

flexpaper

Flexpaper

二次开发入门教程

发布: ajava.org
编写: mark

收藏本站 Java技术QQ群

DIY

AJAVA

独立Java社区

mark 在线  QQ帐号绑定 | 设置 | 消息 | 提醒 | 门户管理 | 管理中心 | 退出

积分: 2359 | 用户名: 管理员 

首页

技术论坛

我的博客

朋友圈

资源中心

原创教程

新鲜博文

源码实例

在线文档

行业资讯

AJava-独立Java社区

 请输入搜索内容

文章

搜索

热搜: Java struts spring hibernate mysql

一小撮不明真像的群众正在围观XXX

 way

 mark

 小末、g

 RealDream

 fisherhe

 chn8088

 没有烦恼...

 klping

 db792309

 tcy901209

 wzl

 ant

 慕0阿的糖

 colinbu

站内动态

本周话题

公告:本站会员采用QQ+实名+照片注册,请朋友们准备好你的美颜,用QQ登录注册(点击),老站用户需要重新注册,保管好自己的AJava账户和密码。若丢失密码可联系我们。(访问AJava时可以选择QQ登录或者本站会员登录) -- 2011-12-04

详细...

切勿:网页的登录系统怎么做的??求解 (5 小时前)

mark: 天气太冷了,不想去拜年 (昨天 11:47)

way: 写程序敲代码和发微博一样有激情,愿幸福像内存一样经常溢出~! (昨天 01:09)

way: 祝大家新年快乐希望在以后的日子里,男生像Oracle般健壮;女生像android 4.0般漂亮;桃花运像

大家讨论一下如何实现报表展现的?

几乎每个报表系统都是基于ash的,有纯flash的,大家略所欲言,期待看报表达自己的体会和想法。...

AJAVA.ORG

独立 JAVA 社区

第 1 页

目录

前言.....	3
1. Flexpaper 介绍.....	3
1.1 Flexpaper 是什么?	3
1.2 Flexpaper 可以做什么?	4
1.3 Flexpaper 版本.....	4
1.4 Flexpaper 插件.....	5
插件 1.FlexPaper slide	6
插件 2. FlexPaper Annotations	6
插件 3. FlexPaper Watermarking	7
插件 4. FlexPaper AdaptiveUI	8
1.5 SWFTools 介绍	8
2. Flexpaper 二次开发准备工作.....	10
2.1 获得 Flexpaper 源码.....	10
2.2 安装 Adobe Flash Builder	12
2.3 把 Flexpaper 源码导入 Adobe Flash Builder.....	12
2.3.1 创建 Flex 项目.....	12
2.3.2 把 flexpaper 源码复制到 Flex 项目目录.....	16
3. SWFTools 的简单使用	16
3.1 使用 swftools 生成第一个 SWF	17
3.2 SWFTools 参数介绍	18
4. Flexpaper 简单使用.....	21
4.1 第一个 Flexpaper 例子.....	22
4.2 Flexpaper 例子解释.....	24
4.3 Flexpaper 参数说明.....	25
5. Flexpaper 二次开发.....	27
5.1 Flexpaper 重点类包介绍.....	27
5.2 增加(修改)自己的 mxml 应用程序.....	28
5.3 删除 Flexpaper 浏览器的打印按钮.....	33
5.4 删除、修改 Flexpaper 浏览器的 FP 关于我们按钮.....	33
5.5 删除 Flexpaper 浏览器的 LOGO.....	35
5.6 解决中文 SWF 不能读取问题.....	36
5.7 Flexpaper 二次开发的一些常见错误及解决.....	38
5.8. Flexpaper 重新编译、打包.....	39
后记.....	41

前言

由于需要在ajava.org网站的在线文档栏目中增加PDF文档的在线阅读功能,mark这只菜鸟游遍了各个搜索引擎和网站,花了2天时间终于完成一些基本功能,为感谢大家的支持,特发布《Flexpaper二次开发入门教程》,与各位朋友共享,也为ajava.org网站增添一些原创内容。

在本教程中可以学到:

1. 了解Flexpaper、FlashBuilder、SWFTools
2. 学习使用Flexpaper
3. Flexpaper的某些功能的二次开发
4. Flexpaper在二次开发的一些常见错误及解决

本教程的测试环境分别为:

- 1.Windows XP + IE7 + FF8 + 遨游2.5.15
- 2.Windows 2003 + IE7 + FF8 + 遨游2.5.15
- 3.CentOS 5 + FF8

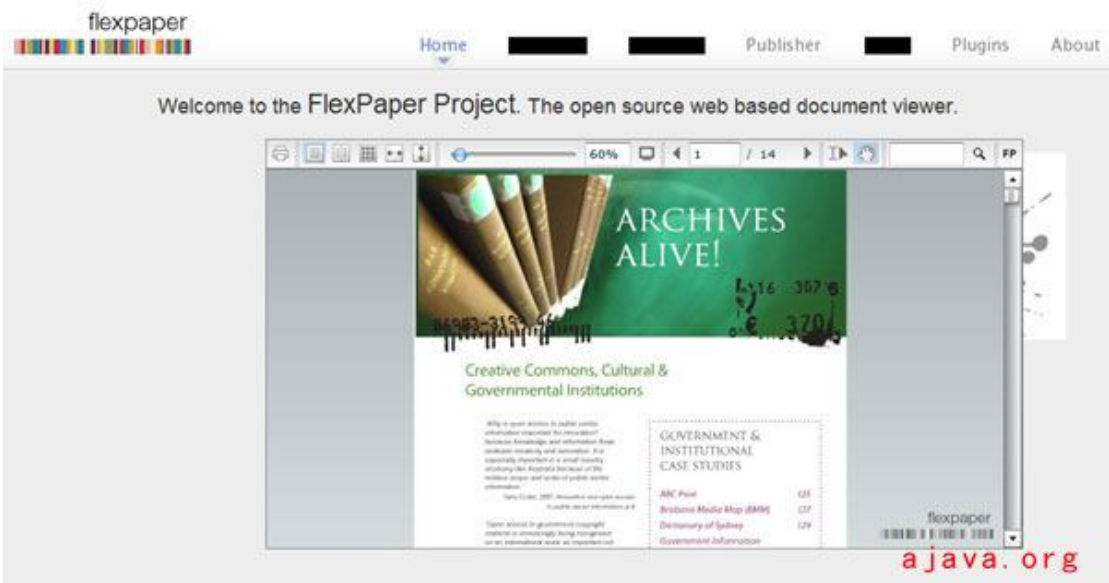
在这里要说明的是,本教程只是入门级的基础教程,只有涉及片面的内容,mark以前未使用过Flex,水平有限,因此本教程中可能会有一些错误,欢迎各位到本站指正。

1. Flexpaper 介绍

1.1 Flexpaper 是什么?

Flexpaper是由一家新西兰的Devaldi公司开发的在线浏览各种文件的组件,它采用的是GPL开源的协议,即源代码开放并可以最其进行修改使用,这一切都是免费得,但不能对Flexpaper源码修改后封装成商业软件进行发布和销售。

Flexpaper官方主页: <http://flexpaper.devaldi.com>



1.2 Flexpaper 可以做什么？

如果你有一些DOC、PPT、PDF等格式的文档，想在网站里提供给大家在线阅读，那怎样做好呢？现在Flexpaper可以为你做到，当然Flexpaper还有一个搭档，那就是SWFTools，通过SWFTools生产SWF，再通过Flexpaper组件显示，这对黄金搭档可以为你实现的在线文档功能。使用Flexpaper浏览PDF，那需要PDF软件环境的支持吗？答案是不需要的，Flexpaper还可以被当做Flex的库来使用，这些都使开发、使用变得更简单。

1.3 Flexpaper 版本

Flexpaper的初始版本1.0发布于2009年，到现在已经有42个版本。swc版本为组件包，使用swc组件包需要导入到工程中再调用，flash版本为可以单独执行的，里面有例子、JS文件、SWF文件，我们采用flash版本很容易就可以使用Flexpaper显示在线文档。下面列出最新的几个版本。

Filename ▼	Summary + Labels ▼	Size ▼	Uploaded ▼
FlexPaper_1.5.0_SDK_3.5_swc.zip	FlexPaper v1.5.0 (r530) Library, built for Adobe Flex 3.5 (SDK build 12683) Featured	841 KB	Dec 28
FlexPaper_1.5.0_flash.zip	FlexPaper v1.5.0 (r530), standalone flash version Featured	2.0 MB	Dec 28
FlexPaper_1.4.7_flash.zip	FlexPaper v1.4.7 (r519), standalone flash version	2.0 MB	Nov 10
FlexPaper_1.4.7_SDK_3.5_swc.zip	FlexPaper v1.4.7 (r519) Library,	840	Nov 10

Filename ▼	Summary + Labels ▼	Size ▼	Uploaded ▼
	built for Adobe Flex 3.5 (SDK build 12683)	KB	
FlexPaper_1.4.5_swk_SDK_3.5_swk.zip	FlexPaper v1.4.5 (r509) Library, built for Adobe Flex 3.5 (SDK build 12683) <small>Deprecated</small>	832 KB	Jun 2011
FlexPaper_1.4.5_flash.zip	FlexPaper v1.4.5 (r509), standalone flash version <small>Deprecated</small>	2.1 MB	Jun 2011
FlexPaper_1.4.2_flash.zip	FlexPaper v1.4.2 (r466), standalone flash version <small>Deprecated</small>	992 KB	Apr 2011
FlexPaper_1.4.2_SDK_3.5_swk.zip	FlexPaper v1.4.2 (r466) Library, built for Adobe Flex 3.5 (SDK build 12683) <small>Deprecated</small>	815 KB	Apr 2011
FlexPaper_1.4.1_flash.zip	FlexPaper v1.4.1 (r450), standalone flash version using swfobject <small>Deprecated</small>	899 KB	Dec 2010
FlexPaper_1.4.1_SDK_4.0_beta_swk.zip	FlexPaper v1.4.1 beta (r450) Library, built for Adobe Flex 4.0	836 KB	Dec 2010
FlexPaper_1.4.1_SDK_3.5_swk.zip	FlexPaper v1.4.1 (r450) Library, built for Adobe Flex 3.5 (SDK build 12683) <small>Deprecated</small>	717 KB	Dec 2010
FlexPaper_1.4.0_flash.zip	FlexPaper v1.4.0 (r401), standalone flash version using swfobject <small>Deprecated</small>	897 KB	Sep 2010
FlexPaper_1.4.0_swk.zip	FlexPaper v1.4.0 (r402) Library, built for Adobe Flex 3.5 (SDK build 12683) <small>Featured</small> <small>Deprecated</small>	717 KB	Sep 2010

