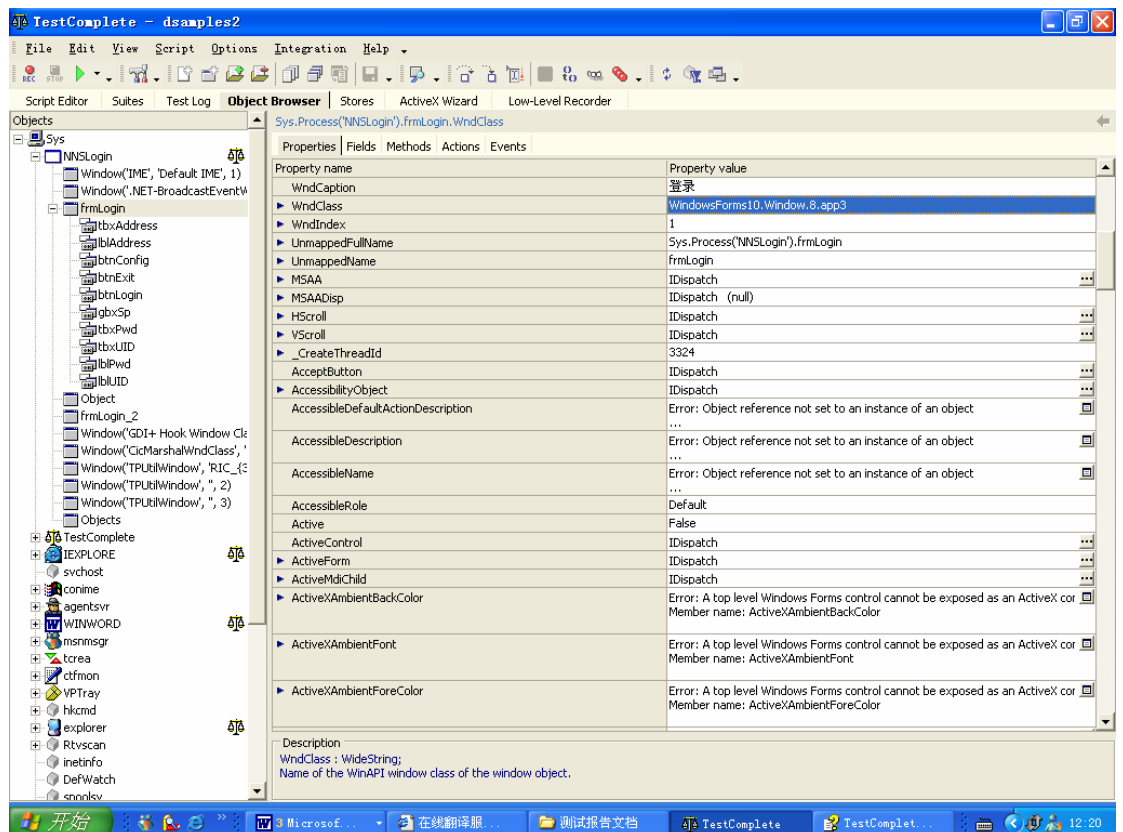
end;


```

●本 TEST 主要方法应用:

- ▲ 进程的重启（见上红色标注①）。启动一个进程，如果打开多个窗口，回放时因为窗口都开启，容易找不到窗口。此时可以重启一下窗口所在的进程，则可以很快的找到窗口。
- ▲ 打开窗口的参数设置（见上红色标注②）。加入窗口参数设置，process.window()有三个参数（WndClass, WndCaption, WndIndex）。参数获取方法：找开设置窗口，切换 TAB 页至 ObjectBrowser 页面，Sys 下第一

个即为应用，展开相应的应用，找到设置窗口，则其属性右视窗相应显示，上三个参数可以找到，如下图。



▲ log.message()的记载（见上红色标注③）。记录相应的操作日志，便于以后的脚本维护。

## II. TEST2 脚本编写

