

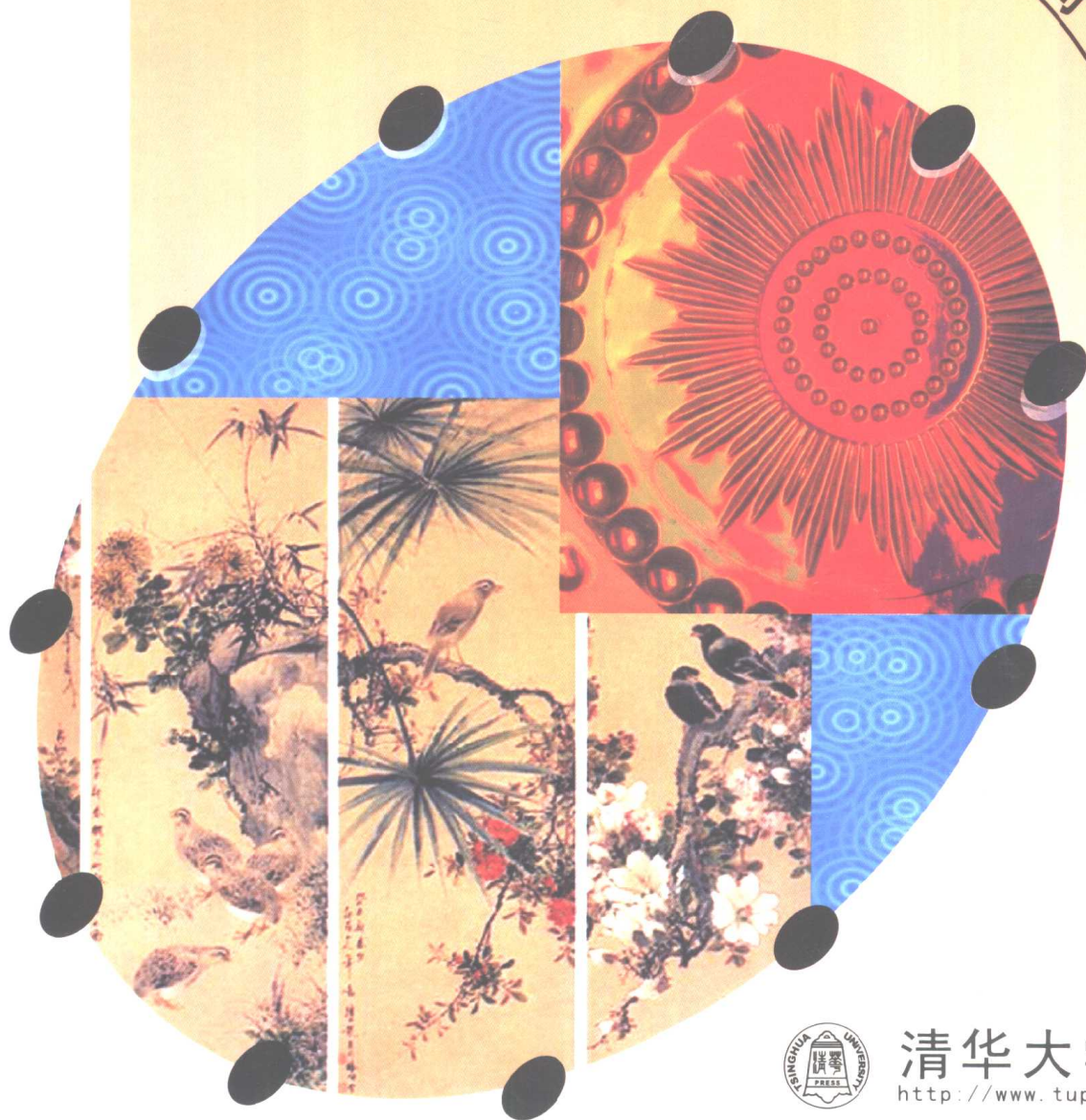


专题应用丛书

Authorware

朱诗兵
等
编著

Authorware 与多媒体编程



清华大学出版社
<http://www.tup.tsinghua.edu.cn>

Authorware 与多媒体编程

朱诗兵 等编著

清华大学出版社

(京)新登字 158 号

内 容 简 介

本书重点介绍了 Authorware 多媒体编程中涉及到的变量、函数、表达式和脚本语句,以及用户自定义函数、Windows API、ActiveX 控件、Xtras、数据库操作、知识对象和一些常用编程技巧等方面的热点技术。

