



# **SmartSWT**

## 用户手册

版本 <3.1>

## 目录

1. 简介	3
1.1 SmartSWT 介绍	3
1.2 SmartSWT 与其它 RIA 平台的对比	4
2. 运行环境	5
3. 快速入门	5
3.1 安装	5
3.2 运行	6
3.3 快速开发 SmartSWT 应用——HelloWorld	9
3.3.1 开发一个普通 SWT 应用	9
3.3.2 普通 SWT 应用—>SmartSWT 应用	10
3.3.3 运行	11
3.4 其它注意事项	12
4. 联系方式	12
4.1 业务联系	12
4.2 商务合作	13

# 用户手册

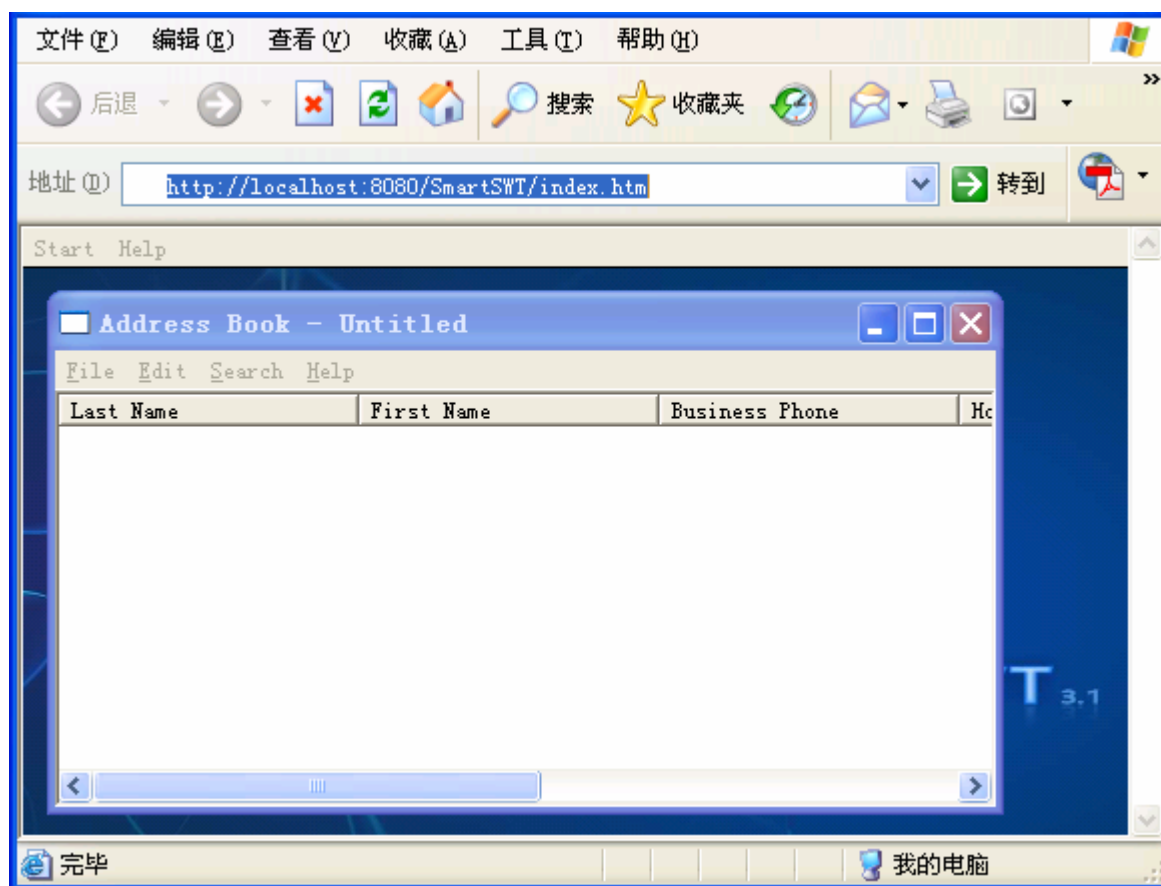
## 1. 简介

### 1.1 SmartSWT 介绍

SmartSWT 是 RIA 开发平台，允许你使用 Eclipse 的 SWT 控件开发 WEB 应用程序。标准的 SWT 应用程序几乎不需要任何修改，就可直接发布到 SmartSWT 平台成为 B/S 模式的 WEB 应用程序，客户端可通过浏览器访问该程序。

