

# Workflow Guideline

Design + Develop

# Overview

- Principle
- Design in Flash
- Export Assets for Development
- Develop in FDT

# Principle

好的流程应该做到：

- 表现层（视图）与应用层（逻辑）分离。
- 设计过程与开发过程可以独立地进行。
- 便于排错与测试。

# Project Folder Structure



/bin

发布目录

/build

用于Ant发布的模板文件

build.xml

/fla

Flash源文件

/lib

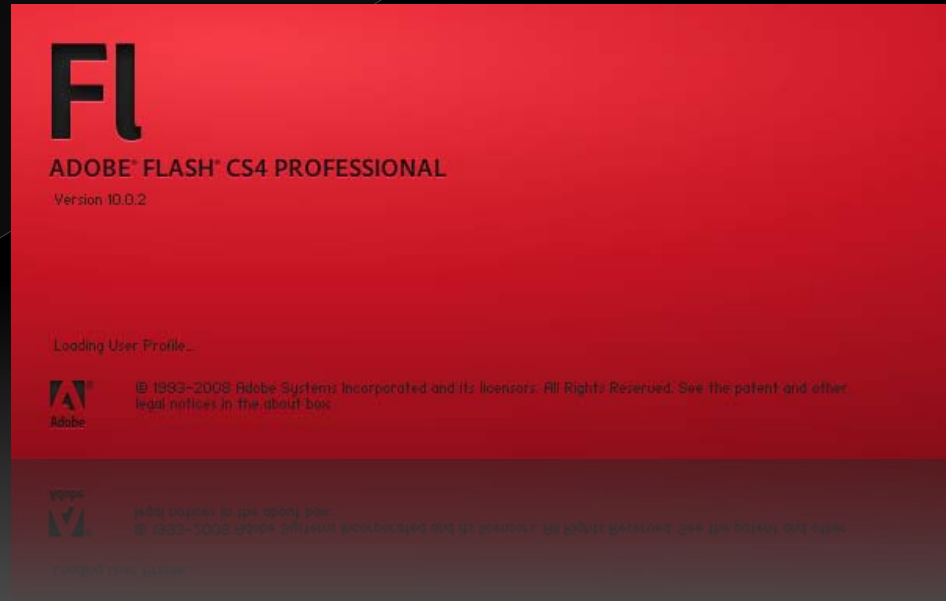
SWC类库和资源

/src

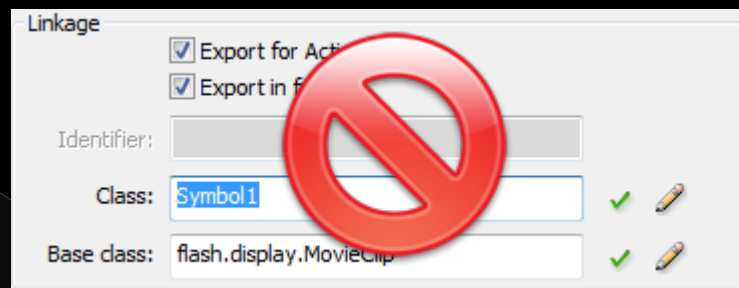
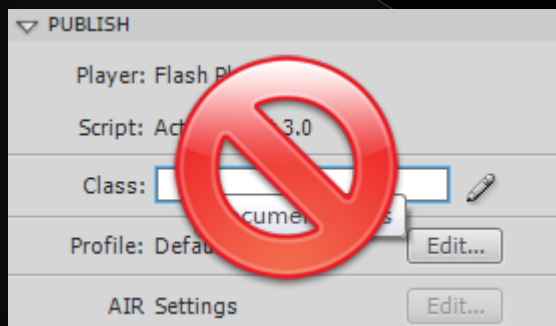
ActionScript文件

# Design in Flash

- Animation
- Visual asset
- Component skin



# Design in Flash



# 类绑定

# Design in Flash

- 尽可能少地在时间线上放置代码。
- 不要调用程序对象，而是发出事件。

```
dispatchEvent(new Event(Event.COMPLETE));
```

# Export Assets for Development

- SWC
- SWF

# SWC

- SWC文件包含各种资源（MC，Graphic，Sound...），和编译过的ActionScript代码。
- SWC通常用于打包组件和类库。
- Flash和Flex/FDT都可以导出SWC。

