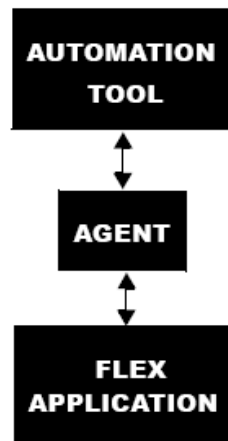


TIB 自动化测试工作室

<http://www.cnblogs.com/testware>

使用 QTP 进行 Flex 自动化测试

自动化测试工具与 Flex 应用程序之间是通过代理（Agent）来交互的，如图所示：



这也就是为什么后面在使用 QTP 进行 Flex 自动化测试之前要先安装 Flex 插件、配置和编译 Flex 程序的原因。

前提条件

在使用 QTP 进行 Flex 4 的测试之前，需要安装以下软件：

- 1、HP QuickTest Professional 10（支持 Internet Explorer 7 或 Internet Explorer 8）或者 HP QuickTest Professional 9.5（支持 Internet Explorer 6 或 Internet Explorer 7）
- 2、Adobe Flex 4 Plug-in for Mercury QuickTest Pro
- 3、Microsoft Internet Explorer（6 以上版本）
- 4、Flash Player ActiveX control，Flash Player 的版本要求与 Flex SDK 的一致（Flex SDK 的系统要求参见<http://www.adobe.com/products/flex/systemreqs>）

安装 Flex 插件

- 1、下载 Adobe Flex 4 Plugin for HP QuickTest Pro

下载地址：

http://download.macromedia.com/pub/labs/flex/4/flex4_automation_install_100509.zip

- 2、解压下载到的文件 flex4_automation_install_100509.zip

双击解压后文件夹中的 Install_QTP_Plugin.bat 进行安装。

- 3、安装后，打开 QTP，在启动界面的插件管理器中可以看到新添加的 Flex 和 AIR 插件，

如图所示。



编译 Flex 程序

在使用 QTP 加载 Flex 插件对 Flex 程序进行测试之前，需要对 Flex 程序进行特定的编译，否则 Flex 控件将不被 QTP 所识别。下面以 Flash Builder 4 为例，新建一个 Flex 程序并进行相关配置和编译。

- 1、打开 Flash Builder 4，新建一个 Flex 项目，选择应用程序类型为 Web。
- 2、新建 Flex 项目后，在项目名右键单击，选择属性，然后选择“Flex 编译器”，打开如图所示的界面。



- 3、在“附件的编译器参数”输入框中添加如下参数：

-include-libraries "C:\Program Files\Adobe\Adobe Flash Builder 4\sdk\4.1.0

```
\frameworks\libs\automation\automation.swc" "C:\Program Files\Adobe\Adobe Flash Builder 4
\sdk\4.1.0\frameworks\libs\automation\automation_agent.swc" "C:\Program Files\Adobe\Adobe
Flash Builder 4\sdk\4.1.0\frameworks\libs\automation\qtp.swc" "C:\Program Files\Adobe\Adobe
Flash Builder 4\sdk\4.1.0\frameworks\libs\automation\automation_dmv.swc"
```

4、打开 Flash Builder 4 安装目录中的“C:\Program Files\Adobe\Adobe Flash Builder 4\sdk\4.1.0\templates\automation-runtime-loading-files”目录，运行目录中的“build.bat”批处理文件。

5、在相同的目录下，找到“runtime-loading.mxml”文件，打开并拷贝其中的内容，粘贴到新建项目的 mxml 文件中，文件内容如下所示：

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute"
creationComplete="actionScriptFunction()" >
  <mx:Script>
    <![CDATA[
      import flash.external.*;

      public function init():void
      {
        myLoader.addEventListener(IOErrorEvent.IO_ERROR, ioErrorHandler);
      }
      private function ioErrorHandler(event:IOErrorEvent):void {
        trace("ioErrorHandler: " + event);
      }
      public function actionScriptFunction():void
      {
        init()

        myLoader.source = Application.application.parameters.automationswfurl;
      }
    ]]>
  </mx:Script>

  <mx:SWFLoader id="myLoader" width="100%" height="100%"
preinitialize="myLoader.loaderContext = new LoaderContext(false,
ApplicationDomain.currentDomain)" >
  </mx:SWFLoader>