简言之，SmartSWT 平台可将一个普通的 SWT 应用程序，直接部署成为一个 WEB 应用程序。

以下是一个 SWT 应用（Address Book）发布到 SmartSWT 后，在客户端使用 IE 浏览器访问的结果：



SmartSWT 所指的标准 SWT 应用程序是指符合以下条件的 SWT 应用程序：

1. 纯 JAVA 开发的完整独立的 SWT 桌面应用程序
2. 只使用 SWT 及其扩展 API（JFace、GEF 等）做为 UI 表现
3. 没有使用其它 JNI 本地调用技术，或使用了 JNI 技术或 ActiveX 组件，但不做为 UI 组件使用，而且发布为 WEB 应用后仅在服务器端调用，不涉及对客户端操作。后者需测试。

使用以下技术的 SWT 应用程序不能发布为 WEB 应用程序，或可能会有问题：

1. 使用了其它 UI 组件，如：SWING，AWT、ActiveX 等
2. 暂不支持 Eclipse 的 RCP 应用程序（RCP 应用使用了 JNI 技术）

## 1.2 SmartSWT 与其它 RIA 平台的对比

B/S 应用与 C/S 应用相比，B/S 应用的主要缺点是：

1. 客户端 UI 表现能力不足
2. 客户端表现主要基于 HTML、XML 之类的脚本语言，而桌面应用开发基于事件驱动，而且有大量成熟的可视化设计工具，如 Delphi，VS.net 等。相比之下 WEB 应用开发的可视化设计工具更多是设计 UI，而较难把业务逻辑与 UI 集成开发。

而 C/S 的主要问题是：

1. 安装部署不便
2. 升级及客户端程序的维护不方便
3. 需要自己定义客户端和服务端间的通讯协议、数据交换，开发者需要了解 TCP/IP 之类的协议，开发复杂度较高。

为了克服各自的缺点，出现了一些如 SmartClient、Flex 等 RIA 技术，从各自不同的方向对两者进行了改进。

微软的 SmartClient 技术是从 C/S 方向出发改进了 C/S 应用程序部署方式。但其本质上仍然是 C/S 模式的应用，应用程序需要自己维护客户端和服务端的通讯，业务逻辑分布于客户端和服务端，开发复杂度和维护仍然有不足。

Macromedia 的 Flex 技术则从 B/S 方向出发改进 B/S 应用的 UI 表现，它利用 Flash 出众的表现能力做为 UI 表现层，其它则与传统的 B/S 应用开发保持一致，较好的解决了 B/S 的不足。但也有一些问题，如：需要客户端安装一个 Flash 插件，这个问题不大；但开发需要使用基于 XML 的一套专有标记语言，仍然是脚本方式开发，增加了开发复杂度，而且 Flex 的脚本似乎不支持动态控件创建，比如不能临时根据程序逻辑创建一个未定义过的窗口等，造成应用开发上有一定的局限性。

还有如 Bindows 直接在 HTML 基础上进行 CS 界面的模仿，也存在一些性能问题，而且许多控件的实时事件不能完全模拟，表现力与真实的 CS 界面仍然有较大差距。

