



Authorware
编程必备

Authorware

变量、函数、控件 手册与范例

毕广吉 编著

- ◆ Authorware中高级编程人员案头必备
 - ◆ 235种系统变量的详解
 - ◆ 349种系统函数的详解
- ◆ 56个U32和DLL文件中700种函数的详解
 - ◆ 7个Xtras文件中67种函数的详解
- ◆ 20种内部控件和23种ActiveX控件的详解
 - ◆ 各种变量和函数按用途分类的索引
- ◆ 说明变量、函数、控件用法的149个范例
- ◆ 适用于Authorware 6.0, 6.5和7.0三个版本



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

内 容 简 介

本书以手册的形式分别对 Authorware 的系统变量、系统函数、U32 函数、Xtras 函数、Authorware 内部控件和 ActiveX 控件进行详细的解释,并为部分函数和变量编配了用法示例程序。读者除了能按字母顺序查找各个条目的详细资料以外,还能通过本书提供的按用途分类的索引,根据编程需要迅速查到相关的函数和变量。本书是帮助 Authorware 编程者迅速提高编程水平的案头必备的高级资料手册。

本书的配套光盘中提供了本书所有实例的源程序(包括 Authorware 6.0,6.5 和 7.0 三个版本)和部分素材,以方便读者学习。

本书可供所有在 Authorware 平台上工作的编程人员、教学人员使用,并可作为 Authorware 课程的教学参考书。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

Authorware 变量、函数、控件手册与范例/毕广吉编著. —北京:电子工业出版社,2003.9
ISBN 7-5053-9161-5

I. A… II. 毕… III. 多媒体-软件工具, Authorware IV. TP311.56

中国版本图书馆 CIP 数据核字(2003)第 083014 号

责任编辑:冉 哲 李 岩

印 刷:北京东光印刷厂

出版发行:电子工业出版社 <http://www.phei.com.cn>

北京市海淀区万寿路 173 信箱 邮编 100036

经 销:各地新华书店

开 本:787×1092 1/16 印张:30.75 字数:1030 千字 附光盘 1 张

版 次:2003 年 9 月第 1 版 2003 年 9 月第 1 次印刷

印 数:5000 册 定价:45.00 元 (含光盘)

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换;若书店售缺,请与本社发行部联系。联系电话:(010) 68279077。质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

前 言

Authorware之所以盛行于世,并不仅仅是因为它提供了“利用图标在流程线上编程”的这种适合于初学者的编程方式。如果只有初学者在某个软件平台上工作而不能吸引高级编程人员的话,那么其产品的水平往往不会很高,这种软件平台也就不可能得到广泛的应用。使 Authorware 广为流行的更主要的原因是, Authorware还同时提供了丰富的系统变量、系统函数以及对外部程序、外部函数(Xtras 和 U32)、外部控件(ActiveX)的支持。正是由于这一点,才使得初学者不会永远停留在初级水平上,使得高级编程人员有兴趣涉足 Authorware 的编程,从而设计出功能强大的 Authorware 应用软件,展现 Authorware 的魅力,反过来使 Authorware更加流行。

然而,所有打算进入 Authorware 高级编程层次的人中没有一个人不苦于缺少资料的。虽然市场上有关 Authorware 的书籍很多,但基本上都是处于初级阶段,主要介绍“利用图标在流程线上编程”的方式,其他方面的内容很少涉及,就连 Authorware 自身的系统变量和系统函数也不做详细的、实用性的介绍,最多只是简要地罗列一下而已。对于许多学过 Authorware 初级编程,想要进一步提高的人来说,查找 Authorware 高级资料成了一件苦事,本书作者也曾为此花费了很多的时间和精力。

本书是 Authorware 编程者必备的高级资料手册。作者总结了多年的教学经验和编程体会,精选了有关资料奉献给 Authorware 精英们。本书内容分为七个部分,分别对 Authorware 的系统变量、系统函数、U32 文件中的函数、Xtras 中的函数、Authorware 内部控件和 ActiveX 控件进行了详细的解释。为了使读者能更有效地使用本书,作者还精心制作了一个按用途分类的索引,并编写了用法示例程序。

以下数字概括了本书的主要内容:

- 235 种系统变量的详解
- 349 种系统函数的详解
- 56 个 U32 和 DLL 文件中 700 种函数的详解
- 7 个 Xtras 文件中 67 种函数的详解
- 20 种内部控件和 23 种 ActiveX 控件的详解
- 书中所有系统变量、系统函数、U32 函数和 Xtras 函数按用途分类索引(第 6 章)
- 说明变量、函数、控件用法的 149 个范例(适用于 Authorware 6.0, 6.5 和 7.0 三个版本)

本书的配套光盘中提供了本书所有实例的源程序(包括 Authorware 6.0, 6.5 和 7.0 三个版本)和部分素材,以方便读者学习。

希望本书能帮助读者迅速提高 Authorware 编程水平。由于作者水平所限,加之所掌握的资料有限,虽然对各函数的功能做了大量的验证工作,但书中必定有不少疏漏与谬误之处,请读者不吝赐教!

