

Flash. Flex. ActionScript 3.0

开发权威手册

基础入门 + 权威详解

张亚飞 编著



范例源代码
及入门视频

- ◆ 完全支持Flash CS5、Flash CS4、Flash CS3
- ◆ 以范例解释理论，通俗易懂；入门视频，让您轻松上手不含糊
- ◆ 将OOP融入Flash可视化开发，真正提高效率，实现团队协作
- ◆ 全新Flash Player API大曝光：新视频、3D模型、反向运动、IK骨架操纵
- ◆ 包含桌面AIR应用程序开发、网络应用开发、FMS开发，让您一本全拥有

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

Flash Flex ActionScript 3.0

开发权威手册

基础入门 + 权威详解

张亚飞 编著

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书是讲述 ActionScript 3.0 开发从入门到精通的权威手册,全面细致地介绍了 ActionScript 3.0 程序的编写方法以及语法规则,帮助用户建立正确的、良好的编程习惯,主要内容有:建立正确的 ActionScript 认知,纠正用户容易犯的错误,以及当前 Flash 开发领域内的多项错误认识, ActionScript 3.0 基本语法、面向对象开发,Flash 可视化开发, Flex 可视化开发, ActionScript 3.0 类库分类详解,引用实际案例介绍开发、编译、调试到部署这一完整的开发流程。

本书可以帮助开发者快速掌握基于时间轴开发的核心思想,从而提高工作效率。

本书适合从事 Flash 相关工作的人员使用。

图书在版编目(CIP)数据

Flash Flex ActionScript 3.0 开发权威手册:基础入门+权威详解+范例导航+最佳实践/张亚飞编著. --
北京:中国铁道出版社, 2010. 12
ISBN 978-7-113-11887-7

I. ①F… II. ①张… III. ①动画—设计—图形软件,
Flash Flex ActionScript 3.0—手册 IV.
①TP391.41-62

中国版本图书馆 CIP 数据核字 (2010) 第 173479 号

书 名: Flash Flex ActionScript 3.0 开发权威手册
——基础入门+权威详解+范例导航+最佳实践
作 者: 张亚飞 编著

责任编辑: 苏 茜
特邀编辑: 王 惠
封面设计: 付 巍
责任印制: 李 佳

读者热线电话: 400-668-0820
编辑助理: 王承慧
封面制作: 白 雪

出版发行: 中国铁道出版社(北京市宣武区右安门西街8号 邮政编码: 100054)
印 刷: 中国铁道出版社印刷厂
版 次: 2010年12月第1版 2010年12月第1次印刷
开 本: 880mm×1230mm 1/32 印张: 25.125 字数: 1100千
印 数: 3 000册
书 号: ISBN 978-7-113-11887-7
定 价: 59.00元(附赠光盘)

版权所有 侵权必究

凡购买铁道版图书,如有印制质量问题,请与本社计算机图书批销部联系调换。

前 言

真正的软件应用在于领略其“魂”，而非觊觎其“表”。

本书是 ActionScript 3.0 开发从入门到精通的权威著作，全面细致地介绍了 ActionScript 3.0 的程序编写方法以及语法规则，可以帮助用户建立正确的、良好的编程习惯。它包含下面 6 个方面的内容：

- (1) 建立正确的 ActionScript 认知，纠正用户经常容易犯的错误，以及当前 Flash 开发领域内的多项错误认识；
- (2) ActionScript 3.0 基本语法、面向对象开发；
- (3) Flash 可视化开发；
- (4) Flex 可视化开发；
- (5) ActionScript 3.0 类库分类详解；
- (6) 从基础入门到完全精通，从开发、编译、调试到部署，真正完整地开发知识。

为什么必须从这本书开始学习 ActionScript 3.0

本书系统地介绍如何将 ActionScript 3.0 面向对象开发融入 Flash Professional 创作环境中，从而实现 Flash 的可视化开发。

可视化开发的根本目的是提高工作效率、降低成本，这是目前软件开发领域主流的开发思想。可视化开发对每一个雇主和 ActionScript 3.0 开发人员来说都非常重要。因为雇主总是希望在让软件实现特定功能的前提下降低投入，这就对程序开发环境提出了要求。开发人员可以使用一个文本编辑器来手动编写代码，也可以使用可视化的开发工具通过拖放来完成复杂的手动编码，后者显然可以提高效率。

无数可视化开发软件（例如用于 .NET 开发的 Visual Studio，用于 Java 开发的 Eclipse、JBuilder、NetBean，以及 Delphi、PowerBuilder 等）正是为了适应这一要求诞生的。