# SWF vs. SWC

## SWF

运行时加载，资源更新无需重新编译主文件。

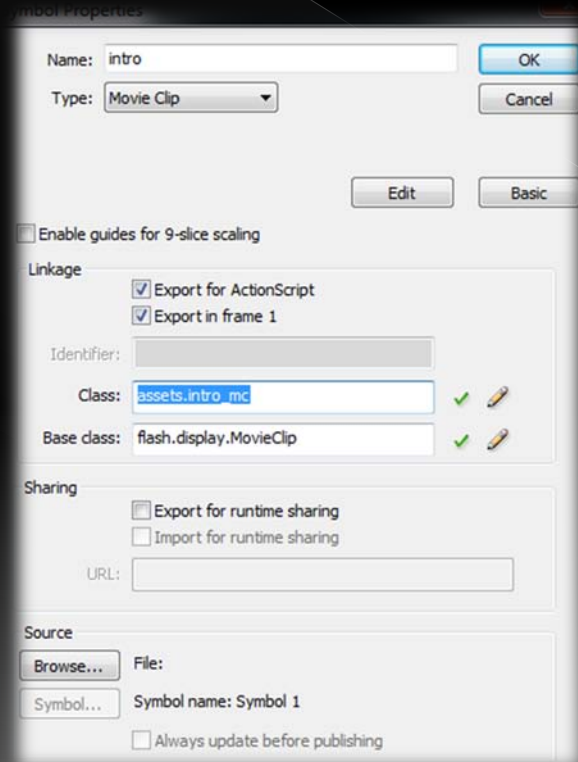
需要额外的代码提供加载和提取类型定义功能。

## SWC

资源编译到最终文件。

强类型，方便代码提示和排错。

# Export SWC in Flash



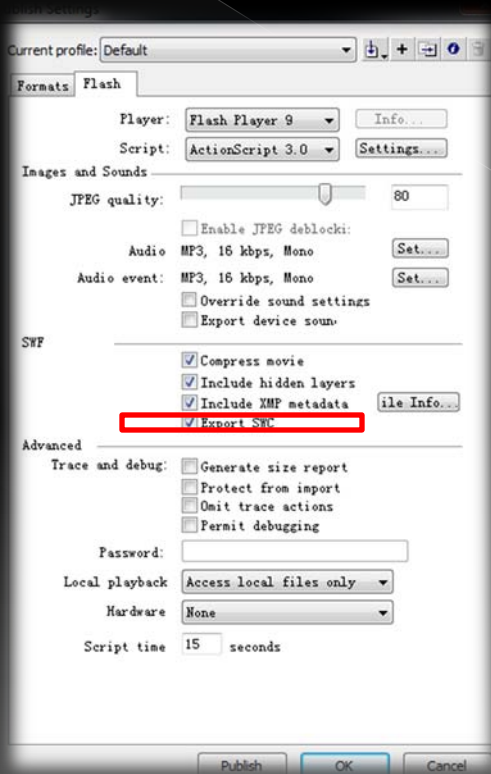
为ActionScript导出元件：



- ✓ 将SWF中的资源统一导出到assets包。
- ✓ 使用\_mc等后缀标识，把ActionScript类和在Flash中导出的资源区分开。

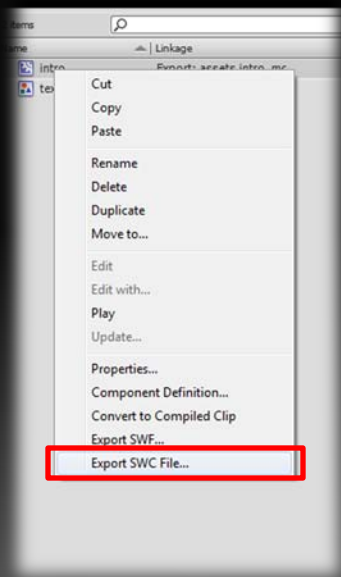
# Export SWC in Flash

在发布面板里导出SWC。



# Export SWC in Flash

我们还可以在库面板里对单独的元件导出SWC。



这种方式用于制作组件。导出的SWC自动引入了一个文档类（实现组件预览功能）：

```
fl.livepreview.LivePreviewParent
```

# Export SWC in Flash

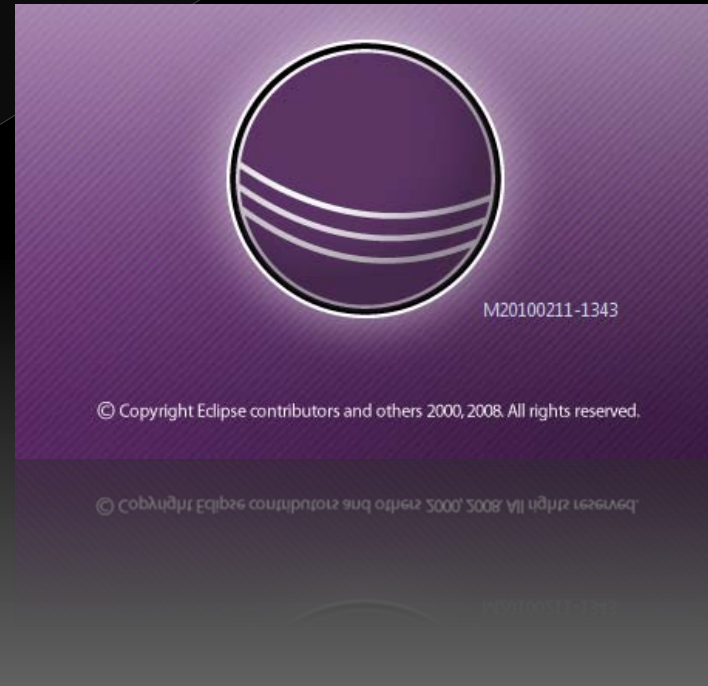
- 两种方式都会把所有包含“第一帧导出”选项的元件导出。
- 在工作流中我们使用第一种方式导出SWC。
- 使用FDT内置的文件浏览器可以直接查看SWC的资源结构。



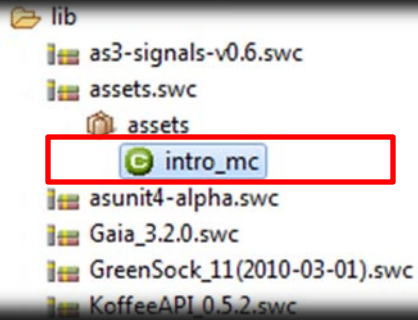
Flash在导出SWC的时候不会导出嵌套的脚本资源。如果内层MC包含了代码，也需要设置为导出。

# Develop in FDT

- API
- Component
- Model/Service
- Mediator/Page/View
- Debug
- Testing



# Use SWC



```
public function init() : void
{
    view = new intro_mc();
    view.addEventListener(Event.COMPLETE, completeHandler);
    addChild(view);

    resizeProxy.add(view, LayoutPreset.FIT_OUT);
}

public function destroy() : void {
```

# Use SWF

```