通过本书的学习,不仅可提高 Authorware 的多媒体编程能力,还可以了解、创建拓展 Authorware 功能的用户自定义函数、动态链接库、ActiveX 控件、Xtras 和知识对象等。全书共分 12 章,全面讲解与实例紧密配合,内容详实,结构清晰,技术分析透彻,编程实例丰富,适合从事多媒体制作的广大用户参考、使用。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

书 名: Authorware 与多媒体编程

作 者: 朱诗兵 等

出 版 者: 清华大学出版社(北京清华大学学研大厦,邮编 100084)

<http://www.tup.tsinghua.edu.cn>

责任编辑: 田在儒

印 刷 者: 北京市清华园胶印厂

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 印张: 18.5 字数: 457 千字

版 次: 2001 年 2 月第 1 版 2001 年 2 月第 1 次印刷

书 号: ISBN 7-302-01295-4/TP·487

印 数: 0001~5000

定 价: 26.00 元

MJS172/04

前 言

随着多媒体技术及其相关技术的发展，多媒体制作得到了广泛的重视，应用范围越来越广。本书着重于 Authorware 的多媒体高级编程，从 Authorware 的变量、函数讲起，全面详实地对拓展 Authorware 功能的用户自定义函数、Windows API、ActiveX 控件、Xtras、数据库操作、知识对象等热点技术进行了阐述，增强了 Authorware 的多媒体编程能力。这是其他任何同类书所没有的。

本书在结构上共分 12 章。第 1 章介绍多媒体素材的管理；第 2 章介绍变量、函数、表达式和脚本语句的应用；第 3 章介绍用户自定义函数的应用；第 4 章介绍 Windows API 的访问；第 5 章介绍 OLE 对象的链接与嵌入；第 6 章介绍 ActiveX 控件的应用；第 7 章介绍 Authorware 对数据库的操作；第 8 章介绍 Xtras 的应用；第 9 章介绍知识对象的应用；第 10 章介绍 Authorware 的网络功能；第 11 章介绍多媒体作品的分化；第 12 章介绍 Authorware 的一些编程技巧。

本书的讲解循序渐进、内容图文并茂，并配合大量的实例，使得 Authorware 多媒体编程简单易懂。本书的所有实例都是作者精心准备的，操作性很强，读者只要认真阅读，参照随书发行光盘中的实例，按照书中所讲述的步骤操作，就可掌握讲授的内容，提高自己的 Authorware 多媒体编程能力。

本书主要由朱诗兵、李迎春编著，王擎天教授审校并定稿。在本书的编写过程中，李学军参加了第 9 章的编写，另外参加编写工作和提供各种帮助的还有：穆道生、卜格鸿、卢斌、刘伟、吴刚、赵忠文等，文字录入方面得到了纪宁宁的大力支持，由于他们的帮助使本书得以顺利完成，在此表示深深的谢意。

限于水平，书中难免存在一些缺点和错误，希望广大读者批评指正。

朱诗兵

2000 年 9 月 7 日

目 录

第 1 章 多媒体素材管理	1
1.1 Authorware 多媒体编程的步骤.....	1
1.1.1 创建新的项目文件.....	2
1.1.2 设置新文件的属性.....	2
1.2 外部媒体内容文件和外部媒体管理器的使用	5
1.2.1 建立外部连接.....	6
1.2.2 使用表达式定义链接.....	6
1.2.3 编辑一个外部文件的连接.....	7
1.2.4 改变外部链接.....	7
1.2.5 修复一个已建立的外部连接.....	7
1.2.6 重新建立链接关系.....	7
1.3 媒体库的创建与使用	8
1.3.1 媒体库的概念、特点.....	8
1.3.2 媒体库窗口的使用.....	9
1.3.3 媒体库的基本操作.....	11
1.3.4 媒体库的编辑与使用.....	12
1.3.5 链接关系的识别与修复.....	13
1.4 模块的创建与使用	15
1.4.1 模块的概念	15
1.4.2 模块关键特征.....	16
1.4.3 模块的创建	16
1.4.4 模块的使用	17
1.4.5 模块格式转换.....	18
1.5 小结	19
第 2 章 变量、函数、表达式和脚本语句的应用	20
2.1 变量、函数、表达式和脚本语句的使用位置	20
2.2 变量	23
2.2.1 系统变量	23
2.2.2 自定义变量	27
2.2.3 变量的不同存储类型.....	30
2.2.4 与变量有关的工作.....	31
2.3 函数.....	31