Flash Professional 就是这样一个可视化开发软件，但它与我们熟知的上述可视化开发软件有很大不同。Flash 的可视化开发是以时间轴为基石的。

通过基于时间轴的 Flash 开发，不单单能够让开发者提高工作效率，还可以完美地让开发者和设计者团队协作。因为，设计者和开发者同样都是在以时间轴为基石进行工作。

本书可以帮助开发者快速掌握基于时间轴开发的核心思想，从而提高工作效率。当你去找 Flash 相关工作时，雇主一定会考虑你的开发是否有效率，或者，对于设计人员来说，也会考虑是否能与开发者在时间轴这个基石上协同工作。

本书的内容适合 Flash CS3 Professional、Flash CS4 Professional 版本。

保持与作者沟通

有时与作者沟通是十分必要的，读者可以从作者处获取知识的更新，或者勘误（如果书中有的话）。作者目前有以下联系方式：

E-mail: zhang-yafei@hotmail.com（首选方式）

Blog: <http://zhang-yafei.spaces.live.com/>

编 者

2010年7月

目 录

第 1 篇 ActionScript 语言基础入门

第 1 章 准备基础知识和配置开发环境	2
1.1 正确认识 ActionScript	3
1.1.1 正确认识 ActionScript 的开发工具	3
1.1.2 正确认识 ActionScript 运行环境和开发语言	4
1.1.3 关于标准和实现	5
1.2 获取和安装 Flash CS4 Professional	6
1.2.1 安装 Flash CS4 Professional	8
1.2.2 创建第一个 ActionScript 程序: Hello World	10
1.3 安装和配置 Flex Builder	12
1.3.1 安装 Flex Builder 3	13
1.3.2 安装 Flash Player 调试器版	14
1.3.3 创建第一个 ActionScript 程序: Hello World	14
1.4 使用本书的范例进行测试	16
1.5 Flash 软件和 Flex Builder 软件结合开发提高效率	17
第 2 章 程序开发的基石——声明变量和使用变量	18
2.1 变量的声明	19
2.1.1 声明和命名变量	19
2.1.2 变量和标识符的命名规范	21
2.1.3 变量的数据类型	21
2.1.4 自定义常量和内建常量	26
2.2 在程序中使用变量	28
2.2.1 为变量赋值和更改变量值	28
2.2.2 使用变量和获取变量值	32
2.3 变量的传值和传址	33
2.3.1 传值变量——按值传递	33
2.3.2 传址变量——按引用传递	34



第3章	ActionScript 语法规则和培养良好的编程习惯	35
3.1	空白和多行书写	36
3.2	点语法	37
3.3	卷曲花括号	37
3.4	分号	38
3.5	圆括号	39
3.6	字母的大小写	39
3.7	程序注释	39
3.7.1	单行注释和尾随注释	39
3.7.2	多行注释	40
3.7.3	文档注释	41
3.8	关键字	41
3.9	培养良好的编程习惯	41
3.9.1	通用的变量名、方法名、类名命名约定	42
3.9.2	重构	44
3.9.3	使用注释	44
3.9.4	养成控制内存的好习惯	45
3.9.5	使用弱引用	47
3.9.6	优化代码的一般常识	47
第4章	数据运算和运算规则	50
4.1	ActionScript 表达式	51
4.1.1	简单表达式和复合表达式	51
4.1.2	赋值表达式和单值表达式	52
4.2	算术运算符	52
4.3	赋值运算符	54
4.4	字符串运算符	55
4.5	比较运算符和逻辑运算符	56
4.5.1	考虑数据类型	57
4.5.2	全等运算符	58
4.6	位运算符	59
4.7	其他运算符	60
4.7.1	三元条件运算符	61

4.7.2	typeof 运算符	61
4.7.3	as 运算符	62
4.7.4	is 运算符和 instanceof 运算符	62
4.7.5	圆括号和逗号运算符的结合使用	62
4.7.6	delete 运算符	63
4.8	运算符的使用规则	64
4.8.1	运算符优先级规则	64
4.8.2	运算符结合规则	64
第 5 章	流程控制	66
5.1	语句和语句块	67
5.2	if...else 条件语句	68
5.3	switch 条件语句	72
5.4	do...while 循环	74
5.4.1	第一种模式	74
5.4.2	第二种模式	75
5.5	for 循环	76
5.6	for...in 循环	76
5.7	for each...in 循环	77
5.8	其他流程控制语句	78
5.8.1	break 语句	78
5.8.2	continue 语句	79
5.8.3	标签语句	80
5.9	流程控制总结	81
5.10	使用 with 语句	81
第 6 章	使用内建的全局函数	83
6.1	使用 trace() 函数	84
6.2	转义操作函数	84
6.2.1	escape() 函数和 unescape() 函数	84
6.2.2	encodeURIComponent() 函数和 decodeURI() 函数	85
6.2.3	encodeURIComponent() 函数和 decodeURIComponent() 函数	86
6.3	转换函数	86
6.3.1	parseFloat() 函数	86