</mx:Application>
```

6、在 Flash Builder 中设计程序界面并添加代码，如下所示：

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute"
```

```
creationComplete="actionScriptFunction()" >
```

```
<mx:Script>
  <![CDATA[
    import flash.external.*;

    public function init():void
    {
        myLoader.addEventListener(IOErrorEvent.IO_ERROR, ioErrorHandler);
    }
    private function ioErrorHandler(event:IOErrorEvent):void {
        trace("ioErrorHandler: " + event);
    }
    public function actionScriptFunction():void
    {
        init()

        myLoader.source = Application.application.parameters.automationswfurl;
    }

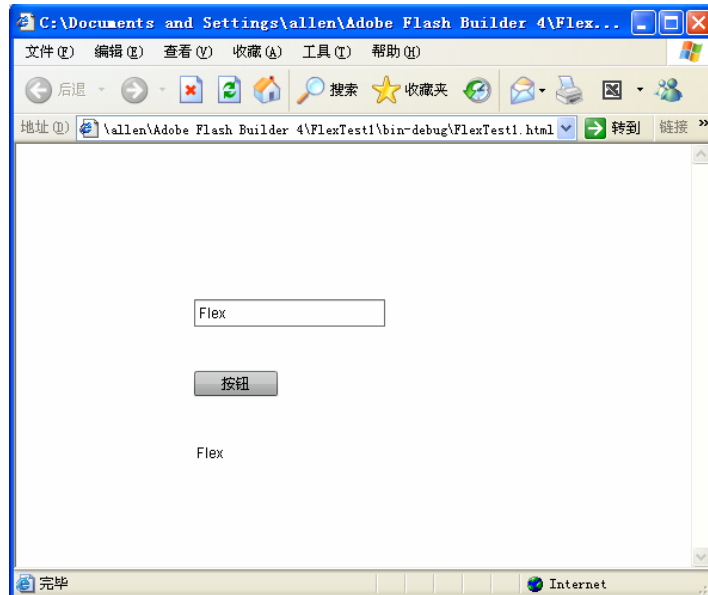
    protected function button1_clickHandler(event:MouseEvent):void
    {
        // TODO Auto-generated method stub
        label1.text = textInput1.text ;
    }

    ]]>
</mx:Script>

<mx:SWFLoader      id="myLoader"          width="100%"    height="100%"
preinitialize="myLoader.loaderContext    =    new    LoaderContext(false,
ApplicationDomain.currentDomain)" >
  </mx:SWFLoader>
  <mx:Button      x="149"      y="190"      label=" 按 钮 "      id="button1"
click="button1_clickHandler(event)"/>
  <mx:Label x="149" y="249" text="标签" id="label1"/>
  <mx:TextInput x="149" y="130" id="textInput1"/>

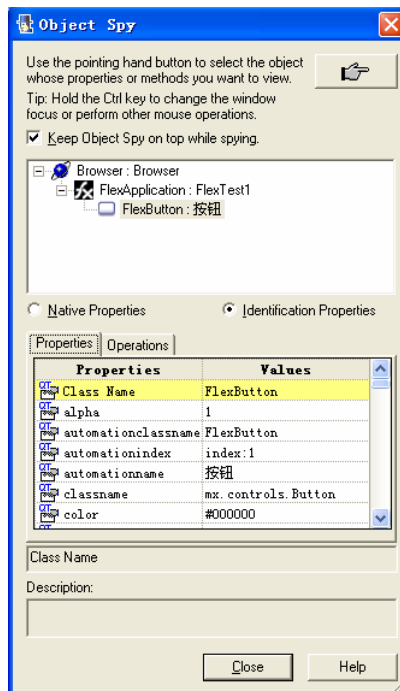
</mx:Application>
```