2.3.1	系统函数.....	31
2.3.2	自定义函数.....	36
2.3.3	与函数有关的问题.....	36
2.4	表达式.....	36
2.4.1	表达式和运算符.....	36
2.4.2	求解表达式的方法.....	39
2.4.3	使用表达式.....	40
2.5	脚本语句.....	41
2.5.1	If...then 语句.....	42
2.5.2	Repeat 循环.....	43
2.5.3	与编写脚本语句有关的问题.....	44
2.6	变量、函数、表达式和脚本语句的应用技巧.....	44
2.6.1	在 Authorware 中获取系统信息.....	45
2.6.2	在屏幕中间运行程序.....	47
2.6.3	Authorware 动态显示函数图像.....	47
2.6.4	文件调用时的变量共享.....	49
2.6.5	利用变量制作复合按钮.....	49
2.6.6	按钮出现时间的控制.....	51
2.6.7	在 Authorware 中实现拉框.....	52
2.7	小结.....	54
第 3 章	用户自定义函数的应用.....	55
3.1	用户自定义函数的载入.....	55
3.2	部分随带的 U32 函数详解.....	56
3.2.1	Apwkeys.u32.....	56
3.2.2	copyfile.u32.....	57
3.2.3	Cover.u32.....	57
3.2.4	Edit.U32.....	57
3.2.5	A5wmme.U32.....	58
3.3	Microsoft Agent 在 Authorware 中的应用.....	64
3.3.1	Microsoft Agent 的函数.....	65
3.3.2	Msagent.u32 的应用实例.....	67
3.3.3	分化包含 Microsoft Agent 的程序.....	68
3.3.4	Microsoft Agent 角色的创建.....	68
3.4	利用 tMsControls.u32 制作子菜单.....	69
3.5	动态 GIF 文件在 Authorware 中的显示.....	71
3.6	MIDI 音乐在 Authorware 中的播放.....	74
3.7	利用 BudApi.u32 进行声音控制.....	75

3.8 具有强大功能的 MjsApi.u32	79
3.8.1 自动动态改变用户的电脑屏幕分辨率设置	79
3.8.2 在 Windows 桌面上建立快捷方式	79
3.8.3 安装特殊字体保证 Authorware 使用的字体不变形	79
3.8.4 在 Authorware 中打开其他程序	80
3.8.5 用鼠标单击网址自动启动浏览器	80
3.8.6 控制窗口的最大化、最小化、关闭功能	81
3.8.7 合成和分解 Windows 标准 RGB 色彩值	81
3.8.8 获取和检测用户电脑光驱	82
3.9 编写自定义函数	82
3.9.1 用 Delphi 编写 Authorware UCD	84
3.9.2 用 VC++ 编写 Authorware UCD	87
3.10 小结	90
第 4 章 访问 Windows API	91
4.1 Windows API 函数的载入	91
4.2 不规则窗口的实现	94
4.3 快捷方式的建立	96
4.4 应用程序中实现关机功能	98
4.5 屏蔽 Ctrl-Alt-Delete	99
4.6 调用“关于”窗口	100
4.7 编写自己的动态链接库(DLL)	102
4.7.1 Win32 环境下动态链接库的编程原理	102
4.7.2 DLL 应用实例	105
4.8 小结	108
第 5 章 OLE 对象的链接与嵌入	109
5.1 创建 OLE 对象	109
5.2 OLE 对象的相关操作	111
5.3 OLE 处理函数	114
5.4 使用 OLE 对象的实例	115
5.5 小结	118
第 6 章 ActiveX 控件的应用	119
6.1 使用 ActiveX 控件的方法	119
6.1.1 ActiveX 控件的载入	119
6.1.2 注册 ActiveX 控件	120
6.1.3 对 ActiveX 控件的控制	122

6.2	ActiveMovie 控件的使用	122
6.2.1	ActiveMovie 控件的属性	122
6.2.2	ActiveMovie 控件的应用实例	123
6.3	Adobe Reader ActiveX 控件的使用	125
6.3.1	Acrobat Reader ActiveX 控件的语法格式	126
6.3.2	Acrobat Reader ActiveX 控件的应用实例	127
6.4	RichTextBox 控件的使用	128
6.4.1	RichTextBox 控件的语法格式	129
6.4.2	RichTextBox 控件的应用实例	130
6.5	映像控件的使用	133
6.5.1	Kodak 图像编辑控件的语法格式	133
6.5.2	Kodak 图像编辑控件的应用实例	134
6.6	应用 ActiveX 控件实现 Flash4 动画的播放	136
6.7	使用 VB 创建自己的 ActiveX 控件	138
6.7.1	创建 ActiveX 控件的步骤	138
6.7.2	创建 ActiveX 控件的具体方法	139
6.7.3	DataListView 控件的使用	145
6.8	小结	149
第 7 章	Authorware 对数据库的操作	150
7.1	利用 ODBC 操作数据库	150
7.1.1	ODBC 驱动程序	150
7.1.2	ODBC 数据源	151
7.1.3	自动配置 ODBC 数据源	155
7.1.4	SQL 语言的使用	157
7.1.5	ODBC.U32 的使用	159
7.2	利用 ActiveX 数据库控件操作数据库	162
7.2.1	创建 ActiveX 数据库控件	162
7.2.2	ActiveX 数据库控件的应用	166
7.3	用 ASP 实现对数据库的访问	171
7.4	小结	173
第 8 章	Xtras 的应用	174
8.1	使用 Xtras	174
8.1.1	Xtras 的种类	174
8.1.2	Sprite Xtras	175
8.1.3	Sprite 图标的属性	176
8.1.4	Scripting Xtras	178