```
procedure Test2;
```

```
var
```

```
    p, w: OleVariant;
```

```
begin
```

```
    p := Sys.Process('NNS.SYS.Client');
```

```
    w := p.MainForm;
```

```
    w.Activate;
```

```
    ***** begin 切换 TAB 页面至指标 TAB 页，并选中指标集打开新增指标集弹出窗口
```

```
    *****
```

```
    w.trvGuidLine.HScroll.Pos := 0;
```

```
    w.trvGuidLine.Click('指标集|namewpp001');
```

```
    w.trvGuidLine.HScroll.Pos := 0;
```

```
    w.trvGuidLine.DbClick('指标集|namewpp001');
```

```
    ***** end 切换 TAB 页面至指标 TAB 页，并选中指标集打开新增指标集弹出窗口
```

```
    *****
```

```
    w.toolBar.Click(15, 15); //点击 toolBar 新增按钮
```

```

//***** begin 新增指标集子项，新增三条记录 *****
w.grdControl.Click(27, 30);

Sys.Keys('wp01[Tab]name1[Tab][Tab]20[Tab][Tab]unit[Tab]comment[Tab][Enter]wp02[Tab]name2[Tab][Tab]20[Tab][Tab]unit[Tab]comment[Enter]wp03[Tab]name3[Tab][Tab]20[Tab][Tab]unit[Tab]comment');
//***** end 新增指标集子项，新增三条记录 *****

w.Object_9.Click(1, 5);
w.toolBar.Click(59, 14); //点击 toolBar 保存按钮
//以下是对弹出窗口保存对话框的特殊处理，将鼠标点击确定按钮转换为 ENTER 键
w := p.MainForm_2;
w.Activate;
//w.MainForm.Click; //录制时记录鼠标轨迹
Sys.Keys('[Enter]'); //转换为 ENTER 键
//记录操作日志
Log.Message('新增三条指标项记录', DateToStr(Now) + ' ' + TimeToStr(Now));
end;

```

● 本实例主要方法应用：

- ▲ 对弹出窗口的特殊处理，当打开多个窗口，如主窗口上弹出一个窗口，点击弹出窗口的确定按钮，对其鼠标轨迹 TestComplete 只能录制到主窗口，此时可用 Enter 键等同于“确定”按钮。
- ▲ 在功能确定的情况下，录制时鼠标只作必要的点击，关闭无关的页面，按规范操作，否则会录到赘余代码。
- ▲ 使用 sys.keys()事件做键盘输入时，输入错误重新错误后，编码中要过滤错误输入。

### III. TEST3 脚本编写