6.3.2	parseInt()函数	87
6.3.3	Number()函数和 String()函数	87
6.3.4	Boolean()函数	87
6.3.5	int()函数和 uint()函数	88
6.3.6	XML()函数和 XMLList()函数	89
6.3.7	Vector()函数	90
6.4	判断函数	90
6.4.1	isXMLName()函数	90
6.4.2	isFinite()函数	90
6.4.3	isNaN()函数	91
第 7 章	自定义函数	92
7.1	函数的定义	93
7.1.1	函数定义的位置	93
7.1.2	创建可以共享的函数	95
7.1.3	测试函数的定义	96
7.2	函数的调用	96
7.3	参数的定义	98
7.3.1	在函数中定义参数的初始值	98
7.3.2	...(rest)参数——变参数函数	99
7.4	使用函数表达式定义函数	99
7.5	变量的作用范围和生命周期	101
7.5.1	了解变量的作用范围和生命周期	101
7.5.2	局部变量的声明	102
7.5.3	全局变量的声明	102
7.5.4	全局变量和局部变量的使用	103
7.5.5	易出错的变量使用	104
7.6	闭包函数（绑定函数）——一定要理解原理	104

第 2 篇 ActionScript 的面向对象编程和编译

第 8 章	面向对象编程（OOP）——创建类和使用类	108
8.1	澄清——必须首先建立正确的面向对象编程思想	109
8.1.1	对象的抽象	109

8.1.2	对象的使用	110
8.1.3	再谈“对象的抽象”和类	110
8.1.4	面向对象程序设计的优点和基本特征	111
8.1.5	对比——面向对象和结构化程序设计的区别	111
8.2	创建类	112
8.2.1	创建一个 ActionScript 类	112
8.2.2	类的定义	114
8.3	类的使用	116
8.3.1	怎样创建类的实例	117
8.3.2	Object 类的实例化	117
8.3.3	Array 类的实例化	118
8.3.4	访问对象属性	118
8.3.5	调用对象方法	118
8.3.6	调用静态方法和访问静态属性	119
8.4	定义方法	119
8.4.1	使用函数表达式定义方法	119
8.4.2	创建构造方法	120
8.5	定义属性	121
8.5.1	属性的初始化	121
8.5.2	使用 get 和 set 存取器方法定义属性	122
8.6	在类体内使用语句	125
8.7	在类内使用 this 关键字	126
8.7.1	闭包方法（绑定方法）	126
8.7.2	事件处理中的闭包	128
8.8	密封类和动态类	129
第 9 章	建立更可靠的 OOP 程序——类和成员的访问控制	131
9.1	控制成员访问	132
9.1.1	了解成员访问控制	132
9.1.2	public 成员（公共成员）	133
9.1.3	private 成员（私有成员）	134
9.1.4	internal 成员（包内成员）	135
9.1.5	protected 成员（受保护的成员）	137



9.2 类的可访问性.....	139
第 10 章 重复使用类——继承和组合.....	140
10.1 类的继承.....	141
10.1.1 创建子类.....	141
10.1.2 创建子类时的注意事项.....	142
10.1.3 关于多重继承的错误认识.....	143
10.1.4 使用关键字 super 调用基类中的方法.....	144
10.1.5 使用 super() 语句.....	144
10.1.6 扩展内建 ActionScript 类.....	145
10.2 在类中使用 final 修饰符禁止继承.....	145
10.3 类的组合.....	146
10.3.1 实现类的组合.....	146
10.3.2 组合的优点.....	148
10.3.3 了解 is-a 和 has-a.....	148
10.3.4 关于聚合和合成.....	148
10.4 了解原型继承.....	148
10.4.1 关于类继承.....	149
10.4.2 使用原型继承.....	149
第 11 章 静态成员、静态类和枚举（Enumeration）.....	151
11.1 为什么要定义静态成员和静态类.....	152
11.2 创建静态成员.....	152
11.2.1 使用静态成员.....	153
11.2.2 静态成员和继承.....	153
11.2.3 使用静态成员的注意事项.....	154
11.3 静态类.....	155
11.4 枚举.....	156
11.4.1 创建枚举.....	156
11.4.2 使用枚举的优点.....	157
第 12 章 多态（Polymorphism）——重载和覆载.....	158
12.1 方法的重载（overLoad）.....	159
12.2 重写方法和属性（overRide）.....	160