毕广吉

2003 年

目 录

第 1 章 系统变量	1
1.1 Authorware 中的变量	1
1.1.1 系统变量与用户自定义变量	1
1.1.2 变量窗口	2
1.1.3 基本变量类型	2
1.1.4 其他变量类型	3
1.2 Authorware 系统变量详解	5
第 2 章 系统函数、脚本函数与运算符	36
2.1 Authorware 中函数的分类	36
2.2 Authorware 系统函数详解	37
2.3 脚本函数	110
2.3.1 图标脚本函数	110
2.3.2 文件脚本函数	110
2.3.3 字符串脚本函数	110
2.4 Authorware 中的运算符	111
2.4.1 运算符详解	111
2.4.2 运算符的优先级	113
第 3 章 Authorware 使用的外部函数	115
3.1 UCD 函数的调用方法	115
3.1.1 UCD 函数及其种类	115
3.1.2 UCD 函数的声明与引用	115
3.1.3 设置查找路径	117
3.2 UCD 函数详解	117
3.2.1 A6dir.u32 详解	117
3.2.2 A6wmme.u32 详解	118
3.2.3 AnimGif.u32 详解	125
3.2.4 alTools.u32 详解	125
3.2.5 altoolTip.u32 详解	128
3.2.6 Anicur.u32 详解	132
3.2.7 Apwkeys.u32 详解	132
3.2.8 Apwmouse.u32 详解	133
3.2.9 Binapi.u32 详解	133
3.2.10 Budapi.u32 详解	138
3.2.11 Budkey.u32 详解	174
3.2.12 Budmenu.u32 详解	176
3.2.13 Budwav.u32 详解	181
3.2.14 Copyfile.u32 详解	183
3.2.15 Cover.u32 详解	183
3.2.16 CPUinfo.u32 详解	183
3.2.17 Currency.u32 详解	183
3.2.18 DDE.u32 详解	184

3.2.19	Dib.u32 详解	185
3.2.20	Dib2pal.u32 详解	185
3.2.21	Dirdlg.u32 详解	185
3.2.22	Disptext.u32 详解	185
3.2.23	Edit.u32 详解	186
3.2.24	Filedgs.u32 详解	187
3.2.25	Ftp.u32 详解	188
3.2.26	GWEyes.u32 详解	190
3.2.27	HListBox.u32 详解	190
3.2.28	InstanceControl.u32 详解	192
3.2.29	Jfwapi.dll 详解	192
3.2.30	Jpeg.u32 详解	192
3.2.31	Listbox.u32 详解	193
3.2.32	Memtools.u32 详解	193
3.2.33	MIDILOOP.u32 详解	196
3.2.34	MInputBox.u32 详解	196
3.2.35	mp3player.u32 详解	196
3.2.36	MSAgent2.u32 详解	197
3.2.37	Objsamp.u32 详解	201
3.2.38	Odbc.u32 详解	202
3.2.39	Palette.u32 详解	202
3.2.40	Prnt.u32 详解	202
3.2.41	Revstr.u32 详解	204
3.2.42	RGBcover.u32 详解	205
3.2.43	RTFObj.u32 详解	205
3.2.44	ScrlEdit.u32 详解	207
3.2.45	Simple.u32 详解	207
3.2.46	sInstall.u32 详解	207
3.2.47	Smtip.u32 详解	208
3.2.48	SpeakText.u32 详解	208
3.2.49	SyncSound.u32 详解	209
3.2.50	tMsClipboard.u32 详解	210
3.2.51	tMsControls.u32 详解	210
3.2.52	tMsDSN.u32 详解	213
3.2.53	tMsTextBox.u32 详解	214
3.2.54	TMSystem.u32 详解	215
3.2.55	Winapi.u32 详解	232
3.2.56	WinCtrls.u32 详解	243
第 4 章	Authorware 中的 Xtras	246
4.1	Xtras 的概念和分类	246
4.2	调用 Scripting Xtras	248
4.3	常用 Scripting Xtras 函数详解	249
4.3.1	ActiveX.X32 详解	249
4.3.2	Budapi.X32 详解	250
4.3.3	BudZip.X32 详解	250
4.3.4	Budunzip.X32 详解	250
4.3.5	fileIO.X32 详解	251

4.3.6	mPrint. X32 详解	254
4.3.7	secure. X32 详解	257
第 5 章	在 Authorware 中使用控件	259
5.1	Authorware 的内部控件	259
5.1.1	内部控件列表	259
5.1.2	内部控件属性详解	261
5.2	ActiveX 控件	268
5.2.1	ActiveX 控件的安装与注册	268
5.2.2	ActiveX 控件的用法	271
5.2.3	ActiveX 控件的属性对话框	271
5.3	常用 ActiveX 控件详解	275
5.3.1	AnimateXControl 控件	275
5.3.2	Calendar Control 8.0 控件	276
5.3.3	DisablePro 控件	277
5.3.4	Microsoft Animation Control 6.0(SP4)控件	277
5.3.5	Microsoft Common Dialog Control, Version 6.0 控件	278
5.3.6	Microsoft Date and Time Picker Control 6.0(SP4)控件	279
5.3.7	Microsoft Forms 2.0 CheckBox 控件	280
5.3.8	Microsoft Forms 2.0 ComboBox 控件	281
5.3.9	Microsoft Forms 2.0 CommandButton 控件	283
5.3.10	Microsoft Forms 2.0 Label 控件	284
5.3.11	Microsoft Forms 2.0 ListBox 控件	285
5.3.12	Microsoft Forms 2.0 OptionButton 控件	285
5.3.13	Microsoft Forms 2.0 ScrollBar 控件	286
5.3.14	Microsoft Forms 2.0 TextBox 控件	287
5.3.15	Microsoft MonthView Control 6.0(SP4)控件	288
5.3.16	Microsoft Multimedia Control, Version 6.0 控件	289
5.3.17	Microsoft Slider Control 6.0(SP4)控件	291
5.3.18	Microsoft UpDown Control 6.0(SP4)控件	292
5.3.19	MIDI1.midiPlayer 控件	293
5.3.20	RealPlayer G2 Control 控件	293
5.3.21	SoundDevil 控件	294
5.3.22	TrayIcon 控件	294
5.3.23	Windows Media Player 控件	295
第 6 章	系统变量、系统函数、U32 函数及 Xtras 函数按功能用途索引	296
6.1	常数、变量与函数操作	296
6.2	数学函数	297
6.3	字符及字符串处理	298
6.4	查找与替换	298
6.5	文本处理与显示	299
6.6	字体	299
6.7	提示工具、信息框和对话框	300
6.8	位图	301
6.9	屏幕显示控制	301
6.10	用户界面	302
6.11	绘图	304
6.12	文件与目录操作	304