其它一些技术方案或多或少与上述类似，完美的方案目前阶段尚不存在，任何对 B/S 或 C/S 进行改良的方案，或多或少都要在其它方面付出相应的代价。

本工作室的 SmartSWT 也是从 B/S 方向出发对传统 B/S 模式的一种改良，相比较而言，它与上述方案相比有以下优点：

1. 拥有和 C/S 应用一样的 UI 表现：因为 SWT 控件本身就是使用系统本地控件，实际上使用的是和桌面应用同一个控件。
2. 拥有 C/S 一样的开发方式，使用基于事件驱动的开发方式：SmartSWT 应用本身就是一个标准的 SWT 应用，可以使用成熟的 SWT 可视化开发工具（如：Designer），完全不需要与脚本打交道，开

发人员只需要掌握一种开发技能——SWT 开发技术即可。

3. 拥有和 B/S 一样的部署方式，基于 WEB 访问：开发人员只需关注于用 JAVA+SWT 实现业务逻辑即可，不需同时开发客户端和服务端两套应用，只需按桌面应用的开发方式实现逻辑并部署在服务端，则其它客户端只需通过 IE 浏览器即可访问，不需要维护客户端和服务端的通讯，不需了解 TCP/IP 等通讯协议。

简言之，SmartSWT 技术能让你的标准 SWT 桌面应用程序直接部署成为一个基于 B/S 模式的 WEB 应用程序。

如前所述，任何改良均需付出其它方面的妥协，SmartSWT 也不例外：

1. 客户端的浏览器需要安装一个 SmartSWTClient 的 ActiveX 插件，这点和 Flex 一样，Flex 需要在浏览端安装一个 Flash 播放器插件。
2. 浏览器目前暂只支持 IE5.0 以上浏览器，同时这也限制了客户端必须是 Windows 系统，但服务端并没有限制，可以运行在任何支持 JAVA 的操作系统上。
3. SmartSWT 对网络条件要求较高，要有很好的网络数据传输速度，需要有 2M 以上的带宽，所以它必须运行在局域网内，而且最好是有线局域网，在无线局域网性能表现不佳。

如果您的应用是基于内网的企业应用，那么以上问题将不会成为障碍，SmartSWT 将是您 RIA 开发的上佳选择。

## 2. 运行环境

**服务器：**Windows、Linux、Unix、MacOS 等支持 JAVA 的操作系统。1G 以上主频的 CPU，如果要求支持几十上百个客户端同时访问的话，最好有 1G 甚至更多的内存。

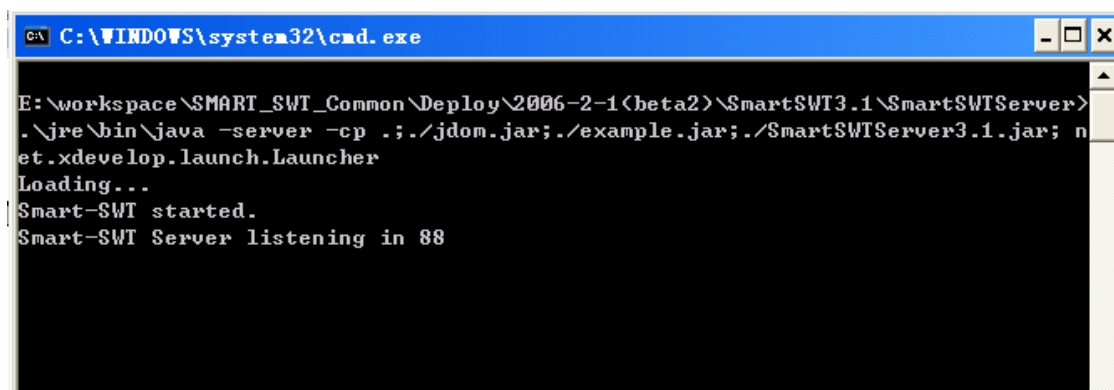
**客户端：**配有 IE5.0 以上的浏览器的 Windows 系统。PIII 以上 CPU，最好有 256M 以上内存。

## 3. 快速入门

### 3.1 安装

以 Windows 系统下的安装为例：

1. 从<http://www.smartswt.net>下载 SmartSWT3.1，并解压。
2. 在客户端机器上，运行 SmartSWT3.1/SmartSWTClient/目录下的 Setup.exe 安装客户端，该安装程序将在客户端安装一个特别定制的小型 JAVA 虚拟机。
3. 在服务端运行 SmartSWT3.1/SmartSWTServer/目录下的 startup.bat，稍等数秒后将出现以下终端界面：