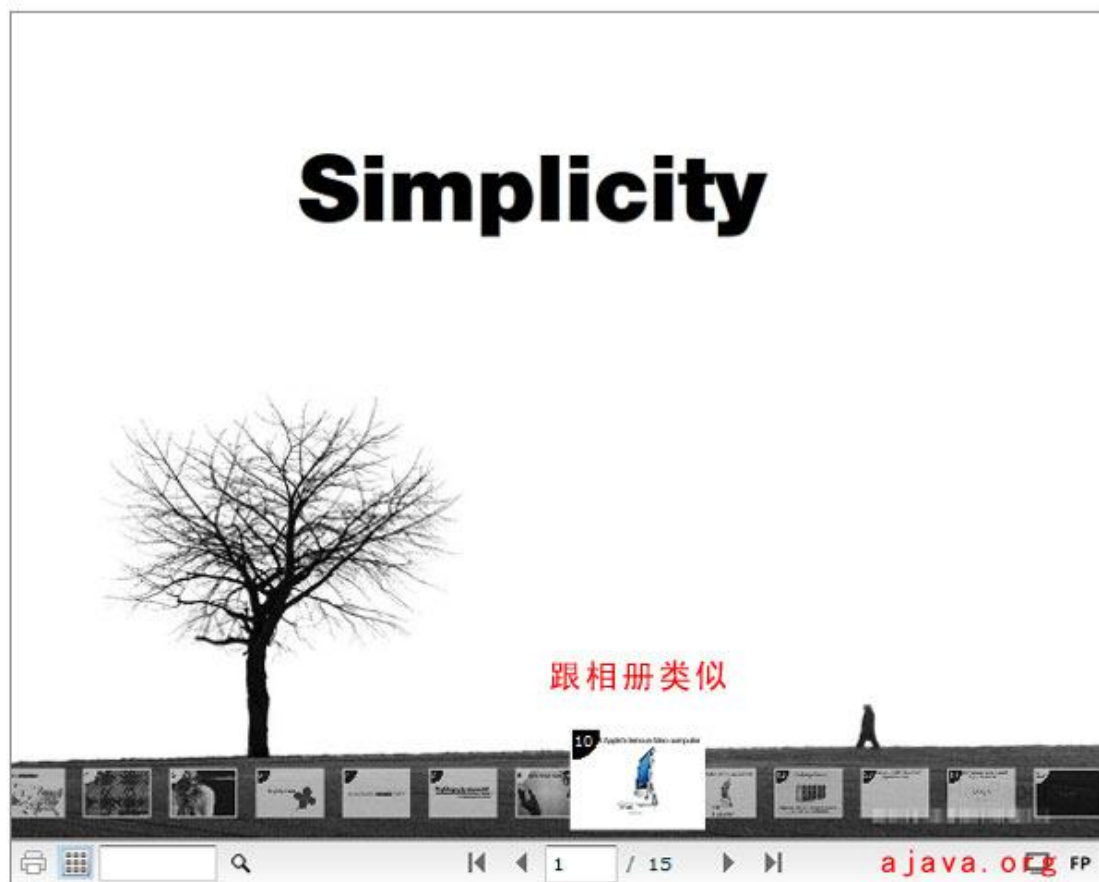
由上面的版本列表中，我们可以看出对应版本采用怎样的FLEX版本编译出来的。

1.4 Flexpaper 插件

Devaldi公司目前提供了4种插件，但这些插件不是免费使用的，要用只有给USD买个Licensing。

插件 1. FlexPaper slide

由下图可以看出，这个插件的功能很像我们经常使用的相册。



插件 2. FlexPaper Annotations

该插件可以通过下面的操作菜单自己画图、加入注解、高亮文字。



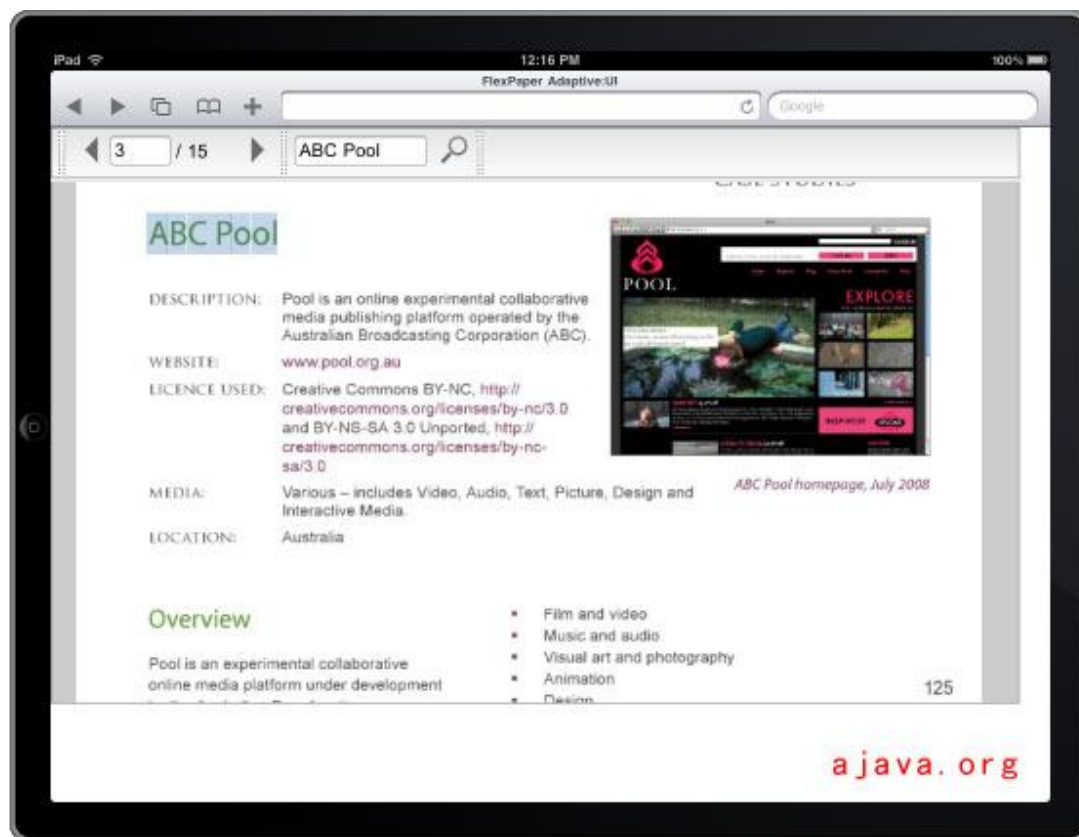
插件 3. FlexPaper Watermarking

使用此插件后可以通过PHP页面来设置FlexPaper的样式



插件 4. FlexPaper AdaptiveUI

通过 AdaptiveUI, 你可以在一些不支持 Adobe Flash 的系统中使用 FlexPaper, 如 Apple iPad。



1.5 SWFTools 介绍

SWFTools是一个SWF处理、转换的开源工具，它的功能主要体现在：

1. SWF的抽取、合并、解析、读写；
2. PDF/JPEG/PNG/AVI/TTF/WAV等格式转换成SWF。

SWFTools是采用GPL开源协议，永久免费，现在SWFTools已经支持Solaris, Linux (32 bit 和 64 bit), FreeBSD, OpenBSD, HP-UX, Solaris, MacOS X and Windows 98/ME/2000/XP/Vista等多个系统。
，下载后安装后可以看到它的工具

python24	文件夹	2012-1-25 20:56
python25	文件夹	2012-1-25 20:56
python26	文件夹	2012-1-25 20:56
swfs	文件夹	2012-1-25 20:56
viewers	文件夹	2012-1-25 20:56
as3compile.exe	797 KB 应用程序	2012-1-25 20:56
font2swf.exe	2,012 KB 应用程序	2012-1-25 20:56
gif2swf.exe	573 KB 应用程序	2012-1-25 20:56
gpdf2swf.exe	17,250 KB 应用程序	2012-1-25 20:56
jpeg2swf.exe	421 KB 应用程序	2012-1-25 20:56
pdf2swf.exe	6,324 KB 应用程序	2012-1-25 20:56
png2swf.exe	387 KB 应用程序	2012-1-25 20:56
rflview.swf	5 KB Shockwave Flash...	2012-1-25 20:56
swfbbox.exe	512 KB 应用程序	2012-1-25 20:56
swfc.exe	3,031 KB 应用程序	2012-1-25 20:56
swfcombine.exe	389 KB 应用程序	2012-1-25 20:56
swfdump.exe	525 KB 应用程序	2012-1-25 20:56
swfextract.exe	706 KB 应用程序	2012-1-25 20:56
swfrender.exe	2,293 KB 应用程序	2012-1-25 20:56
swfstrings.exe	503 KB 应用程序	2012-1-25 20:56
uninstall.exe	78 KB 应用程序	2012-1-25 20:56
uninstall.ini	3 KB 配置设置	2012-1-25 20:56
wav2swf.exe	560 KB 应用程序	2012-1-25 20:56

官方对各个工具的描述:

- **PDF2SWF** A PDF to SWF Converter. Generates one frame per page. Enables you to have fully formatted text, including tables, formulas, graphics etc. inside your Flash Movie. It's based on the [xpdf](#) PDF parser from Derek B. Noonburg.
- **SWFCombine** A multi-function tool for inserting SWFs into Wrapper SWFs, contatenating SWFs, stacking SWFs or for basic parameter manipulation (e.g. changing size).
- **SWFStrings** Scans SWFs for text data.
- **SWFDump** Prints out various informations about SWFs, like contained images/fonts/sounds, disassembly of contained code as well as cross-reference and bounding box data.
- **JPEG2SWF** Takes one or more JPEG pictures and generates a SWF slideshow from them. Supports motion estimation compression (h.263) for better compression of video sequences.
- **PNG2SWF** Like JPEG2SWF, only for PNGs.
- **GIF2SWF** Converts GIFs to SWF. Also able to handle animated gifs.
- **WAV2SWF** Converts WAV audio files to SWFs, using the [L.A.M.E.](#) MP3 encoder library.
- **AVI2SWF** Converts AVI animation files to SWF. It supports Flash MX H.263 compression. Some examples can be found at [examples.html](#). (Notice: this tool is not included anymore in the latest version, as ffmpeg or mencoder do a better job nowadays)
- **Font2SWF** Converts font files (TTF, Type1) to SWF.
- **SWFBBox** Allows to read out, optimize and readjust SWF bounding boxes.
- **SWFC** A tool for creating SWF files from simple script files. Includes support for both ActionScript 2.0 as well as ActionScript 3.0.
- **SWFExtract** Allows to extract Movieclips, Sounds, Images etc. from SWF files.
- **AS3Compile** A standalone ActionScript 3.0 compiler. Mostly compatible with Flex.

gpdf2swf的工作界面:



SWFTools官方网站: <http://www.swftools.org/>

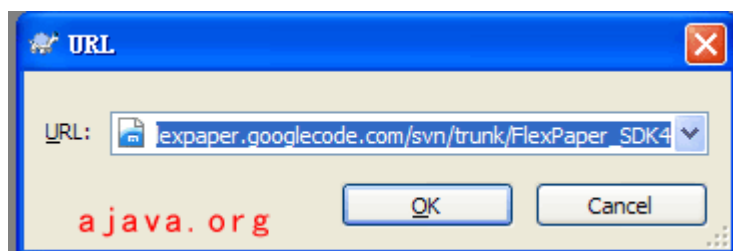


2. Flexpaper 二次开发准备工作

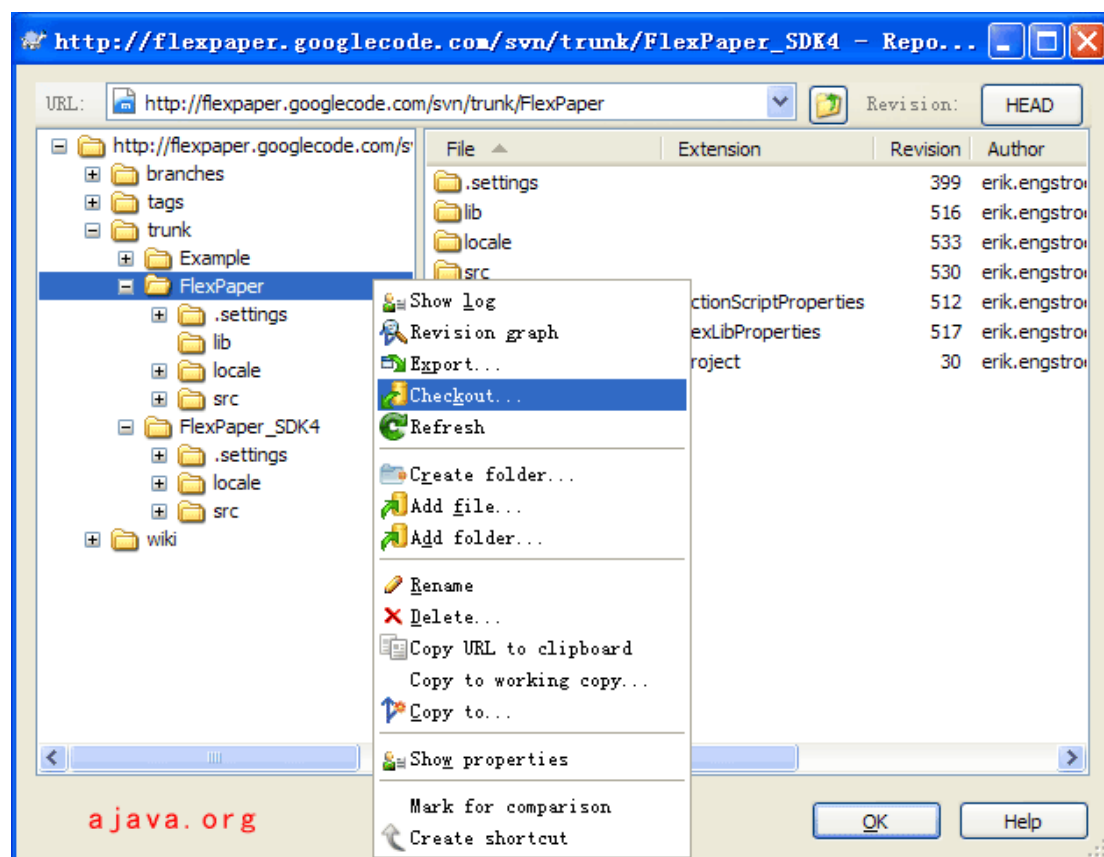
Flexpaper虽然已经提供了不错的功能,但我们还是有可能需要根据自己的需要对Flexpaper进行二次开发,做什么事情都要做足准备,这样才可提高成功率,因此进行Flexpaper二次开发,这一步也是必不可少的,需要什么、怎么去做,我们会在下面慢慢介绍。