```

procedure Test3;
var
  p, w: OleVariant;
begin
  p := Sys.Process('NNS.SYS.Client');
  w := p.MainForm;
  w.Activate;
//***** begin 删除指标记录行并保存所作操作 *****
w.toolBar.Click(41, 14);
w.toolBar.Click(61, 15);
w := p.MainForm_2;
w.Activate;
Sys.Keys('[Enter]');
//***** end 删除指标记录行并保存所作操作 *****

```

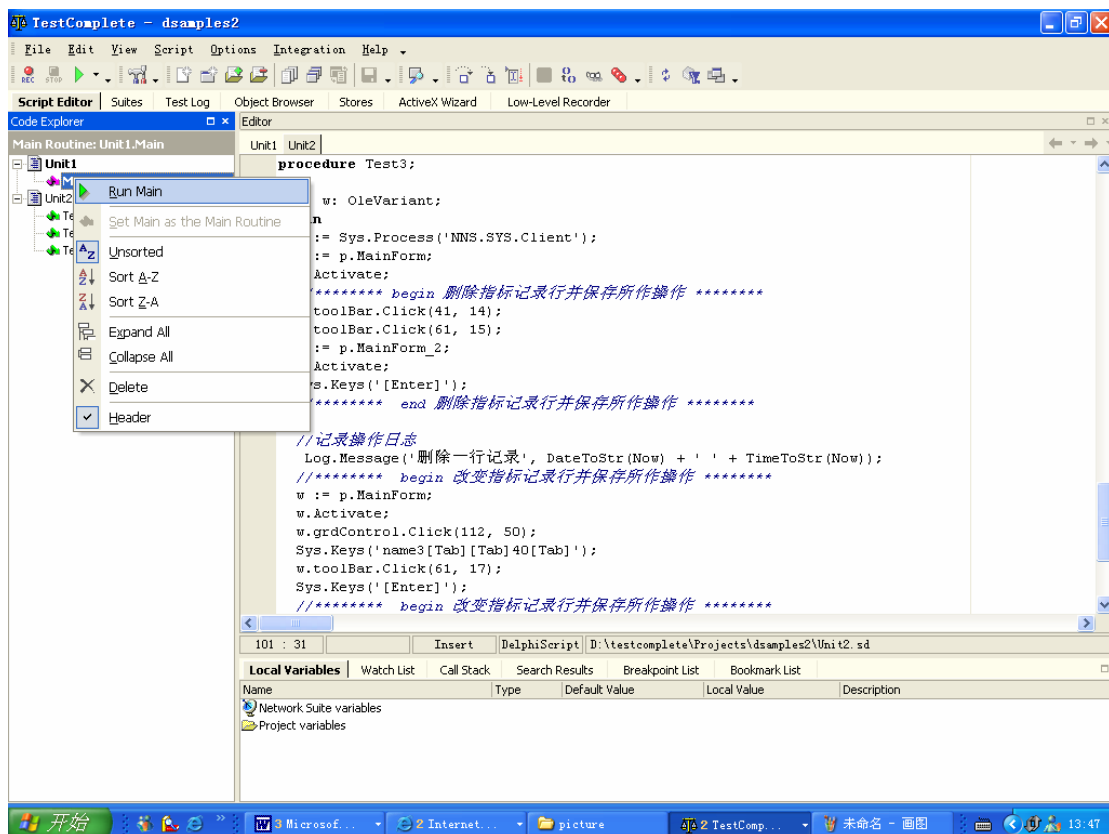
```
//记录操作日志
Log.Message('删除一行记录', DateToStr(Now) + ' ' + TimeToStr(Now));
//***** begin 改变指标记录行并保存所作操作 *****
w := p.MainForm;
w.Activate;
w.grdControl.Click(112, 50);
Sys.Keys('name3[Tab][Tab]40[Tab]');
w.toolBar.Click(61, 17);
Sys.Keys('[Enter]');
//***** begin 改变指标记录行并保存所作操作 *****