SmartSWTServer 默认需要使用 88 端口做为通讯端口，如果有防火墙，需注意将该端口开放。

4. 如果你只是在本机上试用，服务端和客户端都是同一机器，则直接用 IE 打开 SmartSWT/web/目录下的 index.htm 即可，可略过下面 5、6、7 的步骤。
5. 打开 SmartSWT3.1/web/目录下的 index.htm:

```
<HTML>
<body topmargin="1" leftmargin="1" rightmargin="1" bottommargin="1"
<OBJECT classid="clsid:53FEFD3F-F82F-4696-B977-63ADC75F1B29"
    width=100%
    height=100%
    align=center
    hspace=0
    vspace=0
>
<!-- please change the value of Host parameter to your server name
<param name="Host" value="localhost">
<param name="Port" value="88">
</OBJECT>
</body>
</HTML>
```

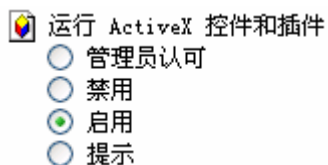
将其中的 value="localhost" 中的 localhost 改为实际运行 SmartSWT 服务端的服务器名称或 IP，如 value="192.168.0.1"。

6. 将改好的 index.htm 放到 WEB 服务器下，如 IIS、Tomcat、WebSphere、Weblogic 等，这里假设放于服务根目录的/SmartSWT/下。
7. 启动 WEB 服务器。

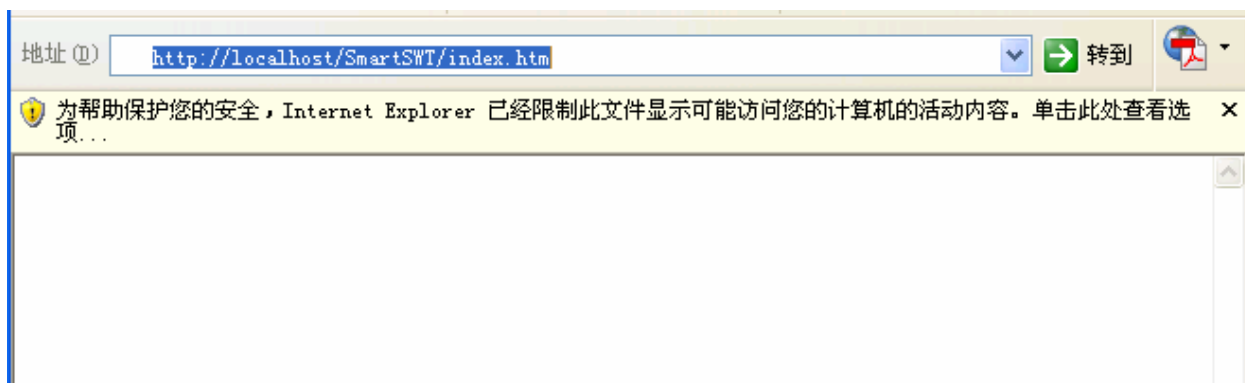
至此 SmartSWT 安装完毕。

### 3.2 运行

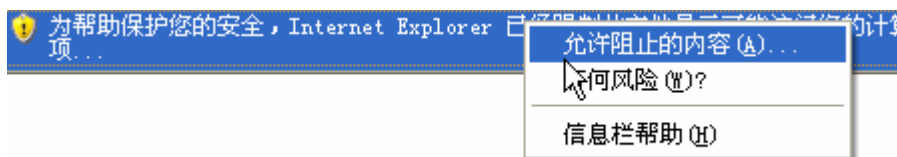
1. 打开 IE，<http://服务器/SmartSWT/index.htm>，这里假设您的 WEB 服务器的 WEB 服务使用的是 80 端口。
2. 请确认你的 IE 浏览器的安全级别使用的是默认的中等安全设置，或者确认该安全级别是允许运行 ActiveX 控件的，如下图：