6.13	声音播放	306
6.14	动画(数字电影)播放控制	308
6.15	视频播放	308
6.16	程序运行与编程	308
6.17	图标操作	310
6.18	一般图标操作	310
6.19	交互图标和各种响应	311
6.20	几种具体的图标	313
6.21	用户与权限	315
6.22	对硬件的操作	315
6.23	取得系统信息	317
6.24	设置系统	318
6.25	注册表	319
6.26	日期与时间	320
6.27	窗口控制	321
6.28	打印	322
6.29	网络	322
6.30	其他函数	324
第7章	Authorware 系统变量、系统函数、U32 函数、Xtras 函数和控件用法范例	325
7.1	AnimateX 控件示例.a6p	325
7.2	Budkey 综合验证.a6p	326
7.3	Calendar 日历控件使用.a6p	331
7.4	Flash 动画播放速度控制.a6p	333
7.5	Flash 动画缩放控制.a6p	334
7.6	Flash 动画旋转控制.a6p	335
7.7	MonthView 控件 60 日历.a6p	336
7.8	MP3 播放器.a6p	337
7.9	MSAgent 用法.a6p	338
7.10	Real 播放器.a6p	340
7.11	tMsControls 用法.a6p	341
7.12	UpDown60 控件用法.a6p	343
7.13	Winapi 测试文件.a6p	344
7.14	Winapi 窗口闪烁.a6p	345
7.15	Winapi 关闭窗口.a6p	346
7.16	Winapi 绘图.a6p	346
7.17	Winapi 绘椭圆.a6p	347
7.18	Winapi 鼠标设置.a6p	348
7.19	Winapi 填充.a6p	348
7.20	Winapi 文本输出.a6p	350
7.21	播放 Director 影片.a6p	351
7.22	播放 Gif 动画.a6p	351
7.23	彩色遮盖桌面.a6p	353
7.24	操纵鼠标.a6p	353
7.25	操作声卡.a6p	354
7.26	测定 CPU.a6p	357
7.27	查找 CD.a6p	358
7.28	查找程序名.a6p	359

7.29	查找窗口调用外部程序.a6p	360
7.30	创建并读写文件.a6p	361
7.31	创建程序组.a6p	362
7.32	创建和删除快捷方式.a6p	363
7.33	创建临时文件.a6p	364
7.34	创建特殊键.a6p	365
7.35	打开和保存文件对话框.a6p	366
7.36	打印.a6p	366
7.37	带滚动条的文本编辑框.a6p	367
7.38	单选钮互锁方法.a6p	368
7.39	弹出式菜单.a6p	369
7.40	弹簧振子.a6p	369
7.41	调色器 1.a6p	371
7.42	调色器 2.a6p	372
7.43	调用外部程序.a6p	372
7.44	动画光标.a6p	373
7.45	动态显示图片.a6p	373
7.46	对象变色移动.a6p	374
7.47	复选框控件用法.a6p	375
7.48	覆盖桌面.a6p	375
7.49	改变按钮大小.a6p	376
7.50	改变按钮和显示对象的位置.a6p	376
7.51	改变屏幕分辨率.a6p	377
7.52	改变图层.a6p	378
7.53	改变音量.a6p	379
7.54	格式化输出.a6p	380
7.55	各种日期时间函数.a6p	380
7.56	关闭多项 Windows 设置.a6p	381
7.57	滚动文本框做目录.a6p	383
7.58	画板.a6p	384
7.59	绘制李萨如图形.a6p	385
7.60	获取 Windows 程序组信息.a6p	386
7.61	获取键表.a6p	387
7.62	计算器.a6p	388
7.63	剪贴板操作.a6p	392
7.64	建立“开始”菜单程序组.a6p	392
7.65	建立可折叠树状索引.a6p	393
7.66	建立选项卡组.a6p	394
7.67	奖号发生器.a6p	394
7.68	禁止再次运行程序.a6p	395
7.69	复制文件.a6p	396
7.70	级联菜单.a6p	397
7.71	两天的距离.a6p	400
7.72	列表框 HListBox 用法.a6p	401
7.73	列表框及其排序.a6p	402
7.74	媒体播放器.a6p	402
7.75	模板输入.a6p	403