3.1.5 安装 Xtra	180
8.1.6 发布 Xtra	181
8.2 事件响应	181
8.2.1 建立一个 Sprite Xtra	182
8.2.2 建立一个事件响应	182
8.3 Authorware 自带 Xtra 详解	184
8.4 Xtras 的典型应用：压缩与解压缩	186
8.5 小结	187
第 9 章 知识对象的应用	188
9.1 知识对象的类型	188
9.1.1 File（文件）类	189
9.1.2 Interface Components（交互界面组件）类	190
9.1.3 Internet（网络）类	192
9.1.4 New File（新文件）类	193
9.1.5 Tutorial（指南）类	194
9.2 知识对象的添加	194
9.2.1 在 New File 对话框中添加 File 类型的知识对象	194
9.2.2 在应用程序流程中添加 Knowledge Object	195
9.3 知识对象图标	196
9.3.1 创建知识对象图标	196
9.3.2 将知识对象的图标与一向导建立联系	197
9.3.3 知识对象图标选项	197
9.3.4 为知识对象图标赋予唯一的标识码（ID）	198
9.4 知识对象的创建	201
9.5 与知识对象有关的变量和函数	204
9.5.1 系统变量	204
9.5.2 系统函数	204
9.6 与知识对象的通信	205
9.6.1 使用 CallTarget 函数可完成的工作	205
9.6.2 可以在 CallTarget 中调用的函数	206
9.6.3 不能与 CallTarget 配合使用的函数	207
9.7 小结	208
第 10 章 Authorware 的网络功能	209
10.1 Authorware 的网络技术	209
10.1.1 传输速率	209
10.1.2 数据流技术	210

10.1.3	Authorware Advanced Streamer	211
10.2	基于 Web 的多媒体作品设计	213
10.2.1	设计前应考虑的问题	213
10.2.2	如何在多媒体作品中访问网络资源	214
10.3	多媒体作品的网上运行	220
10.3.1	网络打包	220
10.3.2	映射文件(Map Files)	222
10.3.3	将映射文件嵌入 HTML 页面中	225
10.4	Web 服务器的配置	226
10.4.1	Web 化打包文件的命名及其大小写	227
10.4.2	与 Authorware 相关的 MIME 类型映射	227
10.4.3	配置服务器	227
10.5	小结	228
第 11 章	多媒体作品的分化	229
11.1	分化需知	229
11.2	分化时所需要的文件	230
11.3	Authorware 的 INI 文件	233
11.4	非网络打包	236
11.4.1	打包需知	237
11.4.2	多媒体作品的打包	237
11.4.3	库文件单独打包	239
11.5	更改打包文件的图标	240
11.6	多媒体作品安装程序的制作	240
11.6.1	Authorware 制作多媒体安装程序	241
11.6.2	Visual Basic 打包程序制作多媒体安装程序	242
11.7	小结	252
第 12 章	Authorware 编程技巧荟萃	253
12.1	Authorware 的简单应用技巧	253
12.2	Authorware 制作“流水”按钮	258
12.3	Authorware 制作纯中文下拉式菜单	260
12.4	Authorware 调用可执行文件后的窗口处理	263
12.5	Authorware 播放 MP3 声音文件	264
12.6	Authorware 下浮动工具面板的定制	266
12.7	Au+thorware 中鼠标右键弹出式子菜单的实现	269
12.7.1	判断鼠标右键的单击	270
12.7.2	鼠标右键单击动作的实时处理	271

12.8 Authorware 中屏幕保护程序的设计.....	275
12.9 图文滚动的制作技巧.....	276
12.10 状态提示信息栏的实现.....	278
12.11 小结	280

第1章 多媒体素材管理

在进行 Authorware 多媒体编程时,对各种多媒体素材的制作,是一项最耗时的工作,而管理这些内容同样也是相当纷繁复杂的。Authorware 提供了四种管理工具,协助用户对程序进行管理以达到节省时间提高效率的目的。这四种工具分别是:外部媒体内容文件、外部媒体管理器、库和模块。使用这些工具,用户可获得如下收益:

- 无需复制,就可在程序中重复使用同样的内容。
- 可以简化对在程序中多次重复使用的内容的修改操作。
- 对于用户已设计好的一些结构化设计,可以重复利用而无需重新创建。
- 创建一个程序,其内容和流程结构可以被其他开发者所共享。
- 对程序进行打包操作后,仍可对其中的内容进行更新维护。

本章描述了在进行 Authorware 多媒体编程时,如何与多媒体素材打交道,也就是如何在多媒体程序中管理各种媒体。在具体阐述多媒体素材管理之前,首先必须了解应用 Authorware 进行多媒体编程的步骤,这对于后续章节的学习也是非常有益的。

1.1 Authorware 多媒体编程的步骤

为了更好地使用 Authorware 进行多媒体程序的开发,需要了解 Authorware 开发多媒体程序的具体过程。多媒体项目的创建过程,一般都包括四个步骤:

- (1) 编写脚本;
- (2) 收集和整理素材;
- (3) 构建程序框架并添加素材;
- (4) 程序调试和分发。

编写脚本是开发多媒体作品的第一步。进行一个多媒体项目的创意时,首先应该将相关人员,如美工、程序员、文字编辑人员等组织起来,大家一起对项目进行讨论,研究整体框架和风格,然后编写出相应的脚本文件。第二步根据脚本文件的内容,收集相关素材,例如整理文本、扫描图片、创作图像、制作片头等。第三步程序员根据创作构思和讨论的表现风格,构建程序框架,并且在 Authorware 中集成所有的素材,完成项目文件。当项目文件创作完成之后,在分发作品之前需要进行测试工作,查看程序中是否存在缺陷,这是第四步。经过测试后的项目文件就可以分发给用户了。

上面所提到进行多媒体创作的步骤,适合于多人组成的开发小组共同开发。如果是普通个人用户使用 Authorware 开发多媒体作品,那么编写脚本、收集素材以及完成程序都由同一个人来做。

1.1.1 创建新的项目文件

运行 Authorware 应用程序, 选择 File|New File 菜单选项, 系统将弹出一个 New File 对话框, 如图 1.1 所示。

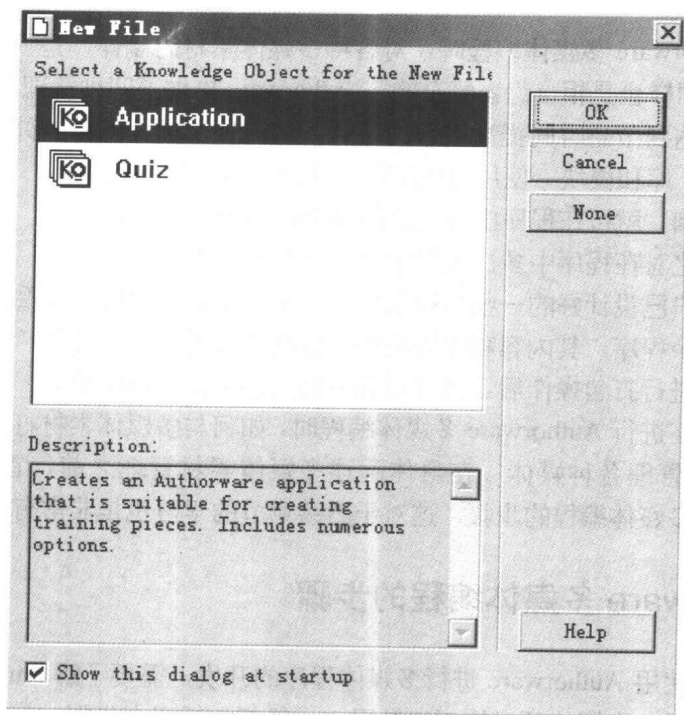


图 1.1 New File 对话框

在该对话框中, 提示用户是否为新的项目文件添加 Knowledge Object。此时在 New File 对话框中可以添加的 Knowledge Object 包括 Application 和 Quiz, 前者可以帮助用户创建有关学习培训的项目文件, 后者为用户提供了创建小测验类型的项目文件的模板。有关 Knowledge Object 的具体用法, 将在后面详细介绍。此时用户可以直接单击 Cancel 命令按钮, 取消添加 Knowledge Object。在 New File 对话框中, 用户应该注意到在左下角有一个 Show this dialog at startup 复选框。在默认情况下, 该复选框是被选中的, 代表每次开启新文件时都显示该对话框。如果用户不希望在开启新文件时显示该对话框, 可以在复选框上单击, 去掉选中标志。