12.2.1	方法的覆载	161
12.2.2	属性的覆载	162
第 13 章	接口和抽象——错误最深重的地方	163
13.1	创建和实现接口	164
13.1.1	接口的定义	164
13.1.2	创建接口	165
13.1.3	接口的实现	166
13.1.4	在接口和实现接口的类中使用默认参数值	168
13.1.5	接口的合并	168
13.2	创建抽象类	168
13.2.1	什么是抽象类	169
13.2.2	定义抽象类	169
13.3	接口和抽象类的区别	170
13.3.1	区分接口和抽象类	170
13.3.2	唯一能使用抽象类的地方	171
第 14 章	类、接口和函数的组织——使用包	174
14.1	为什么要使用包——预防类的冲突	175
14.2	包的使用	176
14.2.1	使用包实现合理的分类管理	176
14.2.2	包的命名	178
14.2.3	访问类的方法和属性	178
14.2.4	使用包组织接口和实现接口	179
14.2.5	使用包组织函数和访问函数	180
14.3	类、函数的可访问性	181
14.3.1	单个 .as 文件内定义多个类	181
14.3.2	单个 AS 文件内类的继承	182
第 15 章	命名空间	184
15.1	创建和使用命名空间的基本步骤	185
15.1.1	定义命名空间	185
15.1.2	应用命名空间	186
15.1.3	引用命名空间	187



15.1.4	在包外应用命名空间	188
15.2	内建命名空间	189
15.2.1	访问控制属性	189
15.2.2	AS3 命名空间	189
15.2.3	flash_proxy 命名空间	190
15.2.4	object_proxy 命名空间	190
第 16 章	反射、数据类型检查和数据类型转换	191
16.1	数据类型检查	192
16.1.1	编译时进行类型检查	192
16.1.2	运行时进行类型检查	192
16.2	数据类型转换	193
16.2.1	强制数据类型转换	193
16.2.2	隐式数据类型转换	193
16.2.3	使用 as 运算符进行数据类型转换的注意事项	194
16.2.4	Upcasting (向上转型) 和 Downcasting (向下转型)	194
16.3	装箱 (Boxing) 和拆箱 (Unboxing)	195
16.3.1	装箱	195
16.3.2	拆箱	196
16.3.3	装箱后的数据类型检查	197
16.4	早期绑定和后期绑定	197
16.5	反射 (Reflection)	198
16.5.1	反射要实现什么	198
16.5.2	反射 API	198
16.5.3	类型的详细信息	200
第 17 章	使用 ASDoc 输出帮助文档	202
17.1	使用 ASDoc 工具	204
17.2	文档注释标签	204
第 18 章	处理程序异常——建立可靠的应用程序	210
18.1	错误和异常情况	211
18.2	异常情况处理	212
18.2.1	使用 try...catch...finally 代码块	212
18.2.2	使用 throw 语句	214

18.2.3	何时使用异常捕捉	215
18.3	使用 Error 处理异常错误信息	216
18.3.1	使用内建的异常类型	216
18.3.2	扩展 Error 以创建自定义错误	217
18.4	全局错误处理	219
18.4.1	当前 SWF 文档的全局错误处理	219
18.4.2	加载文档的全局错误处理	222
第 19 章	指定类路径 (Classpath)	224
19.1	在 Flash 中指定类路径	225
19.1.1	定义全局类路径	225
19.1.2	定义文档级类路径	226
19.2	使用 Flex Builder 指定类路径	227
19.3	编译器如何解析类引用	228
 第 3 篇 与 Flash 软件紧密结合的可视化应用程序开发		
第 20 章	使用 Flash 软件进行可视化开发	230
20.1	了解 Flash CS4 Professional 软件	231
20.1.1	了解时间轴、帧和层	234
20.1.2	使用场景	235
20.2	快速掌握 Flash 开发环境	236
20.2.1	使用“动作”面板在帧中编写 ActionScript 代码	236
20.2.2	自定义 ActionScript 编辑器环境	237
20.2.3	使用代码提示功能	238
20.3	第一个程序: Hello World	240
20.3.1	第一个可视化开发范例	240
20.3.2	使用纯代码开发范例	242
20.3.3	对比总结开发效率	243
20.3.4	文档类和 Flash 可视化开发的结合	244
第 21 章	基于时间轴的可视化应用程序开发	246
21.1	在帧中编写程序代码的原则	247
21.1.1	基本的程序定义规范	247
21.1.2	代码执行顺序	248