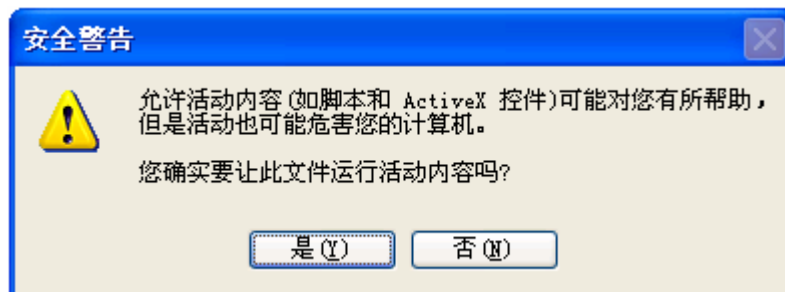
这时，浏览器通常会有如下安全提示（下图以 Windows XP + SP2 + IE6.0 为例）：



请选择允许阻止的内容：



并对随后的安全警告选择确认按钮：



（本工作室多年来为三百家政府部门、金融机构和大中小企业提供可靠的软件，保证 SmartSWT 不含任何恶意性质的代码，您尽可放心试用。）

3. 确认安全警告框后，浏览器中将出现以下信息：

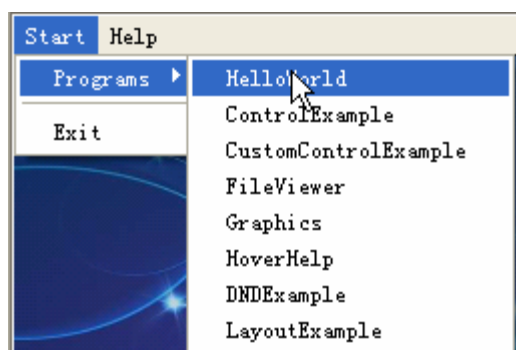
SmartSWTClient will start in 15 seconds, wait please...

大约需要 15 秒的时间启动 SmartSWT，如果你的 CPU 主频较低，或是服务端被第一次访问的话，启动时间可能会稍长点。

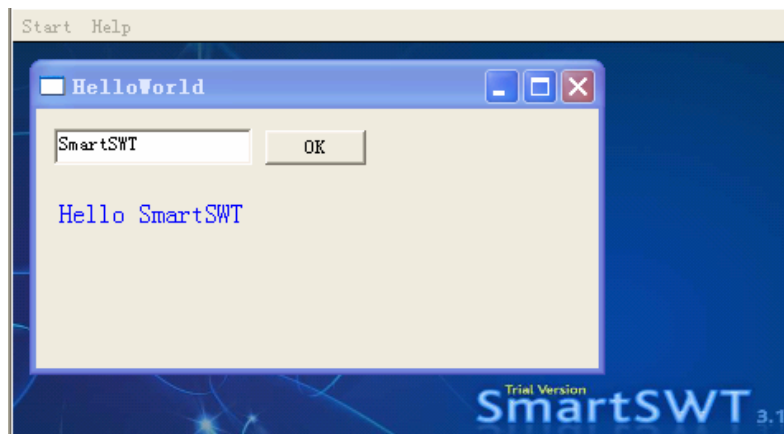
4. 启动后会出现以下界面：



5. 运行内置的 Demo 应用程序，打开 Start->Programes->HelloWorld:



运行效果如下:





6. 试运行其它内置 DEMO 应用程序。

### 3.3 快速开发 SmartSWT 应用——HelloWorld

#### 3.3.1 开发一个普通 SWT 应用

推荐使用 Eclipse3.1 和 Designer4.0 以上版本来开发 SWT 应用。该软件是商业软件，只有一个月的试用期，也可使用 Eclipse 组织自己开发的 VisualEditor 进行 SWT 应用的开发，只是这个编辑器的速度要慢些。