1.1.2 设置新文件的属性

一般来说, 对每一个新建的文件都应设置该文件的属性。设置文件的属性时; 可以选择 Modify|File|Properties... 菜单选项, 系统将弹出一个 Properties: File 对话框, 如图 1.2 所示。在该对话框中包含了三个选项卡: Playback、Interaction、CMI。通常需要设置前两个选项卡的内容。

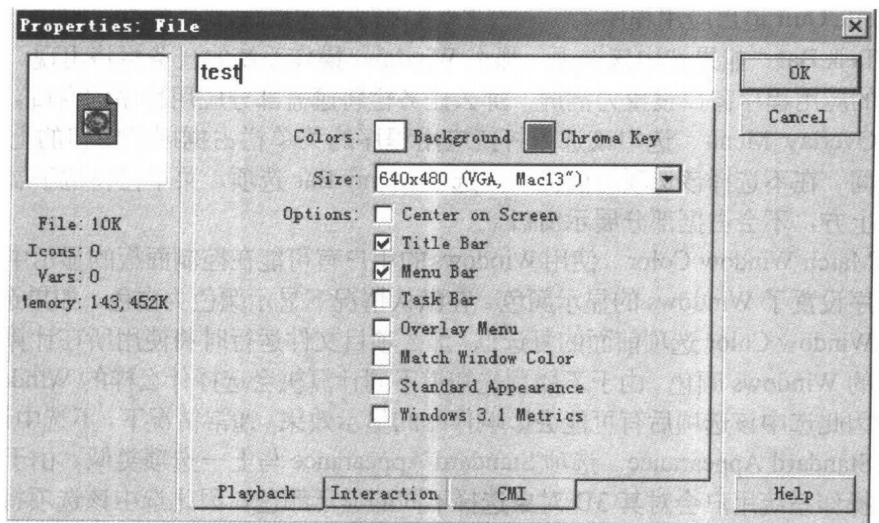


图 1.2 Properties: File 对话框

1. Playback 选项卡的设置

在图 1.2 中显示了 Properties: File 对话框的 Playback 选项卡中的设置内容。在该对话框中，用户可以设置整个项目文件演示窗口的前景色和背景色，演示窗口的显示大小和显示状态。下面将对该选项卡中的设置进行具体说明。

- **Colors 选项** 在 Colors 选项区域中有两个颜色块，分别代表着背景色和色度键。在代表背景色和色度键的颜色块上单击鼠标就可以更改相应的颜色设置。其中，背景色的颜色设置用来代表整个作品的背景颜色，系统默认情况下为白色。对色度键进行设置，需要用户计算机上配置的视频显示卡的支持。
- **Size 选项** 在 Size 下拉列表中，用户可以指定作品展示窗口的大小。单击 Size 下拉列表框右侧的向下箭头，可以看到系统预先设定的窗口大小类型。在这里，最常用的选项为“640×480 (VGA, Mac13”)”，以及最后一项 Use Full Screen。系统通过提供用户选择第一项 Variable 来扩展用户的窗口大小设定。如果选择了第一项 Variable，在文件中可以通过函数 ResizeWindow (width,height) 来设置展示窗口的大小。Size 下拉列表如图 1.3 所示。

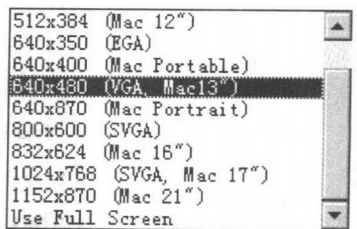


图 1.3 “Size” 的下拉列表选项

- **Options 选项** 在 Options 选项区域中，用户可以对展示窗口显示的其他一些特性进行选择。
- **Center on Screen** 选择该选项将使作品的展示窗口始终位于屏幕的中间。
- **Title Bar** 选中该选项后，在作品的展示窗口中将出现标题栏。
- **Menu Bar** 选中该选项后，在展示窗口中将出现菜单栏。用户可以通过选择菜单