7.76	判断对象之间的覆盖关系. a6p	404
7.77	判断是否框住一个对象的方法. a6p	405
7.78	气泡提示. a6p	406
7.79	区别鼠标的单击、双击、右击、进入. a6p	408
7.80	曲柄和活塞. a6p	409
7.81	驱动器分析. a6p	410
7.82	驱动器类型. a6p	411
7.83	取出 WinCtrls 的控件和属性. a6p	412
7.84	取出已安装的 ODBC 驱动程序名. a6p	413
7.85	取得屏幕分辨率和进程. a6p	414
7.86	取得文件信息. a6p	414
7.87	任意控制动画播放 AVI. a6p	416
7.88	任意控制动画播放 Cel. a6p	416
7.89	设置 Windows 背景图案. a6p	417
7.90	设置屏幕. a6p	419
7.91	使用 Animation60 控件. a6p	420
7.92	使用 Binapi. a6p	421
7.93	使用 Checked 系统变量和制作面板的方法. a6p	421
7.94	使用 ComboBox 控件. a6p	423
7.95	使用 MInputBoxu32. a6p	423
7.96	使用 ScrollBar 控件. a6p	424
7.97	使用 TextBox 控件输入文本. a6p	424
7.98	使用函数播放 CD 声音. a6p	425
7.99	使用函数播放 MIDI 声音. a6p	427
7.100	使用函数播放 WAV 声音. a6p	429
7.101	事件响应的用法. a6p	430
7.102	输入密码. a6p	431
7.103	鼠标左键与右键. a6p	432
7.104	数据绑定. a6p	433
7.105	数据库示例. a6p	433
7.106	通用对话框. a6p	435
7.107	统一图片大小使适合 VCD. a6p	436
7.108	透镜成像. a6p	439
7.109	图形菜单. a6p	441
7.110	为演示窗口添加最大化最小化按钮. a6p	443
7.111	文件系统控件的使用. a6p	443
7.112	文件属性操作. a6p	445
7.113	系统变量 Within 用法. a6p	446
7.114	下拉菜单的操作. a6p	447
7.115	显示 WinCtrls 各种控件. a6p	450
7.116	显示文本. a6p	451
7.117	新建、复制、删除文件、目录并操作属性. a6p	452
7.118	选择并打开文件. a6p	454
7.119	延时. a6p	454
7.120	演示窗口的句柄对比 1. a6p	455
7.121	演示窗口的句柄对比 2. a6p	455
7.122	验证 ExecutingIconID 变量和 ExecutingIconTitle 变量. a6p	455

7.123	移图游戏.a6p	456
7.124	隐藏任务栏.a6p	458
7.125	用 DateTimePicker 控件做下拉日历.a6p	459
7.126	用 Dib 函数显示位图.a6p	459
7.127	用 Edit 编的记事本(按钮).a6p	460
7.128	用 Edit 编的记事本(菜单).a6p	461
7.129	用 Edit 和 fileIO 编辑文本.a6p	462
7.130	用 MCI 播放声音.a6p	463
7.131	用 sInstall 复制文件.a6p	463
7.132	用 SMTP 发送 E-mail 的例子.a6p	464
7.133	用 SoundDevil 控件同时播放 MIDI 和 WAV 声音.a6p	465
7.134	用 Strip 函数滤出字符.a6p	466
7.135	用 SyncSound 同时播放 4 路声音.a6p	466
7.136	用滑动条控制电影最简单方法.a6p	467
7.137	用极坐标绘制曲线.a6p	468
7.138	指针钟.a6p	469
7.139	抓图.a6p	470
7.140	捉球.a6p	471
7.141	字符串反向.a6p	472
7.142	自由落体.a6p	473
7.143	走马灯显示文本.a6p	473
7.144	最简单的 UCD.a6p	474
7.145	图标脚本函数.a6p	475
7.146	利用图标脚本函数绘图.a6p	475
7.147	文件脚本函数.a6p	475
7.148	字符串脚本函数.a6p	476
7.149	文件中的多个字符串脚本函数.a6p	476

第1章 系统变量

1.1 Authorware 中的变量

1.1.1 系统变量与用户自定义变量

在 Authorware 中变量分为系统变量与用户自定义变量两大类。

1. 系统变量

在 Authorware 中共提供了 229 个系统变量,用于记录 Authorware 运行过程中图标、对象的属性以及响应关系、程序运行状态等。用户可以在程序中读取系统变量的值,系统变量的值一般由 Authorware 在运行的过程中自动设置,有些系统变量的值也允许用户在程序中设置。

系统变量名是 Authorware 规定的,用户不能更改,并且 Authorware 对系统变量大小写有具体的规定,无论用户输入系统变量名时的大小写如何,Authorware 都会把它变换成规定的样式。

有些系统变量的作用域是作用于整个 Authorware 程序的,称为全局系统变量,例如系统变量 ClickX, ClickY 等。相当多的系统变量的作用域与某个具体的图标相关联,称为图标变量。图标变量的引用方法是在系统变量名的右边使用“@”字符指定所关联的图标名,例如系统变量 Movable 的值决定了显示对象的移动性,若要使名为“显示图形”的显示图标中的内容不能为用户所移动,就应令该图标的 Movable 值为 FALSE,具体做法是:

Movable@“显示图形”:= FALSE

如果省略了“@”字符和指定的关联图标名,写成

Movable:= FALSE

的样子,则该系统变量将与当前图标相关联;如果以上语句是写在附加计算图标中的,则该系统变量将与附加计算图标所依附的图标相关联。

除了在计算图标中使用系统变量之外,在 Authorware 各种属性对话框中也能使用系统变量以及由系统变量、用户变量和系统函数组成的表达式,只有少数属性对话框中的部分属性参数域除外。