//记录操作日志
Log.Message('修改一条指标记录', DateToStr(Now) + ' ' + TimeToStr(Now));
end;
```

● 本实例中主要方法应用

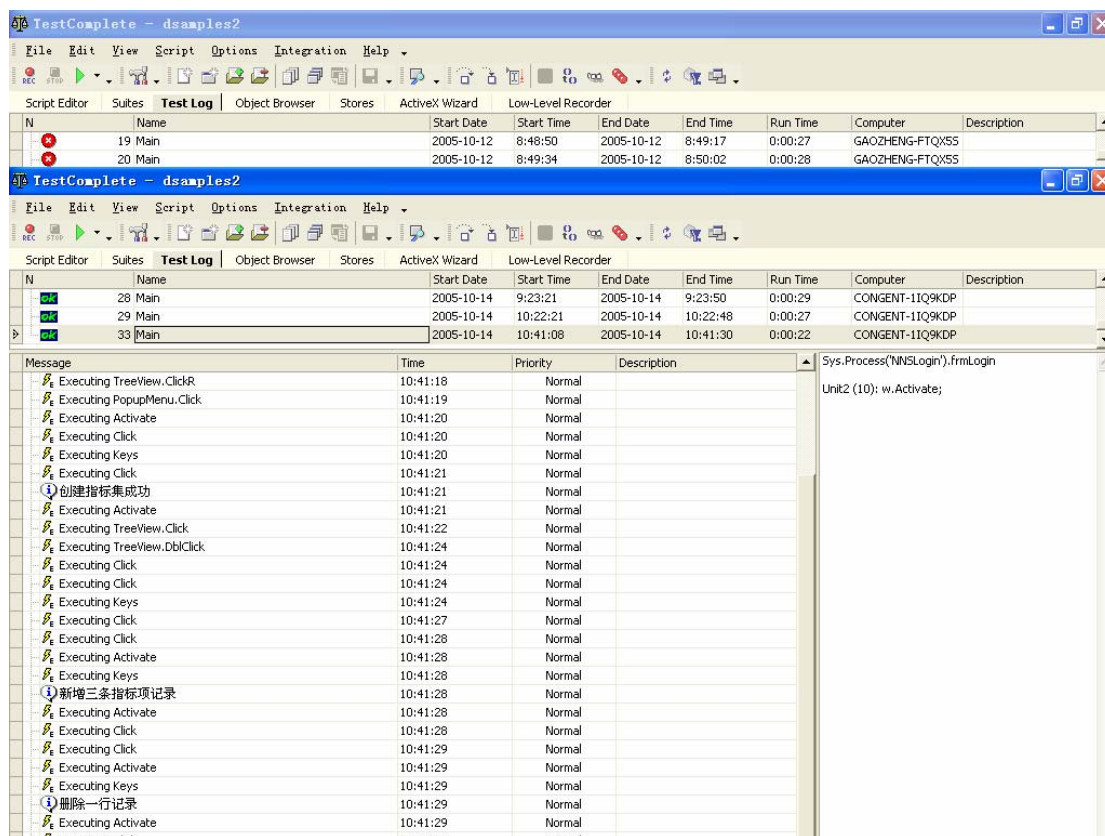
- ▲ 实例中有两个功能维护操作，删除和修改，详见上述编码。
- ▲ 每一步操作都作日志记录。

最后作回放操作，选中主 Main,点击右键，选择“run main”。如下所示：



回放各功能操作，成功后可点击 Test Log 查看操作日志，如下所示：





## 三、自动化测试实例二

### 3.1 测试实例特征

本测试实例通过分组的系列维护功能操作来重现自动化测试过程。本测试实例功能包括 app,addfz,addfa\_childitem 等脚本编写。并对成功和异常操作分别记录操作日志和异常日志。

### 3.2 测试过程

#### 1. 装载应用

略。(与实例一方法实现类似, 请参阅)

#### 2. 创建工程

略。(与实例一方法实现类似, 请参阅)

### 3. 启动应用脚本编写

略。(与实例一方法实现类似, 请参阅)

### 4. 录制功能

#### I. Addfz 脚本编写

```

procedure addfz;
var
  p, w: OleVariant;
begin
  //***** 启动进程 *****
  p := Sys.Process('NNSLogin');
  //***** 进入'用户登录'窗口 frmLogin *****
  w := p.frmLogin;
  w.Activate;
  //***** 键盘输入用户名及密码 *****
  Sys.Keys('admin[Tab]123456');
  w.btnLogin.Click(55, 19);
  //***** WINDOW 切换, 进入'南宁市决策支持与分析系统'窗口 mainlogin, *****
  //*****所以重新获取进程, 此获取很重要, 有时会找不到窗口, 如果重新 *****
  //*****获取一下进程则可 *****
  p := Sys.Process('NNSLogin');
  // Sys.Delay(1000);
  w := p.MainLogin_2;
  w.Activate;
  w.btnOK.Click(48, 6);
  //***** 进程改变, 重新获取 *****
  p := Sys.Process('NNS.SYS.Client');
  //***** WINDOW 切换, 进入主窗口 mainform *****
  w := p.MainForm;
  w.Activate;
  //***** 进入主窗口后, 默认为报表界面, 以下语法是报表 TAB 界面转 *****
  //***** 为分组 TAB 界面*****
  w.tabStrip1.Click(88, 17);
  w.trvGroup.HScroll.Pos := 0; //记录分组时鼠标轨迹
  w.trvGroup.Click('|分组'); //选中分组
  //w.trvGroup.Click('|分组'); //点击分组
  w.trvGroup.ClickR('|分组'); //点击右键作弹出窗口鼠标仍停留在分组 TAB 界面
  //***** 弹出窗口 *****
  w.PopupMenu.Click('新建文件夹');

```