File Quit 退出应用程序。

- **Task Bar** 如果选中该选项, 那么 Windows 操作系统的任务栏将出现, 如果用户的应用程序窗口设置为全屏, 那么任务栏将遮盖部分应用程序的窗口。
- **Overlay Menu** 选中该选项将使应用程序的菜单栏占据展示窗口的上面部分空间。在不选择该选项的前提下, 选中 Menu Bar 选项, 菜单栏将位于展示窗口的上方, 不会占据部分展示窗口。
- **Match Window Color** 使用 Windows 的用户有可能在控制面板的显示中为应用程序设置了 Windows 的显示颜色。在默认情况下显示颜色为白色。如果选中 Match Window Color 选项前面的复选框, 那么项目文件运行时将使用所在计算机上定义的 Windows 颜色。由于不能预先知道不同计算机会选择什么样的 Windows 颜色, 因此选中该选项后有可能会影响作品的显示效果。通常情况下, 不选中该复选框。
- **Standard Appearance** 选项 Standard Appearance 与上一选项类似。由于 Windows 操作系统用户会对其 3D 对象选择不同的显示颜色, 因此选中该选项将在作品中采用用户在 Windows 定义的 3D 对象的颜色。由于不能确定最终作品运行在何种设置的计算机上, 为了避免出现显示效果上的问题, 推荐不选中该选项的复选框。
- **Windows 3.1 Metrice** 在 Window 3.x、Windows 95、Windows 98、Windows NT 中, 菜单栏、标题栏以及边框的尺寸大小均不相同。如果用户选择了该选项前面的复选框, 那么作品的菜单栏、标题栏等都将采用 Windows 3.1 的尺寸。由于 Windows 操作系统的不断更新, 该选项在以后的 Authorware 版本中有可能不再出现。

2. Interaction 选项卡的设置

在 Properties : File 对话框中单击 Interaction 标签, 将打开 Interaction 选项卡, 如图 1.4 所示。

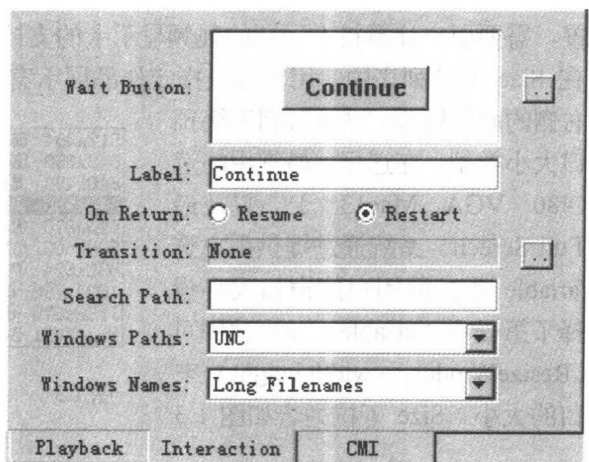


图 1.4 Interaction 选项卡

在 Interaction 选项卡中可以设置以下内容。

- **Wait Button** 用鼠标在 Wait Button 选项区域的 Continue 按钮上或者右侧的 “...” 按钮上单击, 系统将弹出 Buttons 对话框。在 Buttons 对话框中, 用户可以指定用于等待的图标类型。

- **Label** 在 Label 后面的文本框中可以输入整个作品中等待图标的名称, 在默认情况下, 等待按钮的名称为 Continue。
- **On Return** 该选项区域用来指定当用户重新进入程序时从何处开始。设置时, 可以选择 Resume 或者 Restart。如果选择 Resume, 那么当用户再次运行应用程序时, 将从使用者上一次退出的地方开始。Authorware 会自动记录用户在作品中的位置以及变量值, 并且把它保存在 Windows 的系统目录的 A5w_Data 目录下。如果选择 Restart, 那么每次使用者运行应用程序, 程序都从开始处执行。
- **Transition** 单击 Transition 文本框后面的“...”命令按钮, 将打开 Return Transition 对话框, 在该对话框中, 用户可以指定多媒体作品返回时的过渡效果。
- **Search Path** 设置搜索路径主要是针对那些保存在外部的文件而言。在 Authorware 中存在着两种保存媒体文件的方式: 内部和外部。当用户在 Authorware 的作品中导入一外部文件, 如动画文件、视频文件时, Authorware 会自动记录该外部文件的位置。当作品运行时, Authorware 会到相应的位置去寻找外部文件, 如果在相应位置处没有找到, 那么它会到 Search Path 中指定的路径下去搜索。

对于 Authorware 最后打包生成的可执行文件, 一个很重要的问题就是 Search Path 的设置。特别是作品最后保存在光盘或放到其他的计算机上运行时, 如果外部文件的路径改变了, 而 Search Path 中又未指定相应的搜索路径, 那么项目文件的运行将出现问题。