2.1 获得 Flexpaper 源码

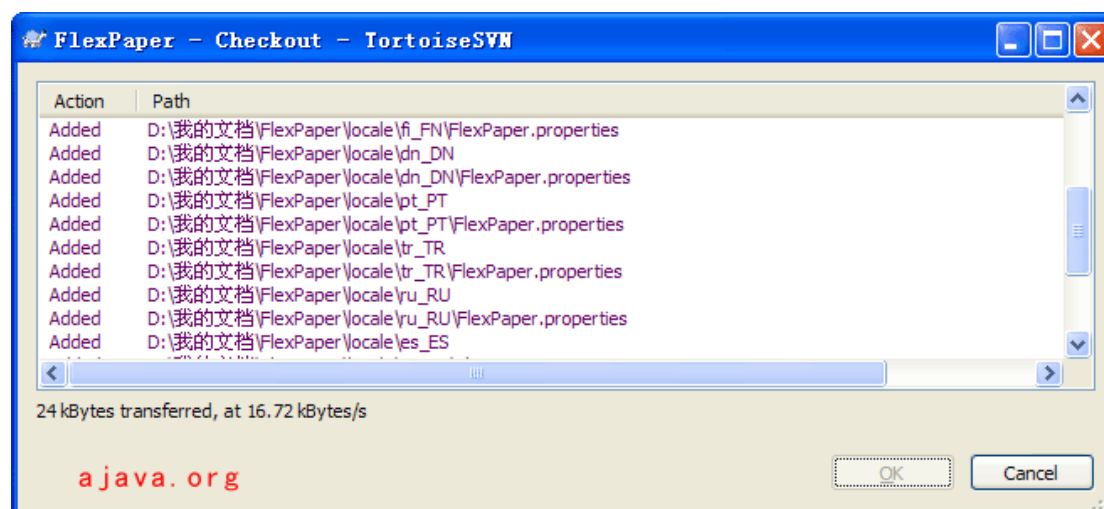
我们可以通过<http://flexpaper.googlecode.com/svn/trunk/>这个地址,获取Flexpaper的源码(我下载的是2012-1-24日的版本)。在获取前,我们需要安装SVN,这里我们使用TortoiseSVN,安装后打开,选择file->TortoiseSVN->Repo-browser,输入上面的地址:



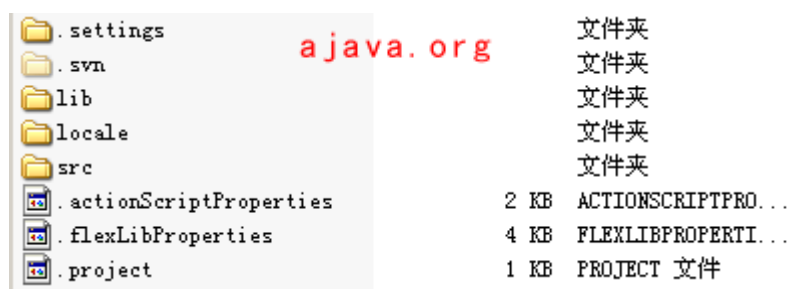
链接上后，可以看到目录结果如下图：



我们选择trunk/FlexPaper目录，右键选择Checkout，在提示框中选择Checkout目录，确认后会开始下载，如下图：



Checkout完毕后，可以看到相应目录下



2.2 安装 Adobe Flash Builder

我们可以到adobe的官方网站下载Adobe Flash Builder，目前最新版本是4.6，由于mark的电脑性能不好，所以采用了4.0版本。

Flash Builder 4.6的下载地址：

https://www.adobe.com/cfusion/tdrc/index.cfm?product=flash_builder

Flash Builder官方主页：

<http://www.adobe.com/products/flash-builder.html>

下载安装后，会提示你注册，由于mark no money，所以不会购买，大家可以在网上找找某些注册码跟注册步骤，看能否处理。



进入到Adobe Flash Builder工作界面后，我们就可以进入下一步。

2.3 把 Flexpaper 源码导入 Adobe Flash Builder

2.3.1 创建 Flex 项目

选择工具栏上的“文件”->新建 -> Flex项目

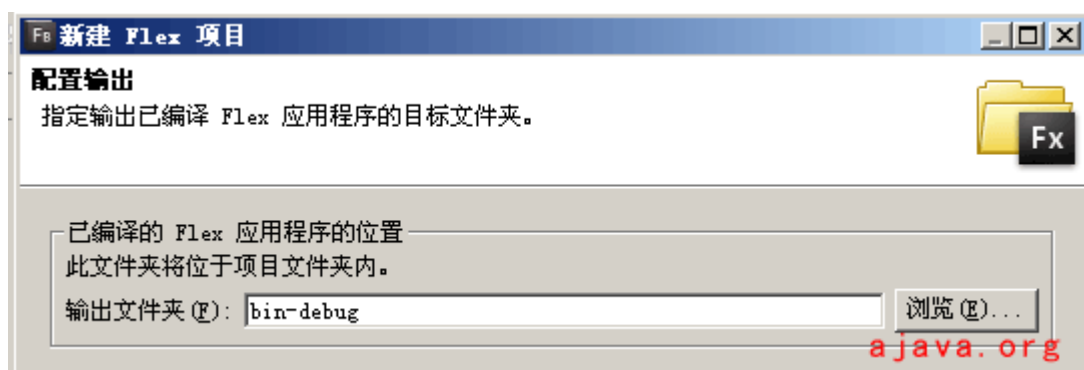


在创建项目窗口，我们主要填写、选择下面内容：

- 输入项目名称，这里我使用了ajavaflexpaper；
- 项目位置为你的存放项目文件的目录，这里我使用了F:\ajavaflexpaper；
- 应用类型我们选择”web(在Adobe Flash Player中运行)”，因为我们要在web页面浏览，所以选择了此项；
- 选择Flex SDK版本，这里采用Flex SDK 4.0；
- 应用的是那种服务器技术，可选择：无/其他、ASP.NET、ColdeFusion、J2EE、PHP等，这里我选择了无/其他。

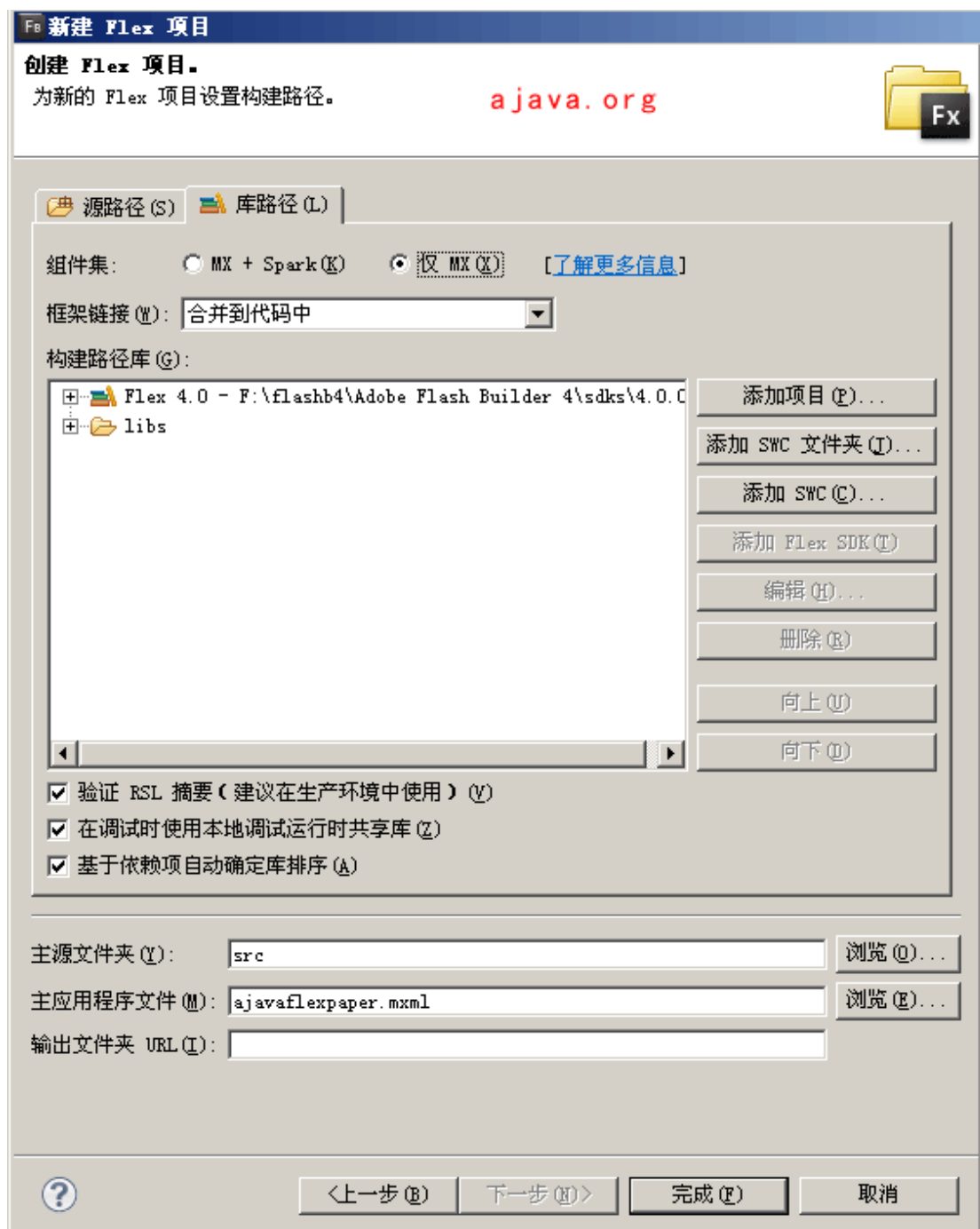


填写、选择完毕后，点击下一步，进入“配置输出”提示框，选择已编译的FLEX应用程序的位置，这里我采用了默认的bin-debug目录，源码重新编译后，文件会放在bin-debug目录。

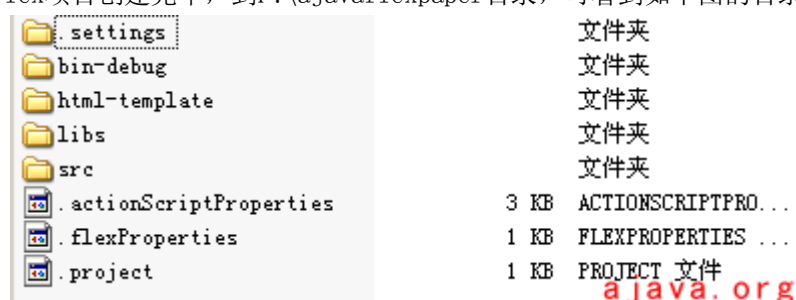


点击下一步，来到最后一步，设置源路径和库路径。

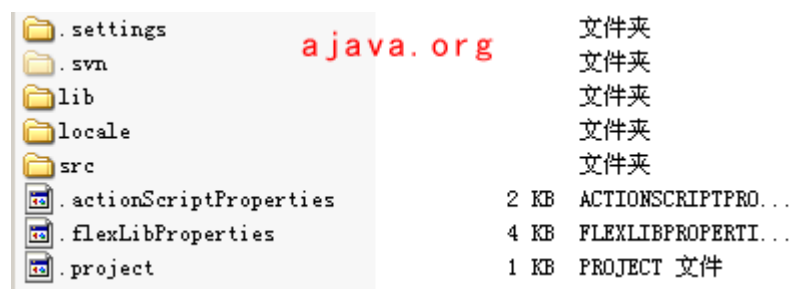
我们选择的内容有组件集选择“仅MX”，框架链接选择“合并到代码中”，其他默认就可以，当然你也可以重新设置你的“主源文件夹”和“主应用程序文件”。