```

//*****begin WINDOW 切换, 进入新增文件夹窗口 frm_InputBox_2 *****
p := Sys.Process('NNS.SYS.Client');
w := p.frm_InputBox_2;
w.Activate;
w.Object.Click(51, 7);
//*****end WINDOW 切换, 进入新增文件夹窗口 frm_InputBox_2 *****

//保持测试环境输入法按 CTRL+SHIFT 转换一次即切换到五笔环境
// sys.keys('^!');
//Sys.Keys('wv xeg imyag ');
w.txtvalue.Text:='分组测试项一';
w.btnOK.Click(49, 7);
w := p.MainForm;
w.Activate;
w.trvGroup.HScroll.Pos := 0;
w.trvGroup.ClickR('|分组|分组测试项一');
//***** 弹出窗口 *****
w.PopupMenu.Click('新建分组');
//*****begin WINDOW 切换, 进入新增分组窗口 frm_NewItem *****
w := p.frm_NewItem;
w.Activate;
w.Object_2.Click(37, 6);
w.txtcode.text := 'fzdm1'; //赋值
w.txtname.text := '分组名称一';
w.Object_2.Click(30, 8);
w.btnOK.Click(25, 4);
//*****end WINDOW 切换, 进入新增分组窗口 frm_NewItem *****
end;

```

● 本 TEST 主要方法应用:

- ▲ 赋值部分, 可用控件直接赋值来代替键盘输入, 尤其对于同时找开多个窗口时, 这个方法可以很快地找到所在的窗口并赋值。
- ▲ 其它方法如窗口切换, 输入法切换请参考上述脚本注释。

## II. Addfz\_childitem 脚本编写

```

procedure addfz_childitem;
var
  p, w: OleVariant;
begin
  p := Sys.Process('NNS.SYS.Client');
  w := p.MainForm;
  w.Activate;

  w.toolBar2.Click(8, 15);

```

```

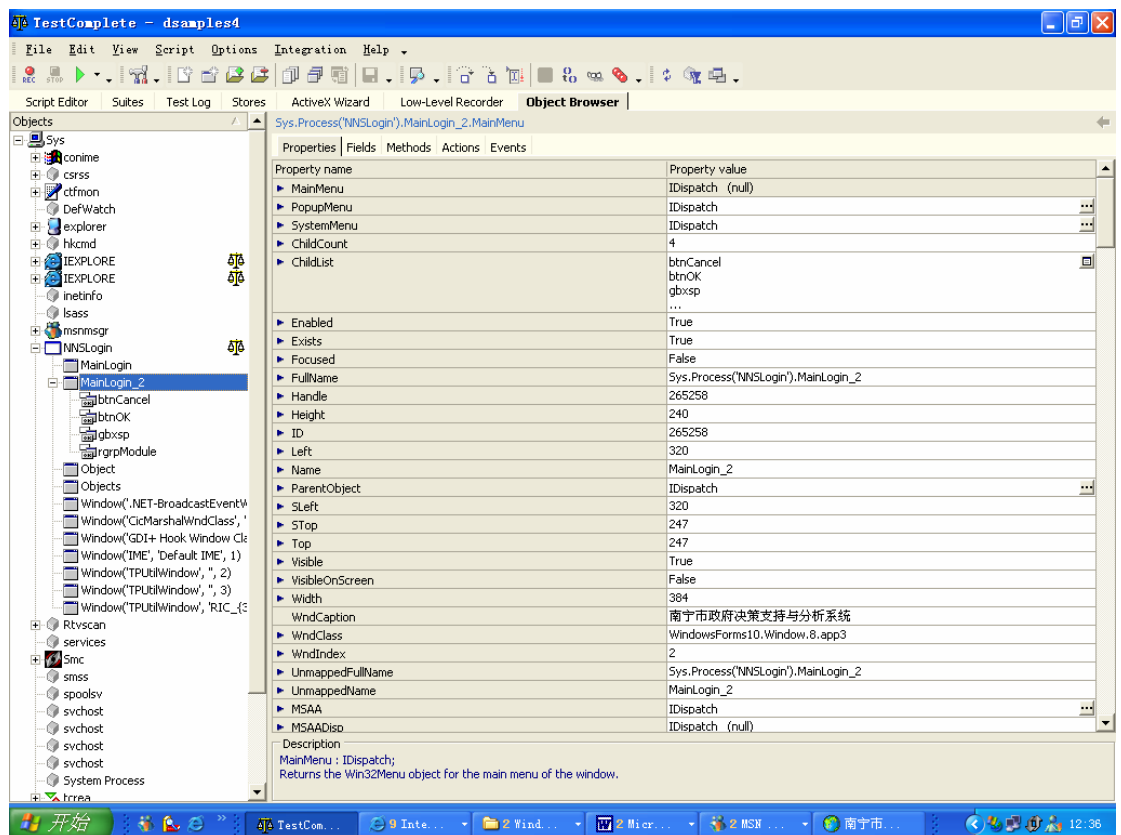
sys.delay(1000);
//***** 录制过程中可能会录入脚本如下类似对象 Object_8， 此为一个整体对
*****
w.grdControl.Click(21, 31);
//w.Object_9.Click(21, 6); ①
Sys.Keys('01[Tab]od gajfg [Tab]wh [Tab][Enter][Caps]02[Tab][Caps]od gajffg [Tab]wh
[Tab][Enter][Caps]03[Caps][Tab]od gajfdg [Tab]wh [Tab]');