2. 用户自定义变量

Authorware 允许用户使用自定义变量。自定义变量名称(变量名)的第一个字符必须是英文字母或下划线,其余字符可以是英文字母、数字或下划线,还允许在变量名中使用空格。变量名最多可以含有 40 个字符。Authorware 对变量名是不区分大小写的,但习惯上为了与系统变量相区别,采用首字母小写、其余每一个单词首字母大写的的方式。需要注意的是,不要使用 Authorware 保留字作为自定义变量名,以免引起混淆。

自定义变量也允许像系统变量一样,使用“@”字符与一个图标建立关联,从而把自定义变量指定为自定义图标变量。相对于这种情况而言,不使用“@”字符与图标相关联的自定义变量称为全局自定义变量。

用户自定义变量可以随着使用、随着定义,即新变量名第一次在计算图标代码窗口或图标属性对话框等场合出现时就意味着定义了一个新变量,Authorware 此时会弹出一个如图 1-1 所示的“新变量”对话框。

在图 1-1 所示的对话框中,Name 文本框中是变量名,在 Initial value 文本框中允许为新变量输入一个初

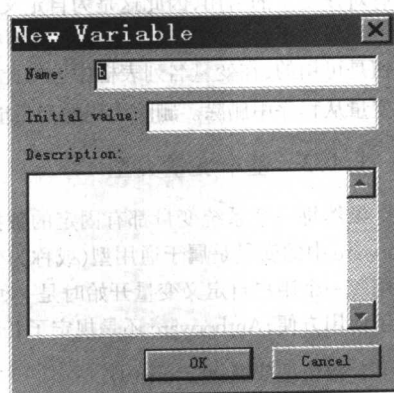


图 1-1 “新变量”对话框

始值,而在 Description 文本框中可以输入一些对该变量的描述性文字,以利于对程序的理解,完成上述设置后单击 OK 按钮确认并退出该对话框。

另一种定义新变量的方法是在变量窗口中进行的,详细方法见下节。

1.1.2 变量窗口

变量窗口是 Authorware 集中管理变量的地方。利用菜单命令 Window → Variables 或直接单击工具栏上的

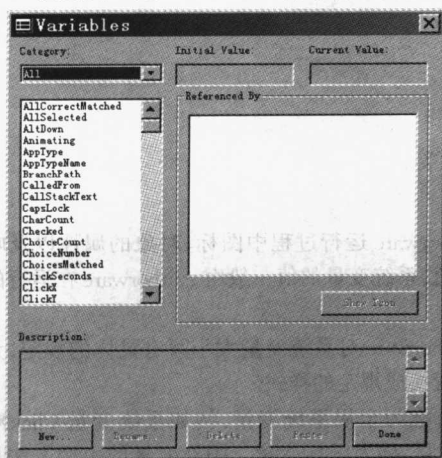


图 1-2 变量窗口

的按钮就可以打开变量窗口,如图 1-2 所示。

Category 下拉列表框中列出了系统变量的分类列表,用户可以从中选择一种系统变量类型,此时在其下方的变量名列表框中将列出所有属于此类的系统变量。Category 下拉列表框中的最后一项是当前程序名,选中该项后,变量名列表框中将显示在该程序中已定义的所有自定义变量和所有已装入的 UCD 函数名。

在变量名列表框中选定一个变量名后,就会在 Initial Value 域中显示该变量的初始值,在 Current Value 域中显示该变量的当前值。在 Referenced By 域中显示哪些图标引用了该变量,如果在这里选中了一个图标名,Show Icon 按钮将变为可用,单击此按钮,Authorware 就会在设计窗口中加强显示出此图标所在的位置,所有这些信息对于分析调试程序有很大的参考价值。在 Description 域中显示对该变量的描述性文字,这些文字对于帮助用户理解变量的意义和用法都有很大的帮助。

如果程序设计正处于文本输入状态(例如在计算图标窗口中、显示图标处于文本工具的输入状态时等),则 Paste 按钮变为可用。单击此按钮,就会将该变量名粘贴插入到文本中。如果光标在显示图标文本工具的输入域中,还会自动加上一对大括号 {}。

New 按钮提供了定义新变量的途径,单击此按钮会弹出图 1-1 所示的对话框,可以进行新变量的定义。

在变量名列表框中选中一个自定义变量后,Rename 按钮变为可用,单击此按钮弹出如图 1-3 所示的变量重命名对话框,在这里对变量重新命名后单击 OK 按钮即可。注意,在此处将变量重命名后,系统会同时改变程序中所有对该变量的引用,因此这是为自定义变量重命名的惟一正确方法。

如果一个自定义变量没有被任何图标所引用,则变量窗口中的 Delete 按钮是可用的,在变量名列表框中选中此变量,然后单击 Delete 按钮就可将该变量从程序中删除。删除无用的变量能优化程序,提高程序的运行效率。

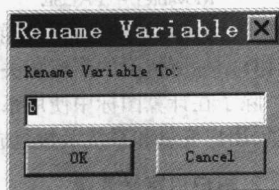


图 1-3 变量重命名对话框

1.1.3 基本变量类型

虽然每一个系统变量都有固定的数据类型,用户自定义变量却可以被定义成不同的类型。实际上 Authorware 中的变量是属于通用型(或称为变体型)的,也就是说各种变量类型没有严格的区别,并且能够彼此通用。一个用户自定义变量开始时是一种变量类型,以后还可以多次为该变量赋予其他类型的数据。不过为了使用方便,Authorware 还是规定了几种数据类型。