SmartSWT3.1 支持用 SWT 3.1 开发的 SWT 应用或与 SWT 3.1 兼容的应用。兼容的意思是说，不论您的 SWT 应用程序采用哪个版本开发，如果可以在 SWT 3.1 环境下正确运行，则谓之兼容。

1. 在Eclipse下用您喜欢的方式创建一个SWT应用程序，如果您想省事的话，可以直接访问 <http://www.eclipse.org/swt/snippets/> 下载该页面的Hello World链接的Snippet1.java代码。
2. 请注意用红字部分的代码替换原代码，要发布到 SmartSWT 上的应用必须使用 `Display display = Display.getDefault()` 的方式获得 Display 对象，并且最好对创建的 Shell，使用 `setSize()` 给予一个初始的尺寸，代码如下：

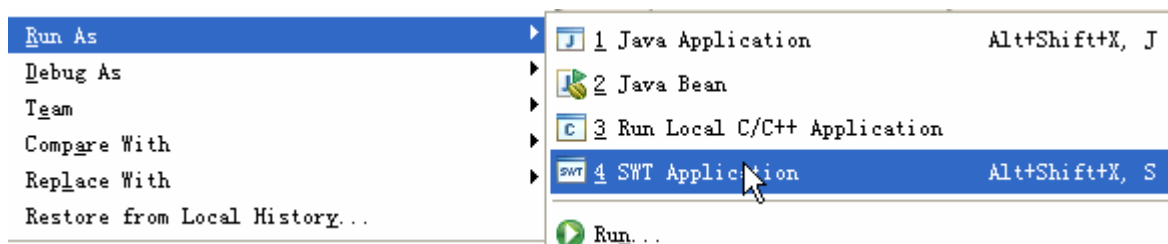
```
package org.eclipse.swt.snippets;

/*
 * example snippet: Hello World
 *
 * For a list of all SWT example snippets see
 * http://www.eclipse.org/swt/snippets/
 */
import org.eclipse.swt.widgets.*;

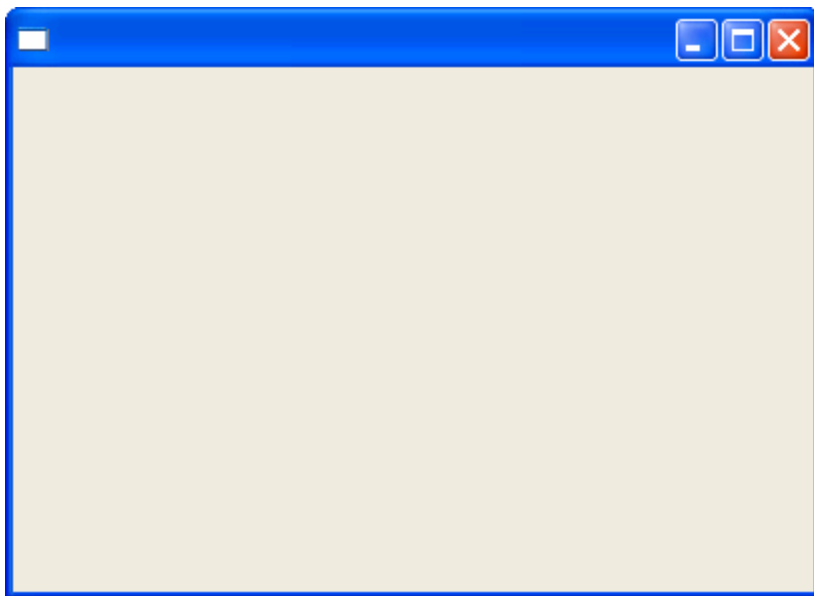
public class Snippet1 {

    public static void main (String [] args) {
        Display display = Display.getDefault(); // replaced: Display display = new Display();
        Shell shell = new Shell(display);
        shell.setSize(400,320); // add this code to give it a default size
        shell.open ();
        while (!shell.isDisposed ()) {
            if (!display.readAndDispatch ()) display.sleep ();
        }
        display.dispose ();
    }
}
```