//w.Object_8.Click(21, 9);
w.toolBar2.Click(68, 15);
Sys.Keys('[Enter]');

```

end;

- 本 TEST 方法主要应用：
  - ▲ 基类与子类的录制。这只是工作中的经验之谈，还不知是否完全正确。我在录制经常会录到一些脚本运行时，会出现 ‘object not found’ 信息。如果定位到对象浏览器时 (object browser),对象名称并未错误。后来我发现，原来对象名称并未错误，只是它录制了基类，如果上红色标注 ①。
  - ▲ 现在说明检查脚本时运用 object browser 的一个方法：当想查询一个窗口相应地参数设置信息时，可以打看相应窗口，然后进入 testcomplete 的 object browser 页面，然后双击其窗口名称，如下所示：



可以引用相应的参数。

▲ 其它相关方法请参考脚本注释。

## 四、 自动化测试实例三

### 4.1. 测试实例特征

本测试实例是包括对报表的维护功能操作，包括新建报表，设计报表（拖放分组，拖放指标等）。并对成功和异常操作分别记录操作日志和异常日志。

### 4.2 测试过程

#### 1. 装载应用

略。（与实例一方法实现类似，请参阅）

#### 2. 创建工程

略。（与实例一方法实现类似，请参阅）

#### 3. 启动应用脚本编写

略。（与实例一方法实现类似，请参阅）

#### 4. 录制功能

##### I. Addbb 脚本编写

//测试配置环境前提：数据库中存在分组边境贸易进出口总额  
//及指标批发零售业商品销售情况

```
procedure addbb;  
var  
    p, w: OleVariant;  
begin  
    p := Sys.Process('NNSLogin');  
    w := p.frmLogin;  
    w.Activate;  
    Sys.Keys('admin[Tab]123456');  
    w.btnLogin.Click(23, 7);
```

```

p := Sys.Process('NNSLogin');
w := p.MainLogin;
w.Activate;
w.btnOK.Click(45, 8);
// ***** begin 重启进程，打开新增报表窗口 frmEditTable_2 *****
p := Sys.Process('NNS.SYS.Client');
w := p.MainForm;
w.Activate;
w.trvBaseTable.VScroll.Pos := 0;
w.trvBaseTable.HScroll.Pos := 0;
w.trvBaseTable.ClickR('报表');
w.PopupMenu.Click('新建报表');
// ***** end 重启进程，打开新增报表窗口 frmEditTable_2 *****

// ***** begin 获得窗口 frmEditTable_2 并找到控件 *****
p := Sys.Process('NNS.SYS.Client');
w := p.frmEditTable_2;
w.Activate;
w.Object_2.Click(28, 14);
// ***** end 获得窗口 frmEditTable_2 并找到控件 *****