1. 数值型变量(Numerical)

数值型变量用于保存数值,数值可以是整型也可以是浮点型。

作为整型使用时,数值型变量能表示的数据范围是 -2147483648~2147483647,超过这个范围就要出错。例如,-2147483649 会变为 2147483647,2147483648 会变为 -2147483648 等,而且 Authorware 对此根本不报告任何错误信息。

作为浮点型使用时,Authorware 不支持科学记数法(例如:2.5E12),最多允许 15 位有效数字,小数部分

最多允许保留 10 位小数。

对比整型与浮点型两种用法,使用浮点型能够表示更大的整数,例如给变量 x 赋一个 15 位的整数值:

```
x:= 999999999999999
```

会出错变成 -1530494977,而若给变量 x 赋一个 15 位的浮点数:

```
x:= 999999999999999.0
```

却能得到正确的结果。

算术运算符(+, -, *, /, **)是专为数值型变量设计的,Authorware 在对两个量进行运算时就总把两个运算数当做数值型的数据处理,例如:

```
y1:= 100           --数值型
y2:= "2abc0xyz"    --字符型
y3:= TRUE          --逻辑型
y4:= y1 + y2 + y3   --三种类型的数据进行算术运算
```

其结果为 121。

此类运算的规律是,字符串先经过 Number 函数的处理,逻辑值 TRUE 视为 1,FALSE 视为 0。当然这种用法是不值得提倡的。

2. 字符型变量(Character)

字符串变量用于存储字符,最多允许有 30 000 个字符。

在一个字符串常量中,字符“\”作为转义字符来使用:“\r”表示一个回车字符 Return;“\t”表示一个制表字符 Tab;“\”表示一个半角的引号“'”,用于在字符串内包含一个引号;另外,由于“\”字符已作为转义字符来使用,要在字符串中使用一个“\”字符必须连续写两个“\”字符,即“\\”。例如,下例返回 userName = “John”:

```
x:= "userName="^Char(34)^"John"^Char(34)
```

下例返回 c:\Tutorial\Firstaid.a6p:

```
"c:\\Tutorial\\Firstaid.a6p"
```

而

```
"Date \t Time \t Temperature \r 4/21 \t 06:03 \t 63F"
```

则表示:

```
Date Time Temperature
4/21 06:03 63F
```

字符运算只有一种:“^”。Authorware 把“^”运算符连接的数据解释成字符串,数值变量被直接转换为字符,逻辑值 TRUE 被转换为“1”,逻辑值 FALSE 被转换为“0”。例如:

```
y1:= 100           --数值型
y2:= "ABC"         --字符型
y3:= TRUE          --逻辑型
y4:= FALSE         --逻辑型
y5:= y1 ^ y2 ^ y3 ^ y4 --三种类型的数据进行字符串运算,其结果为"100ABC10"
```

3. 逻辑型变量(Logical)

逻辑型变量只有两种取值:TRUE 和 FALSE。在 Authorware 中凡是在应该出现逻辑值的地方出现的表达式,都将被转换为逻辑值处理,用逻辑运算符“&”,“|”,“~”连接的表达式中各运算数据也都被转换为逻辑值处理。转换的方法是:数值 0 视为 FALSE,非 0 数值一律视为 TRUE;字符串(不区分大小写)“TRUE”,“T”,“Yes”,“On”视为 TRUE,其他所有字符串均视为 FALSE。但是如果在显示图标中使用大括号{}中放入一个逻辑表达式的方法来显示逻辑值,却只能显示 1 或 0,而不能显示 TRUE 或 FALSE。

1.1.4 其他变量类型

1. 列表变量(List)

列表变量类似于其他编程语言中的数组,用于存储常数或变量的矢量和矩阵。列表变量用方括号“[]”

括起来,其中的各个元素用逗号分隔。Authorware 中有两种列表变量:线性列表和属性列表。

(1) 线性列表(Linear lists)

线性列表的各元素都是一个具体的值,这些值可以是常量或变量,每一个元素在线性列表中的位置称为索引号 Index,例如:

```
[1,2,"a","b",TRUE]
```

定义一个线性列表有两种方法。方法一是先定义一个空的线性列表变量,然后用赋值符依次为列表中的各个元素赋值,示例如下:

```
userList[1] := FirstName  
userList[2] := mInitial  
userList[3] := LastName  
userList[4] := userID  
userList[5] := password  
userList[6] := Date
```

方法二是先定义一个空的线性列表变量,然后用 AddLinear 函数为列表中的各个元素赋值,示例如下:

```
--先定义一个空的列表变量  
userList := [ ]  
--利用 AddLinear 函数为列表中的各个元素赋值  
AddLinear(userList,FirstName)  
AddLinear(userList,mInitial)  
AddLinear(userList,LastName)  
AddLinear(userList,userID)  
AddLinear(userList,password)  
AddLinear(userList,Date)
```

从线性列表中取出元素值的方法也有两种,方法一是像引用数组一样直接利用索引号引用,例如:

```
tempVar := userList[1]
```

方法二是使用 ValueAtIndex 函数,例如:

```
tempVar := ValueAtIndex(userList,1)
```

利用系统函数 SortByValue 能够对线性列表进行排序,详见 2.2 节。

(2) 属性列表(Property lists)