- **Windows Paths** 设置 Windows Paths 主要用来指定系统变量和函数返回的网络路径类型。在这里, 用户可以单击向下箭头在其下拉列表中选择 DOS (driver-based) 或 UNC (Universal Naming Convention)。选择何种类型与作品运行的网络环境有关。
- **Windows Names** Windows Names 用来设置系统函数和变量的返回文件名格式。在 Windows Names 下拉列表中选择 Long File Names (长文件名) 或 DOS (8.3)。设置 Long File Names, 运行文件名长度可达 255 个字符; 而 DOS (8.3) 使用 DOS 文件系统的命名格式 (8 个字符外加 3 个字符长度的后缀)。Windows Names 的选择主要依赖于作品最后的运行平台。如果作品最后运行在 Windows 95、Windows 98 或者 Windows NT 下, 那么应该选择 Long File Names; 如果作品最后的运行平台为 Windows 3.1 或者用户不能确定, 那么应该选择“DOS(8.3)”。

对于新文件的设置还包括 Font Mapping、Palette、Navigation Setup…。这些设置可参考帮助文件, 这里就不再介绍。

到目前为止, 用户已经学习了如何新建一个项目文件以及对新建文件进行属性设置。接下来的工作就是根据需求向流程线上添加相应的图标, 从而完成多媒体编程。

了解 Authorware 多媒体编程的步骤后, 下面就具体阐述 Authorware 的四种管理工具是如何进行多媒体素材管理的。

1.2 外部媒体内容文件和外部媒体管理器的使用

一个文件如果采用外部内容方式链接到应用程序中, 则应用程序中保存的仅是文件名

和路径。文件内容仍保存于程序之外。内容外置的优越性在于：它可以保证用户无论在开发阶段还是在程序打包之后都拥有对内容的修改权。一方面，程序的使用者可以在没有 Authorware 环境下，按需要更新内容；另一方面，可以保证程序执行过程中的内容更新，如随时在网上发布说明天气变化的卫星云图。只要文件的名称路径保持不变，这种链接就会一直保持下去。这种链接方式并非十全十美，用户应根据实际需要灵活掌握。

在程序开发阶段，使用外部媒体浏览器管理外部文件是十分方便的。用户使用它，可以改变一个文件或一组文件的路径，还可使用变量和表达式重新定义文件的路径。在程序设计初期，程序员往往希望把所有的外部媒体都放在同一个文件夹中，而随着开发工作的展开则更希望把这些媒体信息按种类分开。这时，使用外部媒体浏览器，只需一步操作就可改变整组文件的路径。

外部媒体浏览器不仅向用户提供如下信息：一个外部媒体文件的名称、路径、链接的图标、链接的完整性等，还允许用户进行诸如创建链接、修改链接、修复链接等多种操作，下面就分别加以介绍。

1.2.1 建立外部链接

将一个外部媒体以外部内容方式加载到应用程序中有下列两种操作方法。

- (1) 在使用输入对话框加载文件时，在 Import 对话框中选中 Link to File 选项。
- (2) 在将外部媒体文件拖放到流程线、设计窗口或库中的同时按住 Shift 键。

在按住 Shift 键进行拖放操作时，在光标的右下角处有一个标记表示该文件将以外部链接方式链入到程序中。

1.2.2 使用表达式定义链接

具体操作如下：

- (1) 选中 Window|External Media Browser 菜单选项，打开 External Media Browser 对话框，如图 1.5 所示。

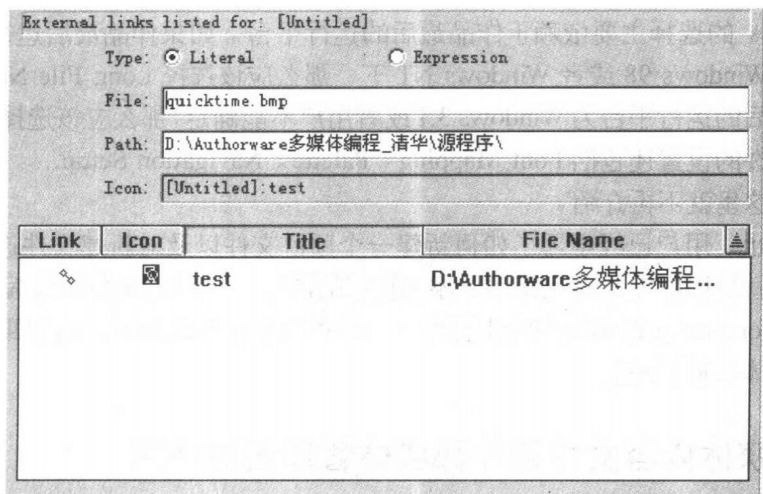


图 1.5 External Media Browser 对话框