// ***** begin 通过控件赋值 *****
// ***** 附记，因五笔输入法现键盘切换还未找到解决方法，现用控件赋值*****
w.Object_2.wtext := 'csbbdm01';
w.Object.wtext := '测试报表名称 01';
w.cbxObject.text := '季报';
// ***** end 通过控件赋值 *****

// ***** begin 点击确定按钮，保存赋值数据 *****
p := Sys.Process('NNS.SYS.Client');
w := p.frmEditTable_2;
w.Activate;
w.btnOK.Click(24, 15);
// ***** end 点击确定按钮，保存赋值数据 *****

// ***** begin 点击保存按钮，保存报表初始分数据 *****
p := Sys.Process('NNS.SYS.Client');
w := p.MainForm;
w.Activate;
w.toolBar.Click(14, 15);
Sys.Keys(['Enter']);
// ***** end 点击保存按钮，保存报表初始分数据 *****

// ***** begin 点击解锁按钮，使 toolBar 各按钮可用 *****

```

```

p := Sys.Process('NNS.SYS.Client');
w := p.MainForm;
w.Activate;
w.toolBar.Click(80, 15);
// ***** end 点击解锁按钮, 使 toolBar 各按钮可用 *****
end;

```

●本 TEST 方法操作报表新建及 TOOLBAR 菜单按钮录制的各功能操作。可见每操作步骤脚本注释。

## II. bbdesign 脚本编写

```

procedure bbdesign;
var
  p, w: OleVariant;
begin
  p := Sys.Process('NNS.SYS.Client');
  w := p.MainForm;
  w.Activate;
  //***** begin 拖入分组'边增贸易进出口总额' *****
  w.tabStrip1_2.Click(94, 9);
  w.trvGroup.HScroll.Pos := 0;
  w.trvGroup.Click('|分组|边增贸易进出口总额');
  w.trvGroup.Drag(118, 309, 398, 22);
  w := p.frmInutDialog;
  w.Activate;

  //***** end 拖入分组'边增贸易进出口总额' *****

  //***** begin 输入分组表名称 *****
  w.txtTableName.Click(27, 11);
  Sys.Keys('888');
  w.btnOK.Click(36, 5);
  //***** end 输入分组表名称 *****

  //***** begin 保存所做修改 *****
  // p := Sys.Process('NNS.SYS.Client');
  // w := p.MainForm;
  // w.Activate;
  // w.toolBar.Click(7, 14);
  // Sys.Keys(['Enter']);
  //***** end 保存所做修改 *****

  //***** begin 调整分组位置 *****
  p := Sys.Process('NNS.SYS.Client');

```

```

w := p.MainForm;
w.Activate;
w.Object_6.Click(59, 9);
w.Object_6.Drag(59, 9, -210, -75);①
//***** end 调整分组位置 *****

//***** begin 切换 TAB 页至指标页 *****
  p := Sys.Process('NNS.SYS.Client');
  w := p.MainForm;
  w.Activate;
w.tabStrip1_2.Click(50, 12);
//***** end 切换 TAB 页至指标页 *****

//***** begin 展开指定指标集并加入相应分组 *****
  p := Sys.Process('NNS.SYS.Client');
  w := p.MainForm;
  w.Activate;
w.trvGuidLine.HScroll.Pos := 0;
w.trvGuidLine.Expand('指标集|批发零售业商品销售情况');
w.trvGuidLine.HScroll.Pos := 0;
w.trvGuidLine.Click('指标集|批发零售业商品销售情况|代码');
w.trvGuidLine.Drag(91, 55, 443, 211);