属性列表的各元素由属性和一个数值组成,属性由“#”字符打头,属性和数值之间用冒号分隔,元素之间用逗号分隔。下例就是一个属性列表变量:

```
[# a:"John",# b:"L",# c:"Sullivan",# d:"jls391",# e:"hedgehog",# f:"8/5/97"]
```

定义一个属性列表有两种方法。方法一是先定义一个空的属性列表变量,然后用赋值符“:=”依次为列表中的各个属性赋值,示例如下:

```
--定义一个空的属性列表变量  
propList := [:]  
--为列表中的各个属性赋值  
propList[# a] := "John"  
propList[# b] := "L"  
propList[# c] := "Sullivan"  
propList[# d] := "jls391"  
propList[# e] := "hedgehog"  
propList[# f] := Date
```

方法二是先定义一个空的属性列表变量,然后使用 AddProperty 函数依次为列表中的各个属性赋值,示例如下:

```
--定义一个空的属性列表变量  
propList := [:]  
--用 AddProperty 函数依次为列表中的各个属性赋值
```

```
AddProperty(propList, # a, "John")
AddProperty(propList, # b, "L")
AddProperty(propList, # c, "Sullivan")
AddProperty(propList, # d, "jls391")
AddProperty(propList, # e, "hedgehog")
AddProperty(propList, # f, Date)
```

从属性列表中取出元素值的方法也有两种,方法一是直接利用属性引用,例如:

```
tempVar: = propList[# a]
```

方法二是使用 PropertyAtIndex 函数,例如:

```
tempVar: = PropertyAtIndex(propList, # a)
```

利用系统函数 SortByProperty 和 SortByValue 能够对属性列表进行排序,详见 2.2 节。

2. 符号变量(Symbol)

符号变量是一种以“#”字符开头的字符变量,紧接“#”字符的第一个字符只能是英文字母或下划线,不能是数字。Authorware 处理符号变量的速度比字符变量更快,符号变量也能像字符变量一样进行运算,以下是符号变量运算的一个例子:

--定义三个符号变量

```
y1: = # a100
```

```
y2: = # s2aaaaaaaa2
```

```
y3: = # TRUE
```

--符号变量的运算

```
y4: = y1 + y2 + y3 --结果为 122
```

```
y5 = y1 ^ y2 ^ y3 --结果为 # a100 # s2aaaaaaaa2 # TRUE
```

```
y6 = y1 | y2 | y3 --结果为 1
```

3. 点变量(Point)

点变量用于指定屏幕上的一个位置,利用点变量也可以进一步创建矩形变量,或判断点与矩形的关系。点变量由系统函数 Point 定义,例如:

```
myPoint: = Point(100,100)
```

在(100,100)处建立了一个点变量 myPoint。详见对系统函数 Point 的解释。

4. 矩形变量(Rect)

矩形变量用于指定屏幕上的一个矩形区域,对于图形处理、判定矩形与矩形之间、矩形与点之间的相对位置等有实际用处。定义矩形变量的方法是使用系统函数 Rect,例如:

```
myRect: = rect(0,0,100,100)
```

和

```
point1: = Point(0,0)
```

```
point2: = Point(100,100)
```

```
rect(point1,point2)
```

都建立了一个矩形变量 myRect,其左上角是(0,0),右下角是(100,100),详见对系统函数 Rect 的解释。

1.2 Authorware 系统变量详解

本节详细介绍 Authorware 所有系统变量,为了方便查阅,本节按字母顺序进行介绍。若想知道实现某一特定功能有哪一个或哪几个系统变量,请查阅第 6 章。

每一个系统变量的解释由【类型】、【格式】、【说明】三部分组成。其中【类型】包括四项:第一项是变量的数据类型,第二项是变量在 Authorware 变量窗口中所属的分类(若想知道某一分类包含哪些系统变量,请查阅 6.32 节),第三项给出变量是只读的(ReadOnly)还是可读写的(ReadWrite),第四项给出变量是全局变量(Public)还是图标变量(Icon)。【格式】中,由方括号“[]”括起来的@“IconTitle”为可省略的内容,省略时表示当前图标。有些系统变量可以单独使用(针对当前图标),也可以作为图标关联变量使用(通过 IconTitle 指定)。

AllCorrectMatched 系统变量

【类型】 数据类型 Logical, 分类 Interaction, 读写 ReadOnly, 作用域 Icon

【格式】 AllCorrectMatched["@IconTitle"]

【说明】 如果当前(或 IconTitle 指定的)交互图标分支结构中所有标为 Correct 响应状态的分支都已经匹配, 则该变量的值为 TRUE; 否则为 FALSE。

AllSelected 系统变量

【类型】 数据类型 Logical, 分类 Decision, 读写 ReadOnly, 作用域 Icon

【格式】 AllSelected["@IconTitle"]

【说明】 如果属于当前(或 IconTitle 指定的)决策图标的所有分支图标都已被使用, 则该变量的值为 TRUE; 否则为 FALSE。

AltDown 系统变量

【类型】 数据类型 Logical, 分类 General, 读写 ReadOnly, 作用域 Public

【格式】 AltDown

【说明】 按下 Alt 键时, 该变量的值为 TRUE。但在编辑状态下, 按下 Alt 键会激活 Authorware 系统菜单。

Animating 系统变量

【类型】 数据类型 Logical, 分类 Icons, 读写 ReadOnly, 作用域 Icon

【格式】 Animating["@IconTitle"]