点击完成，Flex项目创建完毕，到F:\ajavaflexpaper目录，可看到如下图的目录结构：

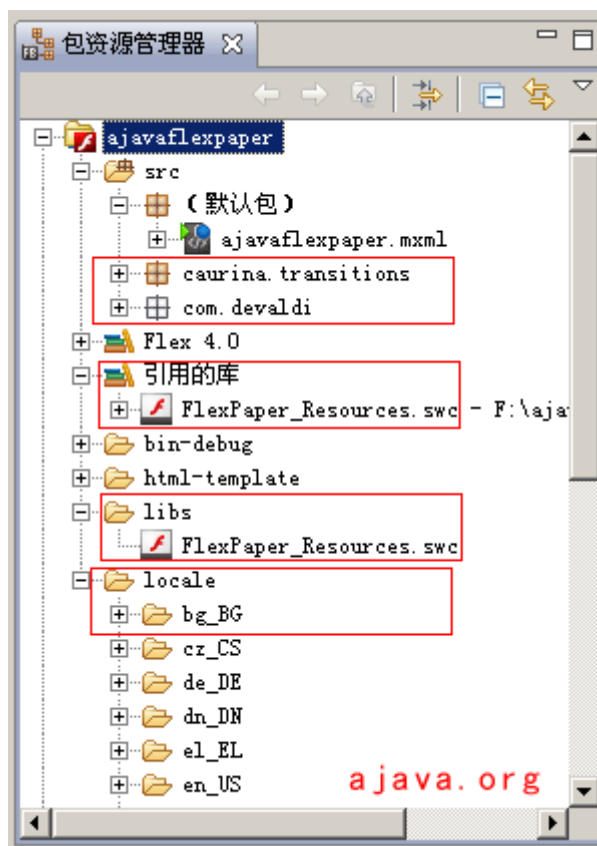


2.3.2 把 flexpaper 源码复制到 Flex 项目目录



我们要复制的目录有：locale、src复制到Flex项目根目录(F:\ajavaflexpaper)中，lib里的FlexPaper_Resources.swc复制到F:\ajavaflexpaper\libs中。

复制完毕后，在Flash Builder 4中右击项目名选择刷新，刷新后可见项目中增加了一些文件，见下图红色框部分：



到目前为止，准备工作已完成，为FlexPaper二次开发做好准备。下节我们先介绍FlexPaper和SWFTTools的简单使用，对使用有了了解后，再进入二次开发内容。

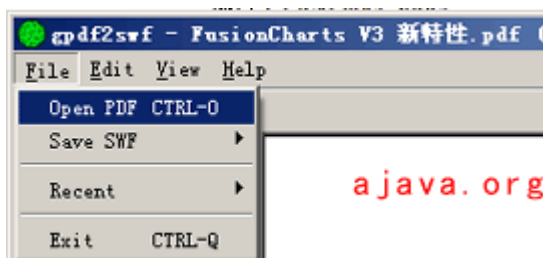
3. SWFTTools 的简单使用

有了第一章的介绍，大家对SWFTTools应该有了基本的认识，下面我们开始演示怎样使用SWFTTools的gpdf2swf.exe、pdf2swf.exe的使用，由于本教程的内容范围较小，所以其他工具使用不在这里介绍。

3.1 使用 swftools 生成第一个 SWF

方式一、使用 gpdf2swf 可视化窗口

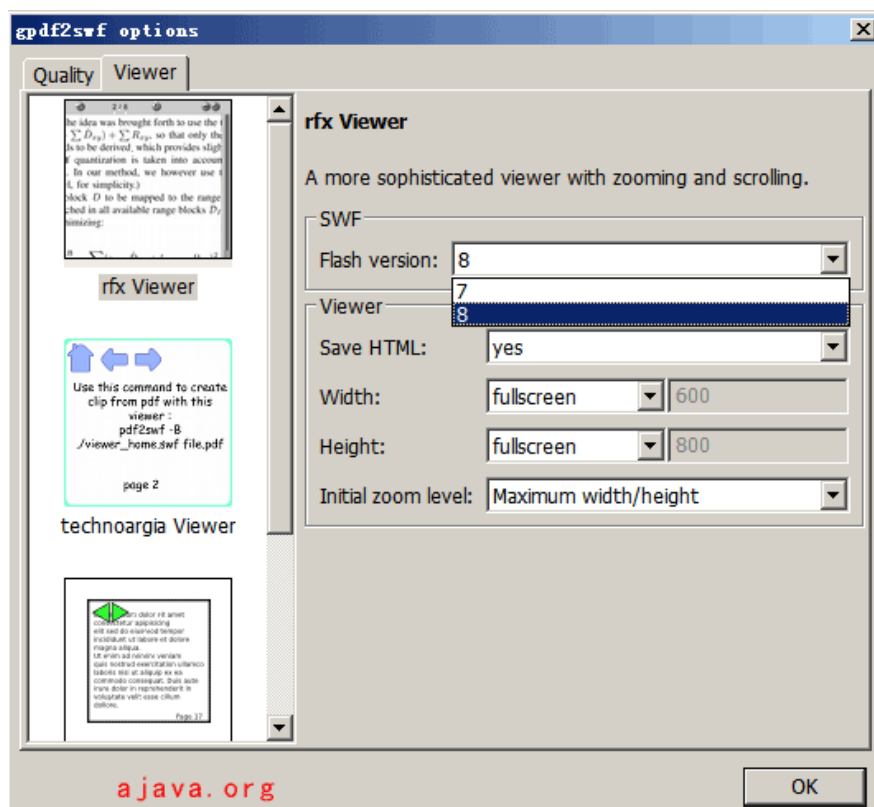
我们打开 gpdf2swf 可视化窗口后，选择 Open PDF，选择要生成 SWF 的 PDF 文件，之后保存为 SWF，你可以选择把所有页面或者部分页面加入到 SWF 文件，操作如下图：



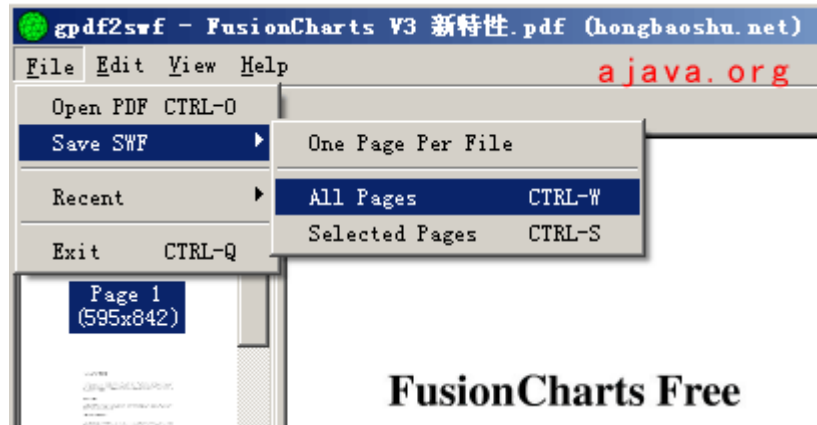
(打开PDF文件)



(选择PDF文件)



(设置参数)



(保存为SWF文件)

在这里大家要注意，使用目前最新版本的SWFTools中gpdf2swf.exe生成PDF，在设置参数中，只能选择Flash 7或者8版本，但目前Flash 9或者10版本都大行其道，因此mark建议生成SWF时采用第二种方式，我们下面继续介绍。

方式二、使用pdf2swf.exe命令

我们打开CMD，进入SWFTools的安装目录，这里我的安装目录是D:\Program Files\SWFTools，pdf2swf.exe的命令形式为：

pdf2swf.exe [-options] file.pdf -o file.swf

我们执行下面的命令：

```
pdf2swf.exe -t F:\ajava\FusionCharts.pdf -s flashversion=9 -o  
F:\ajava\FusionCharts.swf
```

```
D:\Program Files\SWFTools>pdf2swf.exe -t F:\ajava\FusionCharts.pdf -s flashversion=9 -o F:\ajava\FusionCharts.swf  
NOTICE processing PDF page 1 <595x842:0:0>  
NOTICE File contains jpeg pictures  
NOTICE processing PDF page 2 <595x842:0:0>  
NOTICE processing PDF page 3 <595x842:0:0>  
NOTICE processing PDF page 4 <595x842:0:0>  
NOTICE processing PDF page 5 <595x842:0:0>  
NOTICE processing PDF page 6 <595x842:0:0>  
NOTICE processing PDF page 7 <595x842:0:0>  
NOTICE processing PDF page 8 <595x842:0:0>  
NOTICE Writing SWF file F:\ajava\FusionCharts.swf
```

由上图我们可以看到每页转换的提示，最后写入到SWF文件中，打开F:\ajava目录，可以看到FusionCharts.swf已经生成，这个swf的版本是flash 9。

这里用到了pdf2swf.exe的-t、-s、-o参数，下一节我们对pdf2swf.exe的参数作详细介绍。

注意：pdf2swf.exe命令中的文件名、目录名，不能带空格，否则不能生成SWF。

3.2 SWFTools 参数介绍

我们输入：D:\Program Files\SWFTools>pdf2swf.exe -h，就可以获得帮助信息，本小节的参数描述转自某朋友的文章，mark作了简单的修改，因不知道谁是原创者，特说明，表示感谢，若作

者见本教程，可到ajava.org网站联系mark，将在本节中添加原作者大名。

```
D:\Program Files\SWFTools>pdf2swf.exe -h
Usage: pdf2swf.exe [-options] file.pdf -o file.swf

-h , --help          Print short help message and exit
-V , --version       Print version info and exit
-o , --output file.swf Direct output to file.swf. If file.swf contains '
%' <file%.swf>, then each page goes to a seperate file.
-p , --pages range   Convert only pages in range with range e.g. 1-20
or 1,4,6,9-11 or
```

-h , - help	打印帮助信息
-V , - version	打印版本号
-o , - output file.swf	指定输出的swf文件名
-p , -pages range	指定转换的页面范围1-20 或者1, 4, 6, 9-11，使用的页码描述方法与打印机打印文件时候的选页一样
-P , - password password	指定打开pdf的密码
-v , - verbose	转换时输出详细的内容
-z , - zlib	使用Flash 6的zlib压缩机制
-i , - ignore	允许程序修改pdf的绘制顺序，可能会导致结果与原来有差异
-j , - jpegquality quality	设置转换其中的jpeg图片的质量，从0到100，默认值是85。
-s , - set param=value	设置SWF转码时候的参数，具体参数可以用pdf2swf -s help获取
-w , - samewindow	设置转换后的swf打开原pdf中的连接时使用相同的窗口
-t , - stop	在每页结尾添加一个stop()命令
-T , - flashversion num	设置SWF所使用的flash版本号
-F , - fontdir directory	指定字体文件所在路径
-b , - defaultviewer	指定默认的swf导航文件，用来翻页、放大缩小等等
-l , - defaultloader	指定默认的swf加载文件，用来显示加载进程效果
-B , - viewer filename	指定swf导航文件，作用同-b
-L , - preloader filename	指定swf加载文件，作用同-l
-q , - quiet	不打印普通信息，用-qq就不打印警告信息。
-S , - shapes	不使用字体，所有都转为形状。
-f , - fonts	在swf中保存全部字体。
-G , - flatten	在文件中尽量去除影片层，合并它们
-I , - info	不做实际转换，仅显示PDF的信息。
-Q , - maxtime n	如果运行时间超时则退出。

我们可以看到-s参数的用途，设置SWF转码时候的参数，可以通过pdf2swf -s help命令获得更多的参数：

```

D:\Program Files\SWFTools>pdf2swf.exe -s help
PDF Parameters:

PDF device global parameters:
fontdir=<dir>      a directory with additional fonts
font=<filename>    an additional font filename
pages=<range>      the range of pages to convert (example: pages=1-100,210-)
zoom=<dpi>         the resolution (default: 72)
languagedir=<dir>  Add an xpdf language directory
multiply=<times>   Render everything at <times> the resolution
poly2bitmap       Convert graphics to bitmaps
bitmap            Convert everything to bitmaps
SWF Parameters:

```

PDF参数:

PDF device global parameters:

fontdir= 指定字体目录, 与1级参数的-F相若

font= 增加额外的字体文件

pages= 指定页面范围, 与1级参数的-p相若

zoom= 指定分辨率, 默认为72dpi

languagedir= 增加一个xpdf的语言目录, 对非西欧字符有用

multiply= 在几倍分辨率下渲染

poly2bitmap 把其中的图形转成点阵

bitmap 把所有内容转成点阵(包括字体)

SWF参数:

SWF layer options:

jpegsubpixels=<pixels> jpeg图片的分辨率

ppmsubpixels=<pixels> 无损图片的分辨率

subpixels=<pixels> 快速设置上两个参数

drawonlyshapes 所有都转成图形

ignoredraworder SWFs 允许执行一些小优化

linksopennewwindow 链接打开新窗口

linktarget 新链接窗口的名称

linkcolor=<color> 链接的颜色

linknameurl 链接名称与链接URL一致

storeallcharacters 保存所有的字符字体

enablezlib 使用zlib压缩

bboxvars 在as中保存swf的区域大小

dots 保存单点显示

reordertags=0/1 执行某些tag优化

internallinkfunction=<name> 内部链接函数, 如果点击一个内部链接, 将调用该actionscript函数

externallinkfunction=<name> 外部链接函数, 如果点击一个外部链接, 将调用该actionscript函数

disable_polygon_conversion 不要将笔画转成多边形

caplinewidth=<width> 线条最低转换宽度, 比这个细的线条将不转换

insertstop 在swf的每个帧中添加stop()函数

protect 增加protect标签, 禁止在flash中加载该swf

flashversion=<version> 设置最低swf版本

framerate=<fps>	设置帧率
minlinewidth=<width>	将宽度少于某值的矩形转成线条
simpleviewer	使用简单的导航
animate	insert a showframe tag after each placeobject (animate draw order of PDF files)
jpegquality=<quality>	设置jpeg的压缩质量
splinequality=<value>	设置样条曲线的转换质量
disablelinks	禁止链接

4. Flexpaper 简单使用

通过上面三章的内容，大家对Flexpaper、SWFTools应该有大概的了解了，SWF文件也已经生成了，我们开始进入Flexpaper的使用的介绍。

本章中只演示Flexpaper在WEB页面上的使用，我们要下载一个Flash版本的FlexPaper，这里我选择了FlexPaper_1.5.0_flash。

下载地址：<http://code.google.com/p/flexpaper/downloads/list>

下载、解压后，可以看到包含如下文件：

名称	大小	类型
examples		文件夹
js		文件夹
php		文件夹
Changelog.txt	9 KB	文本文档
FlexPaperViewer.swf	402 KB	Shockwave Flash...
highlight.xml	1 KB	XML 文档
index.html	5 KB	Maxthon Document
LICENSE.txt	36 KB	文本文档
LICENSE-COMMERCIAL.txt	8 KB	文本文档
LICENSE-UNLIMITED.txt	10 KB	文本文档
Paper.swf	527 KB	Shockwave Flash...
playerProductInstall.swf	1 KB	Shockwave Flash...

主要文件、文件夹说明：

Examples 存放Flexpaper使用例子

Js 存放Flexpaper调用的JS文件

Php 存放PHP使用的文档、JS、库、SWF文件

Index.html 例子主页

FlexpaperViewer Flexpaper的核心文件，用于浏览PDF

Paper.swf 官方的默认宣传文件

playerProductInstall.swf 如果客户端浏览器的flashplayer版本过低，他是不会嵌入你的swf,而是嵌入这个playerProductInstall.swf到页面上下载flashplayer的安装文件。

4.1 第一个 Flexpaper 例子

那我们开始完成第一个Flexpaper例子吧，一个最简单的例子，我们主要用到FlexpaperViewer.swf、JS文件夹中的flexpaper_flash.js。

创建HTML页面test.html，代码如下：

```
<HTML>
<HEAD>
  <TITLE>ajava.org Flexpaper例子</TITLE>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
</HEAD>

<BODY>
<script type="text/javascript" src="js/flexpaper_flash.js"></script>

<div style="position:absolute;left:10px;top:10px;">
  <p id="viewerPlaceholder"
style="width:660px;height:553px;display:block">Document loading..</p>

  <script type="text/javascript">
    var fp = new FlexPaperViewer(
      'FlexPaperViewer',
      'viewerPlaceholder', { config : {
        SwfFile : escape('FusionCharts.swf'),
        Scale : 0.6,
        ZoomTransition : 'easeOut',
        ZoomTime : 0.5,
        ZoomInterval : 0.2,
        FitPageOnLoad : true,
        FitWidthOnLoad : false,
        FullScreenAsMaxWindow : false,
        ProgressiveLoading : false,
        MinZoomSize : 0.2,
        MaxZoomSize : 5,
        SearchMatchAll : false,
        InitViewMode : 'Portrait',
        PrintPaperAsBitmap : false,
        ViewModeToolsVisible : true,
        ZoomToolsVisible : true,
        NavToolsVisible : true,
        CursorToolsVisible : true,
        SearchToolsVisible : true,
        localeChain: 'en_US'
      }
    }
  </script>
</div>
```



```

</script>
</div>
</BODY>
</HTML>

```

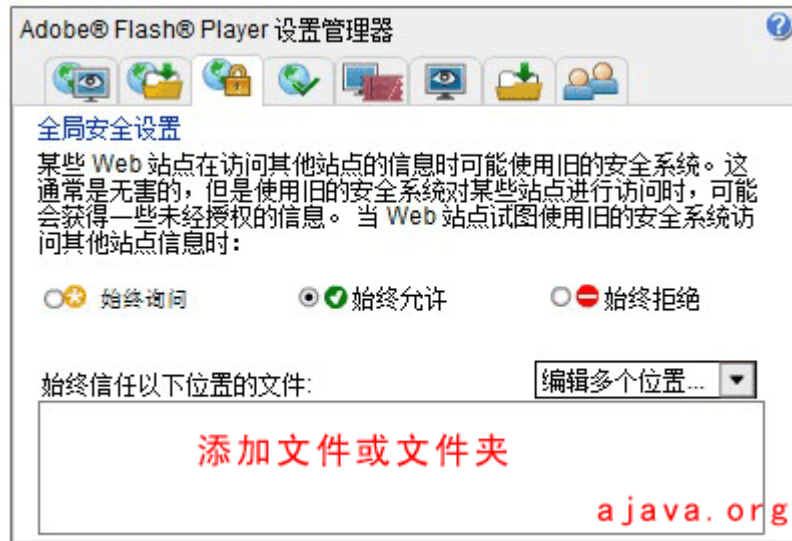
把第三章部分生产的FusionCharts.swf复制到FlexPaper根目录下，我这里是F:\FlexPaper1.5.0\flash，这一步是必须的，如果不是test.html怎样读得到FusionCharts.swf呢？

测试:

打开浏览器，输入test.html的本地地址，就可以看到FlexPaper的浏览效果。这里我分别在前言中的3种测试环境，结果都没问题，效果如下图：



如果你用FF浏览器，出现提示“SecurityError: Error #2148: SWF 文件 file:///F:/FlexPaper1.5.0/flash/FlexPaperViewer.swf 不能访问本地资源FusionCharts.swf。只有仅限于文件系统的 SWF 文件和可信的本地 SWF 文件可以访问本地资源。”，或者IE一直都处于加载状态，这种情况，是由于Flexpaper还没获得Adobe Flash的信任，这时你可以请访问http://www.macromedia.com/support/documentation/en/flashplayer/help/settings_manage_r04a.html#119065，选择始终允许，把FlexPaper1.5.0/flash文件夹添加到信任列表中。



4.2 Flexpaper 例子解释

在上述的例子中的几点解释：

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

ajava.org 网站服务器、数据库、文件、页面编码都是 UTF-8，所以才用 UTF-8 编码？其实不是，Flex 就是基于 unicode 编码的，建议大家采用 UTF-8；

```
<script type="text/javascript" src=" js/flexpaper_flash.js "></script>
```

Test.html 加载是引入 js/flexpaper_flash.js，flexpaper_flash.js 是处理 SWF 嵌入的脚本，大家可以在 346 行增加一个 alert，显示 html 的值，看了就比较清楚。

```
<div style="position:absolute;left:10px;top:10px;">
```

设置 div 样式后，FlexPaperViewer 的位置会有变化，如 position:absolute 改成相对定位，top 改成 200px，大家就可以看到效果的不同。

```
<p id="viewerPlaceHolder" style="width:660px;height:553px;display:block">Document  
loading.</p>
```

此段代码为加载提示，如果没有调用到 FlexPaperViewer，会一直显示 Document loading

```
var fp = new FlexPaperViewer('FlexPaperViewer',  
'viewerPlaceHolder', { config : {.....}})
```

此段代码是创建一个 FlexPaperViewer，它在 flexpaper_flash.js 中有定义，共三个参数

```
config : { SwfFile : escape('FusionCharts.swf'),.....}
```

此段代码里包含对 FlexPaper 参数的设置，详细请看下一节。

4.3 Flexpaper 参数说明

SwfFile (String) 需要使用Flexpaper打开的文档

Scale (Number) 初始化缩放比例，参数值应该是大于零的整数

ZoomTransition (String) Flexpaper中缩放样式，它使用和Tweeners一样的样式，默认参数值为easeOut. 其他可选值包括: easenone, easeout, linear, easeoutquad

ZoomTime (Number) 从一个缩放比例变为另外一个缩放比例需要花费的时间，该参数值应该为0或更大。

ZoomInterval (Number) 缩放比例之间间隔，默认值为0.1，该值为正数。

FitPageOnLoad (Boolean) 初始化得时候自适应页面，与使用工具栏上的适应页面按钮同样的效果。

FitWidthOnLoad (Boolean) 初始化的时候自适应页面宽度，与工具栏上的适应宽度按钮同样的效果。

localeChain (String) 设置地区（语言），目前支持以下语言。

- en_US (English)
- fr_FR (French)
- zh_CN (Chinese, Simple)
- es_ES (Spanish)
- pt_BR (Brazilian Portugese)
- ru_RU (Russian)
- fi_FN (Finnish)
- de_DE (German)
- nl_NL (Netherlands)
- tr_TR (Turkish)
- se_SE (Swedish)
- pt_PT (Portugese)
- el_EL (Greek)
- da_DN (Danish)
- cz_CS (Czech)
- it_IT (Italian)
- pl_PL (Polish)
- pv_FN (Finnish)
- hu_HU (Hungarian)

FullScreenAsMaxWindow (Boolean) 当设置为true的时候，单击全屏按钮会打开一个flexpaper最大化的新窗口而不是全屏，当由于flash播放器因为安全而禁止全屏，而使用flexpaper作为独立的flash播放器的时候设置为true是个优先选择。

ProgressiveLoading (Boolean) 当设置为true的时候，展示文档时不会加载完整个文档，而是逐步加载，但是需要将文档转化为9以上的flash版本（使用pdf2swf的时候使用-T 9 标签）。

MaxZoomSize (Number) 设置最大的缩放比例。

MinZoomSize (Number) 最小的缩放比例。

SearchMatchAll (Boolean) 设置为true的时候，单击搜索所有符合条件的地方高亮显示。

InitViewMode (String) 设置启动模式如"Portrait" or "TwoPage".