//***** end 展开指定指标集并加入相应分组 *****
end;
```

- 本 TEST 主要方法实现:

▲*obj.Drag(ClientX, ClientY, toX, toY, Shift)*, 此事件可记录一个轨迹的起始点至终结点的位置坐标, 可以展现动作的轨迹。

### III. Addzbitem 脚本编写

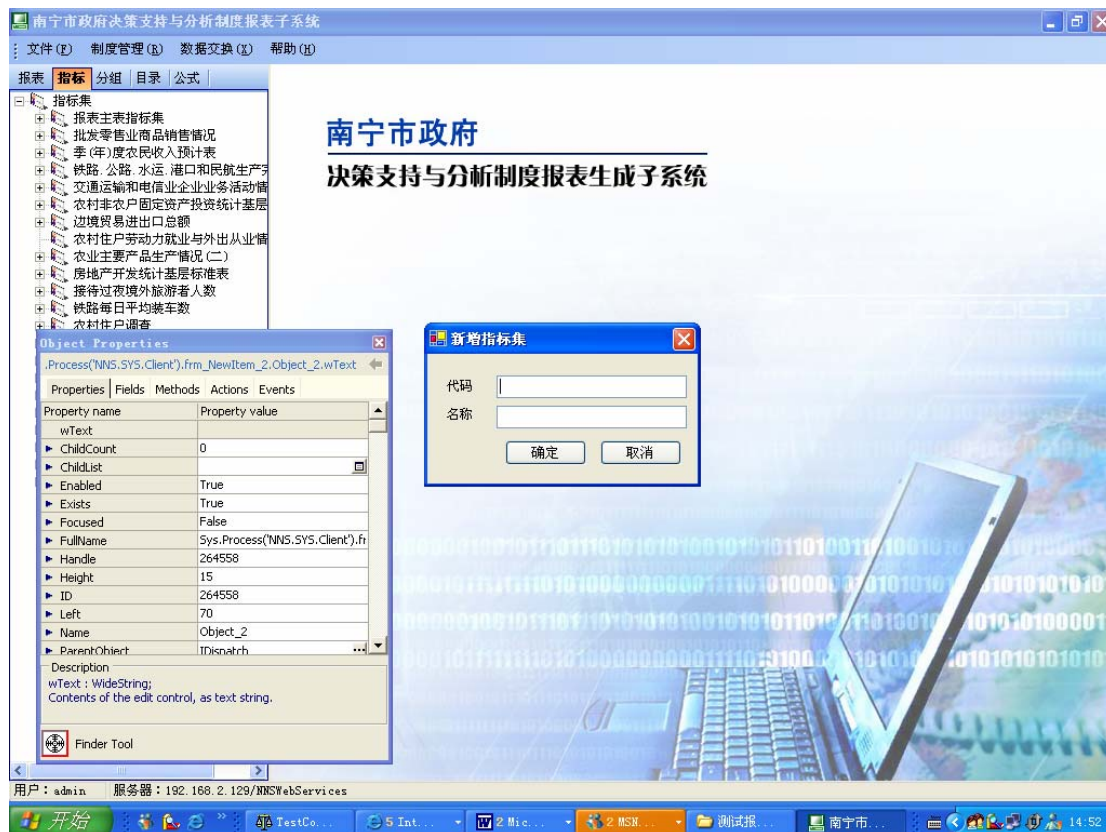
```

procedure addzbitem;
var
  p, w: OleVariant;
begin
  p := Sys.Process('NNS.SYS.Client');
  w := p.MainForm;
  w.Activate;
w.tabStrip1_2.Click(49, 13);
//w.trvGuidLine.HScroll.Pos := 0;
w.trvGuidLine.wSelection := '指标集|批发零售业商品销售情况';
//w.trvGuidLine.Expand('指标集|批发零售业商品销售情况');
w.trvGuidLine.Click('指标集|批发零售业商品销售情况|代码');
w.trvGuidLine.Drag(88, 55, 456, 214);
```



end;

- 本 TEST 主要方法实现：
  - ▲ *obj.expand()* 事件。
  - ▲ 现在说明可以引用 Object Properties 的方法检查对象属性。当定位对象时，可以点选 View→Object Properties,然后点选 Finder Tool,拖动鼠标至相应对象，可更改其属性值来检查录制点是否正确。如下图所示：



这样就可以通过其属性值来确定录制时代码是否正确。

## 五、测试小结

对 TestComplete 熟悉使用的过程中发现其功能非常丰富，虽然现在可以借荐的参阅资料还是很少，另外它有很多帮助写测试脚本的页面工具，比如，object browser,object propertites 及 components palette 等。相信如果可以熟练驾驭它后，它将会成为一个很得力的测试助手。

## 六、附录