最后编译生成如图所示的 Flex 程序：



测试 Flex 程序

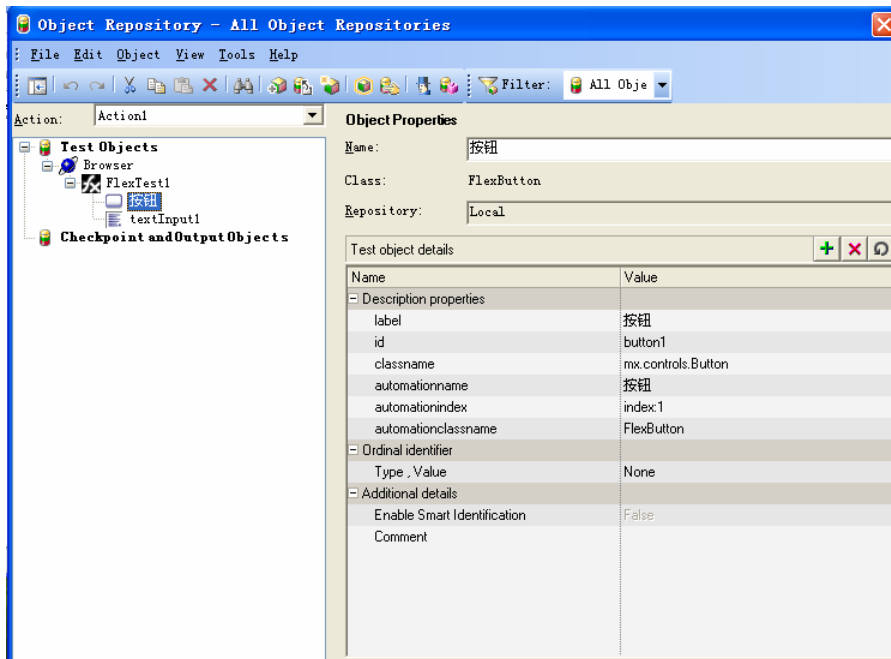
经过前面的编译和配置之后，打开 QTP，加载 Flex 插件和 Web 插件，然后打开刚才编译的 Flex 程序，用 Object Spy 可以看到 Flex 程序中的控件已经可以被识别出来了，如图所示。



接着就可以用 QTP 对 Flex 程序进行录制，得到如下所示的脚本：

```
Browser("Browser").FlexApplication("FlexTest1").FlexTextArea("textInput1").SelectText 0,0  
Browser("Browser").FlexApplication("FlexTest1").FlexTextArea("textInput1").Input "Flex"  
Browser("Browser").FlexApplication("FlexTest1").FlexButton("按钮").Click
```

在录制脚本过程中，QTP 将往对象库中添加几个 Flex 控件，如图所示：



把上述脚本回放，将得到如图所示的结果：



至此，我们可以看到使用 QTP，在添加 Flex 插件和对 Flex 程序进行配置编译的前提下，可以识别 Flex 控件，对 Flex 程序进行录制和回放。

QTP 的其他功能对于 Flex 程序的测试而言同样适用，例如 CheckPoint（检查点）、数据表（DataTable）等，例如下面是对 Label 添加了检查点后的脚本：

```
Browser("Browser").FlexApplication("FlexTest1").FlexTextArea("textInput1").SelectText 0,0
Browser("Browser").FlexApplication("FlexTest1").FlexTextArea("textInput1").Input "Flex"
Browser("Browser").FlexApplication("FlexTest1").FlexButton("按钮").Click
Browser("Browser").FlexApplication("FlexTest1").FlexLabel("Flex").Check CheckPoint("Flex")
```

Flex 3 自动化测试

如果是对 Flex 3 的程序进行测试，则需要安装 Flex 3 的 QTP 插件，Flex 3 插件可以在 Flex Builder 3 的安装目录下找到。通过运行 Flex Builder 3 安装目录下的 \Installers\QTP_Plugin_Installer.exe 来安装 QTP 插件。安装完后启动 QTP，在 Add-ins Manager 窗口中将出现 Flex 3.0.0 插件。

对被测试的 Flex 应用程序进行编译设置的方法与前面所述类似，而且更为简单，只需要打

开 Flex Builder, 右键选择 Project -> Properties -> Flex Compiler, 修改属性“ Additional compiler arguments”, 在该属性中添加如下代码:

```
-locale en_US -include-libraries "C:\Program Files\Adobe\Flex Builder 3\sdk\3.2.0\frameworks\libs\automation.swc" "C:\Program Files\Adobe\Flex Builder 3\sdk\3.2.0\frameworks\libs\automation_agent.swc" "C:\Program Files\Adobe\Flex Builder 3\sdk\3.2.0\frameworks\libs\qtp.swc" "C:\Program Files\Adobe\Flex Builder 3\sdk\3.2.0\frameworks\libs\automation_dmv.swc"
```

TIB 自动化测试工作室

<http://www.cnblogs.com/testware>