ViewModeToolsVisible (Boolean) 工具栏上是否显示样式选择框。

ZoomToolsVisible (Boolean) 工具栏上是否显示缩放工具。

NavToolsVisible (Boolean) 工具栏上是否显示导航工具。

CursorToolsVisible (Boolean) 工具栏上是否显示光标工具。

SearchToolsVisible (Boolean) 工具栏上是否显示搜索。

对比本例子中的参数设置：

SwfFile : escape(' FusionCharts.swf'), //加载的SWF文件为FusionCharts.swf

Scale : 0.6, //初始化缩放比例为0.6, 即60%

ZoomTransition : 'easeOut',

ZoomTime : 0.5,

ZoomInterval : 0.2,

FitPageOnLoad : true, //加载时自动适应页面

FitWidthOnLoad : false, //加载时自动适应页面宽度

FullScreenAsMaxWindow : false,

ProgressiveLoading : false,

MinZoomSize : 0.2, //页面最小可缩小成20%

MaxZoomSize : 5, //页面最大可放大成500%

SearchMatchAll : false,

InitViewMode : 'Portrait',

PrintPaperAsBitmap : false,

ViewModeToolsVisible : true,

ZoomToolsVisible : true,

NavToolsVisible : true,

CursorToolsVisible : true,

SearchToolsVisible : true,

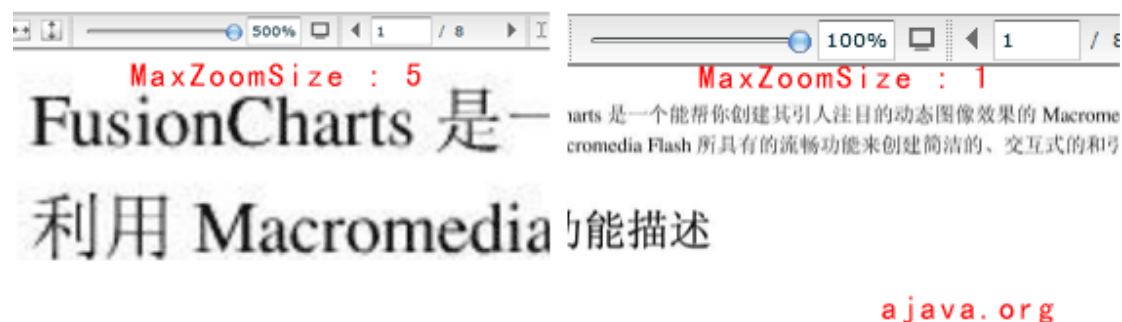
localeChain: 'en_US' //设置地区（语言）为en_US

上面带注解的参数为我们经常设置的参数，下面我们来简单演示一下几个参数改变的效果：

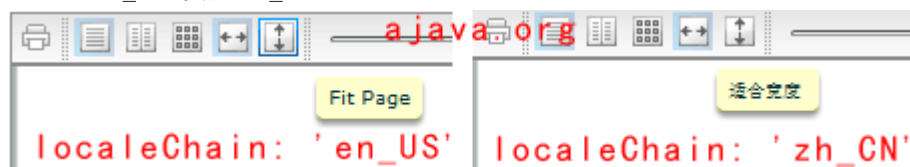
FitWidthOnLoad : false 改成true



MaxZoomSize : 5 改成1



localeChain: 'en_US' 改成 'zh_CN'



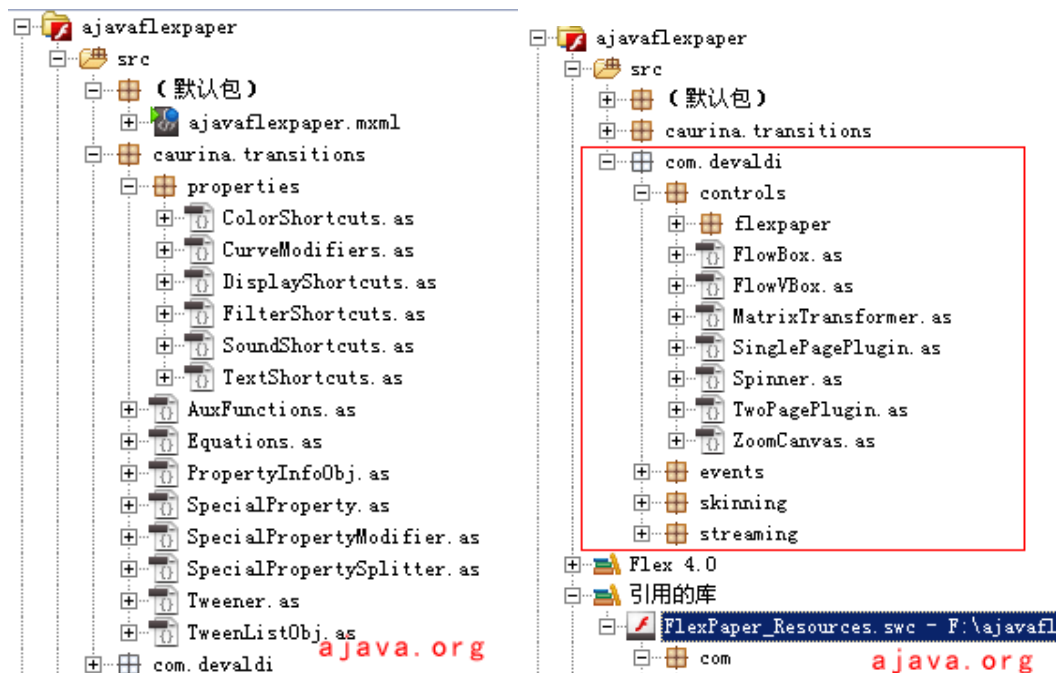
通过本章的介绍,大家应该可以简单的使用Flexpaper了,下一章我们将介绍Flexpaper二次开发。

5. Flexpaper 二次开发

大家通过前几章内容已经基本可以简单的使用Flexpaper了,但某些时候,我们会觉得FlexPaperViewer的某些内容不适合显示出来,如Flexpaper的打印、官方介绍、logo等,这时候我们就需要对Flexpaper进行二次开发,修改Flexpaper的源码来达到效果。

5.1 Flexpaper 重点类包介绍

通过第二章中介绍的二次开发准备工作,我们可以看到Flexpaper的源码了,



1) caurina.transitions为actionscript类包

2) com.devaldi为devaldi公司开发的Flexpaper核心类包，如上图红色框部分。

其中：controls为主控部分，events为事件处理部分，skinning为UI设置部分，streaming为文件流处理部分。

3) FlexPaper_Resources.swc为com.devaldi调用的资源库，我们可以看到一些abc(ActionScript bytecode(字节码))文件，编译过看不到源代码，ABC文件不可以单独运行，必须放入SWF(.swf)文件中才可以被Flash Player执行。

我们下面的例子中只涉及到com.devaldi.controls包文件的修改，其他文件的修改未涉及，各位高手可以研究一下源码，看可否写出自己的插件，呵呵。Mark连Flex菜鸟也不算，还没入门，所以上述功能只是猜测，大家眼睛是雪亮的，欢迎指正。

5.2 增加(修改)自己的 mxml 应用程序

做完二次开发准备工作后，大家可以测试：

- 重新编译后会发现bin-debug有ajavaflexpaper.swf，改名成FlexPaperViewer.swf后使用，刷新页面后发现空白页面，无任何效果。
- 把原有的FlexPaperViewer.mxml右键设置为默认应用程序，重新编译后，会发现空白页面，无任何效果。

针对a点，我们先看看ajavaflexpaper.mxml

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:fx="http://ns.adobe.com/mxml/2009"
                 xmlns:mx="library://ns.adobe.com/flex/mx" layout="absolute"
minWidth="955" minHeight="600">
    <fx:Declarations>
```

```

        <!-- 将非可视元素（例如服务、值对象）放在此处 -->
    </fx:Declarations>
</mx:Application>

```

里面没有作任何处理，所以显示空白页面是很正常的。

针对b点，我们看看FlexPaperViewer.mxml

```

<fp:FlexPaperViewer_Base xmlns:mx="http://www.adobe.com/2006/mxml"
    xmlns:fp="com.devaldi.controls.flexpaper.*"
    xmlns:ct="com.devaldi.controls.*"
    xmlns:st="com.devaldi.skinning.*"

    addedToStage="bindStageEvents(event)"
>

```

我们查找bindStageEvents(e:Event)方法，发现没有获取web页面传递的FlexPaper参数处理方法，FlexPaperViewer_base.mxml里有getUrlParamaters()方法，但发现fullUrl是web页面页面URL地址，不包含FlexPaper参数，具体用法没研究。

通过几位朋友的文章介绍，mark修改ajavaflexpaper.mxml如下

```

<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
    xmlns:fp="com.devaldi.controls.flexpaper.*"
    layout="absolute" width="100%" height="100%"
    applicationComplete="initApp();">

    <mx:Script>
        <![CDATA[
            import mx.controls.Alert;

            public var _aid = 0; //文档ID

            [Bindable]
            public var _Scale:Number = 1; //缩放比例

            [Bindable]
            public var _EncodeURI:Boolean = true;

            [Bindable]
            public var _SwfFile:String = ""; //SWF文件路径

            [Bindable]
            public var _ZoomTransition:String = "easeOut";

            [Bindable]
            public var _ZoomTime:Number = 0.6;

            [Bindable]

```



```
public var _ZoomInterval:Number = 0.1;

[Bindable]
public var _MinZoomSize:Number = 0.3;

[Bindable]
public var _MaxZoomSize:Number = 5;

[Bindable]
public var _FitPageOnLoad:Boolean = false; //加载后适合高度

[Bindable]
public var _FitWidthOnLoad:Boolean = false; //加载后适合宽度

[Bindable]
public var _PrintEnabled:Boolean = true; //是否支持打印

[Bindable]
public var _FullScreenAsMaxWindow:Boolean = false; //是否支付全屏

[Bindable]
public var _ProgressiveLoading:Boolean = false; //是否延迟加载

[Bindable]
public var _localeChain:String = "zh_CN"; //语言

private var isFocus:Boolean = false;

//初始化参数
private function initApp():void{
    var params:Object = Application.application.parameters;
    _Scale = getNumber(params, "Scale", 1);
    _SwfFile = getString(params, "SwfFile", "Paper.swf");
    _EncodeURI = getNumber(params, "EncodeURI", 1);
    _ZoomTransition = getString(params, "ZoomTransition", "easeOut");
    _ZoomTime = getNumber(params, "ZoomTime", 0.6);
    _MinZoomSize = getNumber(params, "MinZoomSize", 0.2);
    _MaxZoomSize = getNumber(params, "MaxZoomSize", 5);
    _ZoomInterval = getNumber(params, "ZoomInterval", 0.1);
    _FitPageOnLoad = getBoolean(params, "FitPageOnLoad", false);
    _FitWidthOnLoad = getBoolean(params, "FitWidthOnLoad", false);
    _PrintEnabled = getBoolean(params, "PrintEnabled", true);
    _FullScreenAsMaxWindow = getBoolean(params, "FullScreenAsMaxWindow",
false);

    _ProgressiveLoading = getBoolean(params, "ProgressiveLoading",
```

```
true);

    _localeChain = params["localeChain"];
    //注册事件监听
    this.addEventListener(MouseEvent.CLICK, onClick);
    this.addEventListener(MouseEvent.CLICK, onDoubleClick);

    //开放给外部 (javascript) 调用
    ExternalInterface.addCallback("hasFocus", hasFocus);
    //ExternalInterface.addCallback("focus", focus);
    ExternalInterface.addCallback("setViewerFocus", setViewerFocus);

}

private function onClick(event:MouseEvent):void{
    this.isFocus = true;
}

private function onDoubleClick(event:MouseEvent):void{
    this.isFocus = false;
}

public function hasFocus():Boolean{
    return isFocus;
}

public function setViewerFocus(isFocus:Boolean):void{
    this.FlexPaper.setViewerFocus();
}

/**
 *
 * 获取String类型参数
 * 如果没有，则返回默认值
 */
private function getString(params:Object, name:String,
def:String):String{
    if(params[name] != null){
        return params[name];
    }
    return def;
}

private function getNumber(params:Object, name:String,
def:Number):Number{
    if(params[name] != null){
```

```

        return params[name];
    }
    return def;
}

private function getBoolean(params:Object, name:String,
def:Boolean):Boolean{
    if(params[name] != null){
        return params[name] == "true";
    }
    return def;
}
]]>
</mx:Script>
<fp:FlexPaperViewer id="FlexPaper"
    width="100%"
    height="100%"
    Scale="{_Scale}"
    SwfFile="{_SwfFile}"
    ZoomTransition="{_ZoomTransition}"
    ZoomTime="{_ZoomTime}"
    MinZoomSize="{_MinZoomSize}"
    MaxZoomSize="{_MaxZoomSize}"
    ZoomInterval="{_ZoomInterval}"
    FitPageOnLoad="{_FitPageOnLoad}"
    FitWidthOnLoad="{_FitWidthOnLoad}"
    EncodeURI="{_EncodeURI}"
    FullScreenAsMaxWindow="{_FullScreenAsMaxWindow}"
    ProgressiveLoading="{_ProgressiveLoading}" />
</mx:Application>

```

代码解释:

applicationComplete="initApp();" - 为初始化时调用initApp()方法

initApp() - 为获取web页面传递过来的FlexPaper参数、注册监听器

_xxxx:xxx - 存储web页面传递过来的FlexPaper参数

<fp:FlexPaperViewer id="FlexPaper".... - 调用FlexPaperViewer.mxml组件

xmlns:fp="com.devaldi.controls.flexpaper.*" - 定义fp, 指向flexpaper包

xxxx="{_xxxx}" - 设置FlexPaperViewer.mxml组件的参数

注: 由于以前完全未接触过FLEX, 所以不知道上面的介绍是否符合FLEX的一般说法。

id="FlexPaper", 这个id若改动请主要web页面和FlexPaperViewer.mxml对应的处理。

上面的代码中对比FlexPaperViewer.mxml, 会发现有几个参数是没有用到的, 如

AutoAdjustPrintSize、PrintPaperAsBitmap等, 如果我们想使用, 只要:

1) 在ajavaflexpaper.mxml添加AutoAdjustPrintSize="{_AutoAdjustPrintSize}", 需要定义
_PrintPaperAsBitmap。

2) 在Web页面中增加AutoAdjustPrintSize=xx参数

Mark想应该可以自己定义一些FlexPaper参数，当然这些参数要有后续的处理，需要对源码进行更深入的修改，这些都等各位高手来尝试了。

5.3 删除 Flexpaper 浏览器的打印按钮

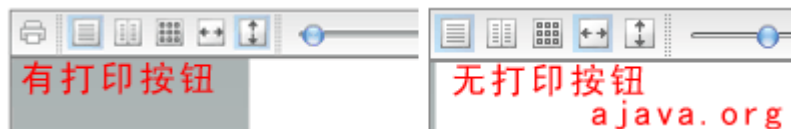
我们找到FlexPaperViewer.mxml文件中的下面部分内容：