21.2	控制时间轴上帧的切换.....	249
21.2.1	控制时间轴的播放	249
21.2.2	时间轴变量初始化陷阱	250
21.3	创建基于时间轴的可视化应用程序.....	252
21.3.1	利用时间轴实现窗体——准备窗体图形元素素材	253
21.3.2	利用时间轴实现窗体——使用关键帧完成窗体状态.....	253
21.3.3	利用时间轴实现窗体——添加交互功能改变状态	255
21.4	使用时间轴实现向导窗体.....	257
21.4.1	准备一个影片剪辑用于实现向导.....	257
21.4.2	使用关键帧完成向导窗体状态	257
21.4.3	添加交互功能实现向导导航.....	260
21.5	利用时间轴分解状态和分组模块.....	264
21.5.1	状态分解的基本依据和最佳做法.....	264
21.5.2	主控部分	265
21.5.3	被控部分	266
21.5.4	子被控部分	267
21.5.5	通知状态改变	268
21.6	游戏开发中的时间轴应用.....	269
21.7	攻克可视化开发的核心原理	270
第 22 章	影片剪辑的嵌套和多时间轴.....	271
22.1	了解多时间轴	272
22.1.1	多时间轴的层次关系	273
22.1.2	定位通信的目标	273
22.1.3	关于绝对和相对路径	275
22.1.4	正确理解影片剪辑	276
22.2	影片剪辑的基本控制.....	277
22.2.1	影片剪辑的拖放	277
22.2.2	动态设置和获取影片剪辑的呈现属性.....	278
22.3	继承 MovieClip 类将影片剪辑元件作为其子类.....	280
22.3.1	创建 MovieClip 子类	280
22.3.2	将子类分配给影片剪辑元件.....	281

第 23 章 使用 Flash UI 组件	282
23.1 在 Flash 中使用组件的通用方法	283
23.2 改变组件外观	284
23.2.1 在运行时使用样式表改变组件的外观	284
23.2.2 在创作阶段手动改变组件的外观	286
23.3 结合 UIScrollBar 组件创建滚动文本框	286
23.3.1 创建滚动文本框	287
23.3.2 改变组件的外观	288
23.4 使用数据填充组件	290
23.4.1 基于 List 的组件	290
23.4.2 在创作环境中使用数据提供者	293
 第 4 篇 与 Flex Builder 软件紧密结合的可视化应用程序开发	
第 24 章 基于窗体的可视化应用程序开发	296
24.1 创建第一个 Flex 应用程序	297
24.1.1 创建 Flex 工程	297
24.1.2 构建窗体界面	300
24.1.3 编写 ActionScript 代码	301
24.1.4 编译和测试	302
24.2 对比 Flex 与 Flash 可视化应用程序开发	302
 第 25 章 使用 Flex Builder 软件	304
25.1 快速掌握 Flex Builder 软件	305
25.1.1 关闭和打开透视图	307
25.1.2 自定义透视图	308
25.2 使用开发透视图	309
25.2.1 可视化布局	309
25.2.2 使用代码提示功能	310
25.2.3 使用大纲视图	313
25.3 在<mx:Script>标签中定义 ActionScript 代码	313
25.4 应用程序编译和运行设置	314
25.4.1 自动编译和 SDK	314
25.4.2 优化编译和查看源代码	314

25.4.3	运行和测试应用程序	315
25.4.4	发布和部署 Flex	316
25.5	在窗体中使用控件	317
25.5.1	控件的种类	317
25.5.2	基本的属性设置	319
25.6	在窗体中使用容器组件	320

第 5 篇 显示对象和互操作

第 26 章	显示对象、显示对象容器和显示列表	324
26.1	显示对象和显示对象容器	325
26.1.1	有哪些类的实例可以作为显示对象	325
26.1.2	有哪些类的实例可以作为显示对象容器	326
26.2	管理显示列表	326
26.2.1	在显示列表中添加显示对象	327
26.2.2	关于 Sprite 和 MovieClip	328
26.2.3	深度和显示叠加顺序	328
26.2.4	添加显示对象	330
26.2.5	操纵容器中的显示对象	330
26.2.6	关于显示对象实例名和显示列表标识	333
26.3	影片剪辑的复制和删除	334
26.3.1	复制影片剪辑	334
26.3.2	删除影片剪辑	337
第 27 章	事件处理和基本的交互控制	338
27.1	基本事件处理	339
27.1.1	了解怎样实现事件处理	339
27.1.2	处理事件	340
27.1.3	了解事件流	341
27.1.4	事件监听	345
27.1.5	阻止冒泡	346
27.1.6	可用的基本交互事件	347
27.2	响应键盘事件	348
27.2.1	获取按键	349