【说明】 若当前(或 IconTitle 指定的)图标中的显示对象正在移动, 则该变量的值为 TRUE; 否则为 FALSE。

【示例】 在程序“移图游戏.a6p”中使用了 Animating 变量。

AppType 系统变量

【类型】 数据类型 Numerical, 分类 General, 读写 ReadOnly, 作用域 Public

【格式】 AppType

【说明】 该变量存储的是打包后的文件所包含的 Runtime 文件类型, 或者未打包的 Authorware 应用程序的类型。该变量有四种值:

1——文件打包时, Runtime 文件类型为 Windows 3.1 下的 16 位文件。

2——文件打包时, Runtime 文件类型为 32 位 (Windows 95/NT) 文件; 文件不打包时, 为 Authorware 的应用程序。

3——文件打包时, Runtime 文件类型为 Macintosh 文件。

4——文件打包时, Runtime 文件类型为 Power Macintosh 文件。

有些应用程序需要知道运行时的文件类型(例如是否已经打包), 此时该函数很有用。

【示例】 在程序“查找 CD.a6p”中有使用 AppType 变量的例子。

AppTypeName 系统变量

【类型】 数据类型 Character, 分类 General, 读写 ReadOnly, 作用域 Public

【格式】 AppTypeName

【说明】 该变量存储的是打包后的文件所包含的 Runtime 文件类型名, 或者未打包的 Authorware 应用程序的类型名。该变量有四种值:

16-bit——文件打包时, Runtime 文件类型为 Windows 3.1 下的 16 位文件。

32-bit——文件打包时, Runtime 文件类型为 32 位 (Windows 95/NT) 文件; 文件不打包时, 为 Authorware 的应用程序。

68K——文件打包时, Runtime 文件类型为 Macintosh 文件, 或者 Authorware 应用程序为 Macintosh 文件。

PowerPC——文件打包时, Runtime 文件类型为 Power Macintosh 文件, 或者 Authorware 应用程序为 Power Macintosh 文件。

BranchPath 系统变量

【类型】 数据类型 Numerical, 分类 Interaction, 读写 ReadWrite, 作用域 Icon

【格式】 BranchPath["@Icon Title"]

【说明】 该变量存储的是当前(或 IconTitle 指定的)交互分支中,用户匹配的最后一个响应所对应的分支类型的数字标识,各数字含义为:0——Continue(继续);1——Exit Interaction(退出交互);2——Try Again(重试);3——Return(返回)。

CalledFrom 系统变量

【类型】 数据类型 Numerical,分类 Icons,读写 ReadOnly,作用域 Icon

【格式】 CalledFrom["@IconTitle"]

【说明】 该变量一般作为图标关联变量使用,存储的是最近调用程序的图标的 ID 标识。

CallStackText 系统变量

【类型】 数据类型 Numerical,分类 Icons,读写 ReadOnly,作用域 Public

【格式】 CallStackText

【说明】 该变量用于存储一个列表,其中列出了图标间的调用与被调用关系信息。

CapsLock 系统变量

【类型】 数据类型 Logical,分类 General,读写 ReadOnly,作用域 Public

【格式】 CapsLock

【说明】 按下 Caps Lock 键时,该变量值为 TRUE;否则为 FALSE。

CharCount 系统变量

【类型】 数据类型 Numerical,分类 Interaction,读写 ReadOnly,作用域 Icon

【格式】 CharCount["@IconTitle"]

【说明】 该变量存储的是当前(或 IconTitle 指定的)图标中,用户在文本输入响应中输入的文本所包含的字符个数。汉字也像英文字母一样被当做一个字符看待。

Checked 系统变量

【类型】 数据类型 Logical,分类 Interaction,读写 ReadWrite,作用域 Icon

【格式】 Checked["@ButtonIconTitle"]

【说明】 该变量一般作为图标关联变量使用。如果按钮是按钮编辑器对话框中所列的四种 Checked(选中)状态中的任何一种,则该变量的值都为 TRUE。如果设置其值为 TRUE,相当于自动选中了该按钮。

【示例】 在程序“单选钮互锁方法.a6p”和程序“使用 Checked 系统变量和制作面板的方法.a6p”中有使用 Checked 变量的例子。

ChoiceCount 系统变量

【类型】 数据类型 Numerical,分类 Interaction,读写 ReadOnly,作用域 Icon

【格式】 ChoiceCount["@IconTitle"]

【说明】 该变量存储的是当前(或 IconTitle 指定的)交互图标分支结构中所有可能的响应数目。

ChoiceNumber 系统变量

【类型】 数据类型 Numerical,分类 Interaction,读写 ReadOnly,作用域 Icon

【格式】 ChoiceNumber["@IconTitle"]

【说明】 该变量存储的是当前(或 IconTitle 指定的)交互图标分支结构所有分支中,与用户输入相匹配的分支路径号。相应的图标从左到右,由 1 开始编号。

ChoicesMatched 系统变量

【类型】 数据类型 Numerical,分类 Interaction,读写 ReadOnly,作用域 Icon

【格式】 ChoicesMatched["@IconTitle"]

【说明】 该变量存储的是在当前(或 IconTitle 指定的)交互图标分支结构中,用户已匹配的分支路径的数目。

ClickSeconds 系统变量

【类型】 数据类型 Numerical,分类 Time,读写 ReadOnly,作用域 Public