```
<mx:HBox styleName="toolbarBackground" width="100%" height="26" borderStyle="solid"
verticalScrollPolicy="off" horizontalScrollPolicy="off" paddingTop="2"
paddingLeft="2" horizontalGap="3">
<st:GradientImageButton icon="{MenuIcons.PRINT_ICON}" styleName="toolbtttn"
width="20" click="printPaper(event)" toolTip="@Resource(key='Print',
bundle='FlexPaper')" visible="{!ReadOnly}" includeInLayout="{!ReadOnly}" />
<mx:Image source="{MenuIcons.BAR}" width="3" height="21" visible="{!ReadOnly}"
includeInLayout="{!ReadOnly}" />
```

把上面代码的st:GradientImageButton部分代码注释掉就可以了

```
<!--
<st:GradientImageButton icon="{MenuIcons.PRINT_ICON}" styleName="toolbtttn"
width="20" click="printPaper(event)" toolTip="@Resource(key='Print',
bundle='FlexPaper')" visible="{!ReadOnly}" includeInLayout="{!ReadOnly}" />
<mx:Image source="{MenuIcons.BAR}" width="3" height="21" visible="{!ReadOnly}"
includeInLayout="{!ReadOnly}" />
-->
```

效果如下图：



这里的st:GradientImageButton调用的是xmlns:st="com.devaldi.skinning.*"包中的GradientImageButton.as文件，再调用其他文件的printPaper(event)处理。

5.4 删除、修改 Flexpaper 浏览器的 FP 关于我们按钮

a) 删除FP关于我们按钮

我们可以在FlexPaperViewer.mxml中找到关于FP的代码：

```
<mx:Canvas width="100%" id="cnv_info" horizontalScrollPolicy="off"
verticalScrollPolicy="off" height="100%">
<st:GradientImageButton id="btnInfo" left="{cnv_info.width-21}"
visible="false" icon="{MenuIcons.SMALL_GRAY}" styleName="toolbtttn" width="20"
height="20" click="showInfo(event);" toolTip="@Resource(key='About',
```

```
bundle='FlexPaper')" addedToStage="{new MenuIcons.SMALL_TRANSPARENT().width > 10}" />
    <st:GradientLabel id="lblProgress" left="{cnv_info.width-26}"
textAlign="center" y="4" fontSize="8" height="19" width="32" text="" tabIndex="0"/>
    <ct:Spinner id="spin1" left="{cnv_info.width-22}"
styleName="gradientlines" y="-1" startImmediately="false"
spinnerType="gradientcircle" width="22" height="22" />
</mx:Canvas>
```

此段代码不能通过注释来达到删除FP的效果，因为注释掉后会发现更多的问题，查找修改起来更麻烦，所以我们通过查找FlexPaperViewer.mxml文件会找到下面三行代码，我们把它们注释掉就可以达到效果。

```
//btnInfo.visible = false;
//btnInfo.visible = (new MenuIcons.SMALL_TRANSPARENT().width > 10);
//btnInfo.visible = (new MenuIcons.SMALL_TRANSPARENT().width > 10);
```

效果如下图：



某些朋友提到在FlexPaperViewer.mxml源码中所有的btnInfo.visible = true; 替换成 btnInfo.visible = false;mark测试过，好像不行，由于测试仓促，特别是上面的方法已经OK了，没心再研究，大家有兴趣可以试试。

b) 修改FP关于我们按钮（设想）

我们打开about.mxml文件，可以看到里面的源码，

```
<mx:TitleWindow xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute"
width="276" height="204" title="@Resource(key='About', bundle='FlexPaper')"
showCloseButton="true" creationComplete="Init()">

<mx:Button x="105.5" y="127" id="btnOk" label="OK"
click="PopUpManager.removePopUp(this);"/>
<mx:Image x="13" y="10" source="{MenuIcons.LOGO_SMALL}" width="157" height="41"/>
<mx:Label x="10" y="62" text="@Resource(key='Developedby', bundle='FlexPaper')"/>
<mx:Label x="10" y="98" text="@Resource(key='Formoreinformation',
bundle='FlexPaper')"/>
<mx:LinkButton x="160" y="98" label="devaldi.com" click="navigateToURL(new
URLRequest('http://flexpaper.devaldi.com/?ref=FlexPaper'), 'devaldi')"
fontWeight="normal" textDecoration="underline" color="#4134EB" paddingBottom="0"
paddingLeft="0" paddingRight="0" paddingTop="0"/>
<mx:Label x="165" y="19" text="1.5.0"/>
<mx:Label x="10" y="80" text="GPL 3 (non-commercial) License."/>
```

设想1. 如果我们可以修改目前Resource的值，可以吗，是不是可以改变FP提示框中显示的内容？

设想2. 如果我们创建一些Resource, 修改about.mxml和FlexPaperViewer.mxml绑定的资源为自己创建的资源, 是不是可以改变FP提示框中显示的内容?

这些都只有等各位高手来测试, mark不想研究了, 也没时间, 呵呵。

5.5 删除 Flexpaper 浏览器的 LOGO

下面是某位高手的对FlexPaperViewer LOGO的处理:

1. 去掉加载页的进度条的 LOGO 图片

源文件: DupImage.as

找到语句: `private var _skinImg1:Bitmap = new MenuIcons.LOGO_SMALL();`

替换为: `private var _skinImg1:Bitmap = new Bitmap();`

2. 去掉工具栏最右边[关于...]对话框的显示的按钮

源文件: FlexPaperViewer.mxml

将源码中所有的 `btnInfo.visible = true;` 替换成 `btnInfo.visible = false;`

3. 除去初始装载时的 LOGO 图标

源文件: FlexPaperViewer.mxml

将控件 `lblProgress` 和 `spin1` 的初始属性分别加入: `visible="false"`

4. 除去加载页的 LOGO 图标

源文件: FlexPaperViewer.mxml

将`<fp:FlexPaperViewer_Base ...>`的属性最后行加入 `creationComplete` 处理:

```
<fp:FlexPaperViewer_Base xmlns:mx="http://www.adobe.com/2006/mxml"
                        addedToStage="bindStageEvents(event)"
                        xmlns:fp="com.devaldi.controls.flexpaper.*"
                        xmlns:ct="com.devaldi.controls.*"
                        xmlns:st="com.devaldi.skinning.*"
                        creationComplete="initApp()">
```

然后加入处理 `initApp()`函数到`<mx:Script>`中

```
//-----二次编程加入
        public function initApp():void {
            paper1.removeElementAt(2);
        }
```

最后将源码中的函数 `papersLoadingHandler(event:Event)` 最后一行加上: `paper1.removeElementAt(2);`

Mark只测试过第四点, 在FlexPaperViewer.mxml文件中找到下面的方法, 并在最后一行加入 `paper1.removeElementAt(2);` 结果可以达到去掉FlexPaperViewer右下角的logo。

```
protected override function papersLoadingHandler(event:Event):void{
    //mark edit 2012-1-25
    //
    spin1.start();
    //
    spin1.visible = true;
    spin1.visible = false;
    super.papersLoadingHandler(event);
```

```
//mark add 2012-1-25
paper1.removeElementAt(2);
}
```

效果如下：



5.6 解决中文 SWF 不能读取问题

这里的中文SWF不能读取问题是指：当SWF文件名为中文时，FlexpaperViewer不能读取到指定的SWF文件，会显示一直处于加载中。这个问题是mark最需要解决的，因为ajava.org网站的PDF基本都是中文名，如果都要改成英文名，那处理起来会很麻烦。

我们先作下面的测试：

测试一、在windows xp系统下，test.html、flexpaper_flash.js无任何改动，将上面的ajavaflexpaper.swf改名为FlexpaperViewer.swf（或使用原有的），test.html中SwfFile：escape('中文ajava.swf')

结果：FlexpaperViewer无法加载中文SWF。

测试二、在windows xp系统下，test.html、flexpaper_flash.js的文件编码都是utf-8（文本编辑器中另存时选择编码为utf-8），将上面的ajavaflexpaper.swf改名为FlexpaperViewer.swf（或使用原有的），test.html中SwfFile：escape('中文ajava.swf')

结果：FlexpaperViewer无法加载中文SWF。

测试三、在windows xp系统下，test.html、flexpaper_flash.js的文件编码都是utf-8（文本编辑器中另存时选择编码为utf-8），将上面的ajavaflexpaper.swf改名为FlexpaperViewer.swf（或使用原有的），test.html中SwfFile：decodeURI('中文ajava.swf')，flexpaper_flash.js中第24行增加decodeURI，SwfFile：decodeURI(config.SwfFile)，

结果：FlexpaperViewer可以加载中文SWF。

测试四、在windows 2003系统下，按顺序测试上面的1-3

结果：三种情况FlexpaperViewer都不可以加载中文SWF。

测试五、在linux系统下，按顺序测试上面的1-3

结果：三种情况FlexpaperViewer都不可以加载中文SWF。

从上面的测试结果来看，中文SWF在windows 2003系统、linux系统中使用中文SWF无论是使用原有的还是ajavaflexpaper.swf改名的FlexpaperViewer.swf，都不能解决加载中文SWF问题。

我们采用以下的方法来寻求解决之道：查找FlexpaperViewer.mxml、FlexpaperViewer_base.mxml文件，找出SwfFile相关的内容，通过Alert.show()方法显示出来，检查是哪里的的问题。

主要检查要注意的代码有：


```
Ajavaflexpaper.mxml - initApp()
FlexpaperViewer_base.mxml - loadSwf(s:String)
FlexpaperViewer_base.mxml - toggleFullScreen()
FlexpaperViewer_base.mxml - getUrlParamaters()
Viewer.as - set SwfFile(s:String)
```

通过检查上述方法，会发现无法加载中文SWF问题的原因是：

Web页面中SwfFile : escape(' 中文ajava.swf ')参数与Viewer.as - set SwfFile(s:String)方法中的参数处理导致问题。

Mark作了以下修改：

1. test.html、flexpaper_flash.js的文件编码都是utf-8
2. test.html中SwfFile : escape(' 中文ajava.swf ')改为SwfFile : decodeURI(' 中文ajava.swf ')
3. Viewer.as - set SwfFile(s:String)方法中，修改如下代码：

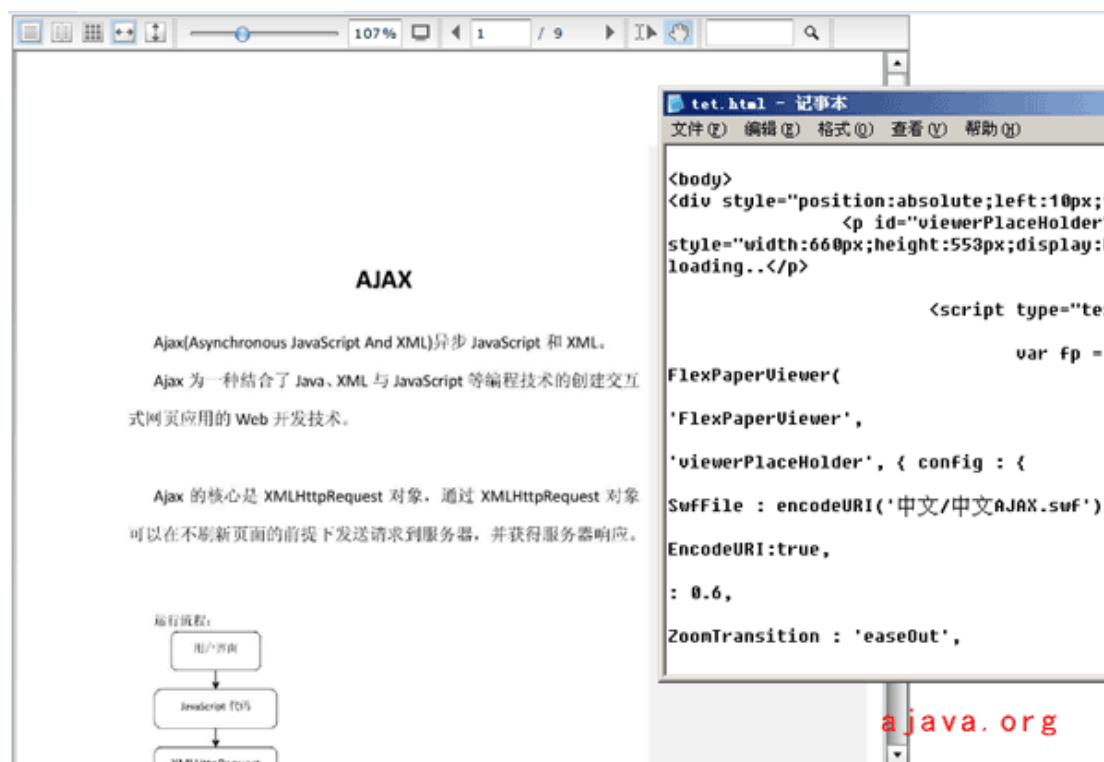
```
-----
if(EncodeURI)
    s = unescape(s);
改为
if(EncodeURI)
    s = decodeURI(s);
-----