public class IntroTest extends Sprite

    private var loader : Loader;

    public function IntroTest()
    {
        loader = new Loader();
        loader.contentLoaderInfo.addEventListener(Event.COMPLETE, completeHandler);
        loader.load(new URLRequest("assets.swf"));
    }

    private function completeHandler(event : Event) : void
    {
        var introClass : Class = Class(loader.contentLoaderInfo.applicationDomain.getDefinition("assets.intro_mc"));
        var intro : MovieClip = new introClass();
        addChild(intro);
        loader.contentLoaderInfo.removeEventListener(Event.COMPLETE, completeHandler);
    }
}
```

# API Overview

- Gaia Flash Framework (SWFAddress, SWFWheel, Greensock Tween, MonsterDebugger included)

<http://www.gaiaflashframework.com/>

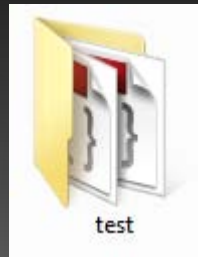
- AS3 Signals

<http://github.com/robertpenner/as3-signals>

- Robotlegs

<http://www.robotlegs.org/>

# Testing



测试用的ActionScript文件

# The Good

- 实现了设计和开发的工作完全独立进行。
- 更好的利用了FDT的代码编辑功能。
- 方便排错与测试。

# The Bad

- 只有程序员可以编译最终的文件，设计师没办法直观的看到最终效果。

# The Solution

- 开发组件供设计师使用。
- 额外提供一个基于Flash编译的测试环境，进行不同的单元测试。