3. 在 Eclipse3.1 中右键点击该类文件，运行该 SWT 应用，如下图：



这是运行效果图：



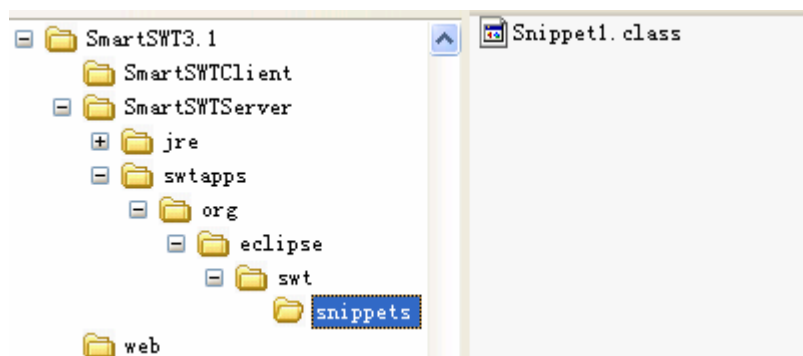
至此完成了最简单的一个 SWT 应用的开发。

### 3.3.2 普通 SWT 应用—>SmartSWT 应用

要使普通的 SWT 应用能够让客户端通过浏览器访问，必须将其部署到 SmartSWTServer 上，使其成为一个 SmartSWT 应用。

#### 1. 拷贝：

将上述开发好的 SWT 应用程序的 Class 文件拷贝至 SmartSWTServer 目录下的 swtapps 目录下，请注意你的包名目录要一并拷贝或创建，如上节的 SWT 应用程序拷过来后，目录文件结构应该是：



#### 2. 修改配置：

用编辑器打开 SmartSWT/SmartSWTServer 目录下的 SmartSWTServer.xml 文件，通过拷贝、粘贴并修改原来的<Application>...</Application>配置的内容，在末尾增加下图中反显部分内容：

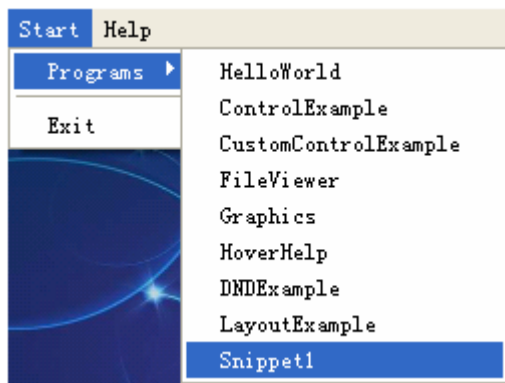
```
<Application name="LayoutExample" mainClass="org.eclipse.swt.examples.layout
  <startup>false</startup>
  <description>LayoutExample</description>
</Application>

<Application name="Snippet1" mainClass="org.eclipse.swt.snippets.Snippet1">
  <startup>false</startup>
  <description>Snippet1</description>
</Application>
</Programs>
</Desktop>
</SmartSWTServer>
```