if(!pagesSplit){

    _swfFile = s;
    if(EncodeURI)
        _swfFile = encodeURI(s);
    else
        _swfFile = s;
}
else
    _swfFile = s;
改为
_swfFile = s;
```

修改完成后，重新编译，在3种测试环境中测试，会发现中文SWF可以加载了，呵呵。

假如你的中文SWF文件是放在中文文件夹中，那可以读取到吗？答案是YES，没问题，你可以自己试试^_^



关于 SWF 中文名中特殊字符的使用

- 1) 系统禁止使用: / \ * ? < > |
- 2) 英文符号 % & ' + 不可用,
- 3) 中文符号 % & ' + 不可用, 可用 、 ? 《 》
- 4) 要注意英文大小写的不同

5.7 Flexpaper 二次开发的一些常见错误及解决

无论你使用Flexpaper的swc版本的源码进行开发, 还是下载源码进行二次开发, 可能遇到一些问题, mark将遇到的一些问题总结一下:

问题1: 仅主题为“halo”的类型“mx.containers.VBox”支持样式“borderThickness”。

FlexPaperViewer_Base.mxml/flexpaper/src/com/devaldi/controls/flexpaper 第 15 行 Flex 问题

解决: 右键项目, 选择属性, 在flex4编译器的Flex SDK版本中选择使用flex3兼容模式。

问题2: F:\flashb4\Adobe Flash Builder 4\sdk\4.0.0\frameworks\libs\spark.swc 的依赖项 spark.skins.spark.VScrollBarSkin 支持的版本最低为 4.0.0, 高于兼容版本 3.0.0。

flexpaper 未知Flex 问题

解决: 因为选择你下的Flexpaper版本使用了spark.swc, 所以不能使用flex3兼容模式。

问题3: 无法为区域设置“zh_CN”解析资源束“FlexPaper”。 flexpaper 未知
Flex 问题

解决：因为编译器找不到“zh_CN”资源，可检查你安装的Flash Builder的
sdfs\xxx\frameworks\locale是否有对应的文件，xxx为你选择的编译器Flex SDK版本。你可以在编译器中添加-locale zh_CN -source-path=locale/{locale}
-keep-all-type-selectors=true参数，指定采用的locale为Flexpaper根目录下的locale。

问题4：页面如何动态显示SWF文件

解决：在页面的SwfFile : encodeURI('中文AJAX.swf')改成SwfFile : encodeURI(“你使用的语言输出的SWF文件地址”)，如php: <?php echo \$url; ?>, java: <% out.print(url);%>, 要注意输出URL的编码处理。

问题5：官方例子的页面不能显示SWF文件或提示沙箱错误。

解决：到adobe官方，增加SWF文件或上级目录到可信任列表，前文中有讲述。

问题6：页面一直显示Document loading..

解决：检查是否有falsh player环境，检查flexpaper_flash.js的路径是否正确。

问题7：调用SWC库、源代码进行开发，自建的mxml应用出现某些flexpaper参数错误

解决：要注意SWC库、源代码的版本，有些版本的参数有所变动，最好看看官方的文档。

问题8：源码修改、test.html里面的参数、JS都没问题，但是设置参数参数后没效果。

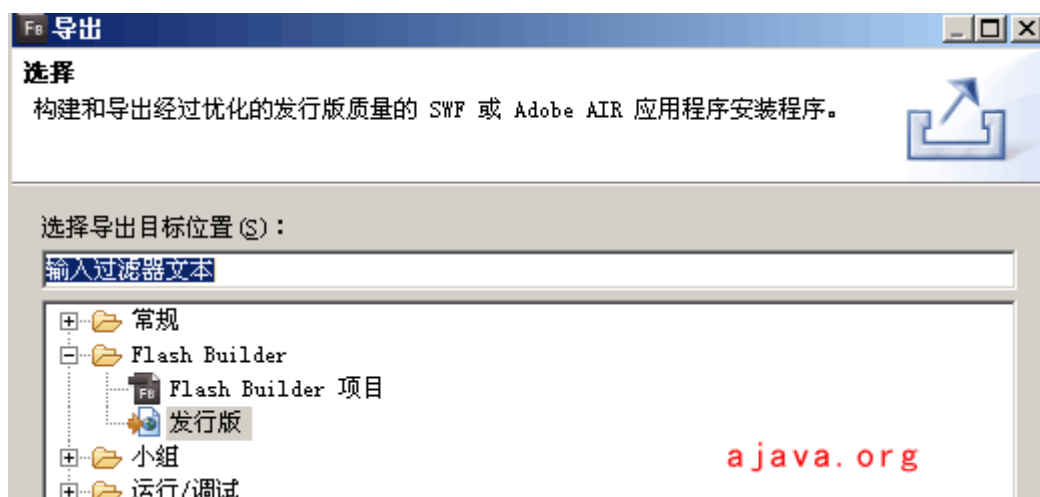
解决：出现这种情况要注意自己创建的mxml应用程序中<fp:FlexPaperViewer
id="FlexPaper"...>的id名称，因为JS里调用的是id名为FlexPaper，所以mxml应用程序的名称要相同，如果想改，应该要JS、html、mxml中统一修改。

5.8. Flexpaper 重新编译、打包

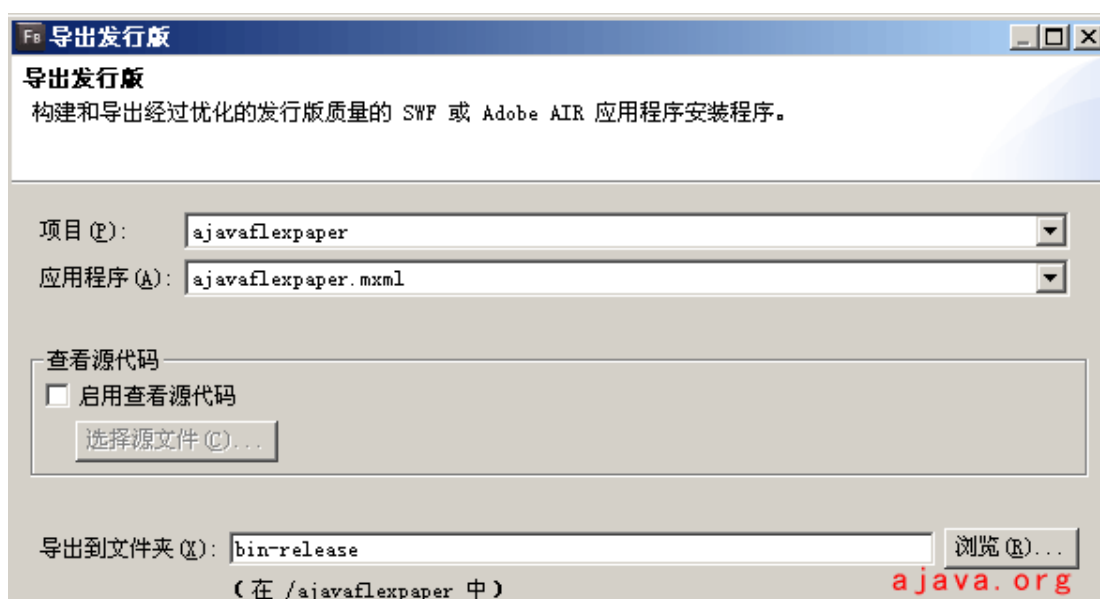
编译：我们每次保存、运行、调试、刷新，Flexpaper项目都会重新编译，每次操作后会发现bin-debug的ajavaflexpaper.swf会更新。

打包：经过重新编译，会发现bin-debug/ajavaflexpaper.swf的文件比较大，一般都有900kb左右，对比官网的FlexPaperViewer.swf，会发现官网的才500kb左右。我们只需选择导出方式生成发行版，就可以优化SWF，生成的SWF文件较小。过程如下：

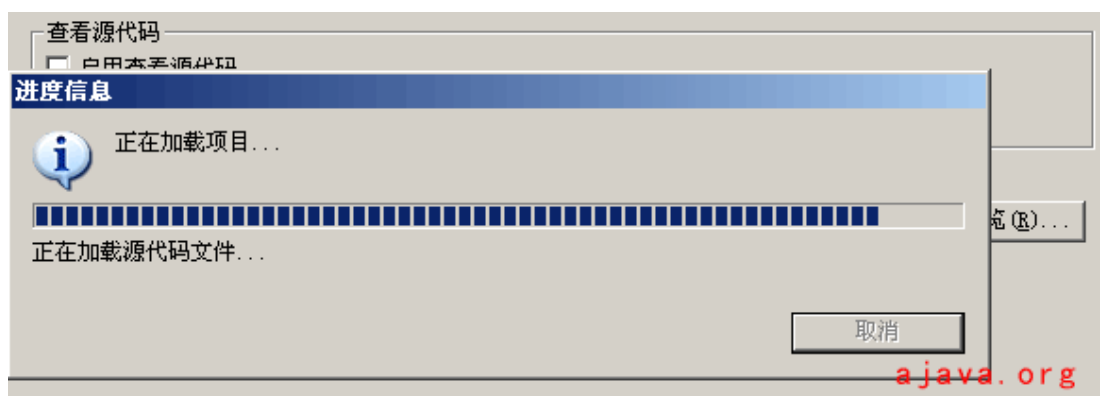
1) 右键项目，选择导出，会出现导出提示框，列表中选择“Flash Builder”->“发行版”



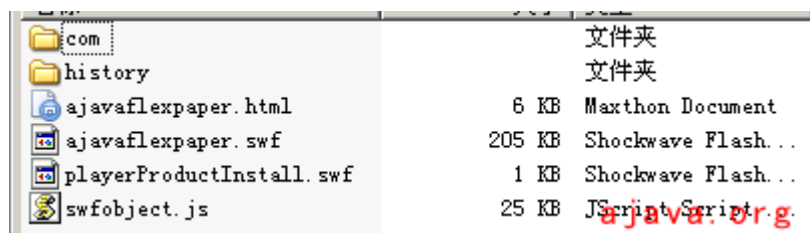
选择导出的项目和应用程序，可以设置导出SWF存放的文件夹



开始生成SWF文件



生成完毕后，可以看到bin-release文件夹下有ajavaflexpaper.swf，复制到其他地方，更改名为FlexPaperViewer.swf就可以使用



后记

在最后我要感谢一下寒冷的天气，如果不是今年天气特冷，mark也不会完成ajava.org网站功能后继续躲在家里写这篇教程。希望大家读完这篇基础教程，可以对FlexPaper有更深入的了解，开始进入FlexPaper更深层次的研究。

以后mark还会不会继续研究FlexPaper？答案是NO，如果不是ajava.org网站的需要，mark是不会去认识、了解FlexPaper，因为makr除了是只菜鸟还是条大懒虫。

本教程中只涉及到很简单的使用，一般大家可能还会涉及到上传、PDF转换SWF、自动转换、页面转换、PDF文字抽取、抽取中文问题处理等等，要实现其中一些或整套功能，只能靠大家了，期待各位的好作品。

最后宣传一下，本教程是由ajava.org独立JAVA社区的mark编写，大家转载请保留出处并保留完整的版本，欢迎大家来ajava灌水、交朋友。