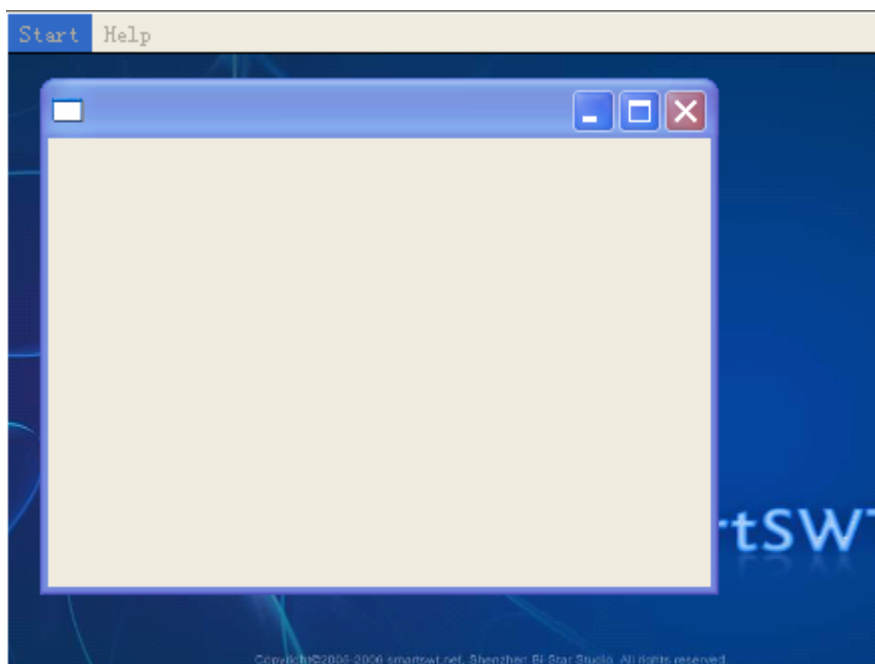
3. name: 表示应用程序的名称,  
mainClass: 表示该 SWT 应用程序的启动类,  
startup: 表示是否在客户端启动 SmartSWT 桌面时自动运行（测试版该设置无效）,  
description: 是对该应用的进一步详细描述, 起注释作用。  
检查无误后, 保存配置文件。

### 3.3.3 运行

1. 点击 SmartSWT3.1/SmartSWTServer/startup.bat 重新运行 SmartSWTServer, 如果之前已经启动, 请先关闭后再启动, 可在终端窗口连续两次 Ctrl+C 中止运行。
2. 通过IE访问: <http://server/smartswt/index.htm>, 如果在本机测试可直接用IE打开 SmartSWT3.1/web/index.htm。
3. 启动后, 在 Start->Programs 菜单可发现新增了一个名为 Snippet1 的应用, 点击它:



在 SmartSWT 客户端的运行效果图如下:



### 3.4 其它注意事项

1. SmartSWT 应用必须使用 `Display display = Display.getDefault()` 的方式获得 Display 对象
2. 最好对创建的 Shell, 使用 `setSize()` 给予一个初始的尺寸
3. 不支持使用多个 Display 的应用
4. SmartSWT 对 SWT 进行了改写, SWT 中注明的非公开的 API 接口可能不能正常在 SmartSWT 下使用, 如 OS 类的 API 等, 如果一定要用, 需部署到 SmartSWTServer 下进行测试
5. 不要使用除 SWT 外的其它 UI 组件, 如 Swing、AWT 等
6. 不要使用其它 ActiveX 的 UI 组件, 可以使用纯业务逻辑处理的 ActiveX 组件, 但需实际测试
7. 最好不要使用其它 JNI 技术
8. 测试版暂不支持 FileDialog 打开文件, 暂不支持在客户端打开文件
9. 不支持 RCP 类的应用, 不要试图在 SmartSWT 中运行 Eclipse 这样程序
10. 不要在 SWT 应用中一下子创建太多的 UI 组件, 应采取延迟初始化的方式, 在需要时创建, 否则将影响在 SmartSWT 中启动的速度
11. 测试版只支持 3-5 个客户端
12. 测试版有使用时间及其它限制

## 4. 联系方式

版权所有: 深圳双子星工作室

网 址: <http://www.smartswt.net/>

### 4.1 业务联系

联系人: 程先生

电 话: 13067369952

邮 箱: [sales@smartswt.net](mailto:sales@smartswt.net)

MSN: [longlybug@msn.com](mailto:longlybug@msn.com)

QQ: 33450

#### 4.2 商务合作

联系人: 黄先生

电 话: 13058038663

邮 箱: [tech@smartswt.net](mailto:tech@smartswt.net)

MSN: [smartswt@hotmail.com](mailto:smartswt@hotmail.com)

QQ: 2957451