

Windows API 函数参考手册

本书编写组 编著

人民邮电出版社

图书在版编目 (C I P) 数据

Windows API 函数参考手册 / 本书编写组编著 . —北京 : 人民邮电出版社 , 2002.10

ISBN 7 - 115 - 10612 - 6

. W... . 本... . 窗口软件 , Windows - 软件接口 - 技术手册 IV . TP316.7-62

中国版本图书馆 CIP 数据核字 (2002) 第 071412 号

内容简介

本书是关于 Microsoft Win32 API 函数的完整参考手册。书中详细介绍了每一个 Win32 API 函数的使用方法, 其中着重介绍其原型表示、函数说明、参数说明、注意事项、相应头文件 (.H)、链接库 (.LIB 或 .DLL) 和典型示例。对于每个函数本书给出了相应的示例部分, 这样读者在阅读相应的 Win32 API 函数时就能很快地了解它的具体功能和使用方法, 便于更快地掌握该接口函数。

本书是从事 Microsoft Windows 操作系统开发和应用人员的必备参考书, 也可作为大专院校相关专业师生自学、教学参考用书。

Windows API 函数参考手册

- ◆ 编 著 本书编写组
责任编辑 张立科
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
读者热线 010-67180876
北京汉魂图文设计有限公司制作
北京 印刷厂印刷
新华书店总店北京发行所经销
- ◆ 开本 : 787×1092 1/16
印张 : 48.25
字数 : 1495 千字 2002 年 10 月第 1 版
印数 : 1 - 0 000 册 2002 年 10 月北京第 1 次印刷

ISBN 7-115-10612-6/TP · 3074

定价 : 82.00 元 (附光盘)

本书如有印装质量问题, 请与本社联系 电话 : (010) 67129223

前 言

Microsoft Win32 API 是 Microsoft Windows 操作系统提供的应用程序接口函数,它基本上概括了应用程序编程中涉及的每个方面。我们在使用 MFC 编程时,所用的 MFC 函数都是封装在 Win32 API 基础之上的,即 MFC 是 Win32 API 的一个上层架构。学习 Win32 API 能更深刻的理解 MFC 的运作机制,同时使用 Win32 API 函数编程要比使用 MFC 更灵活,能编写出更加高效的程序。Microsoft Win32 API 也可以在 VB 和 Delphi 等语言中使用。

Microsoft Win32 API 函数按照功能划分为窗口处理、设备上下文——DC、绘图函数、位图和图标、菜单处理、文件处理、同步、处理文本和字体、硬件和系统、Windows 消息、进程和线程。

本书按功能对 Win32 API 函数进行了划分,对于每一个功能部分的函数都具体给出了它们的函数原型、功能和用法、注意事项、所在头文件、参数调用和典型示例等信息。本书内容如下:

第 1 章介绍了 Win32 应用程序的一些基本知识,例如系统消息传递机制,图形接口编程等,同时,它也介绍了 Win32 API 函数的一些基本知识。

第 2 章介绍了关于窗口处理部分的 Win32 API 函数,例如窗口创建,撤销等。

第 3 章介绍了设备上下文——DC,例如创建和获取 DC、坐标变换等。

第 4 章介绍了绘图函数,例如路径、图元文件等。

第 5 章介绍了位图和图标,例如从 DLL 中获取图标,设置 DIB 的颜色表等。

第 6 章介绍了菜单的处理,例如上下文菜单显示,设置位图菜单项等。

第 7 章介绍了文件的处理,例如文件查找,加锁和解锁文件等。

第 8 章介绍了同步,例如互斥对象,信号量对象和关键段对象等。

第 9 章介绍了文本和字体的处理,例如获取系统中字体的信息,输出各种形式的文本等。

第 10 章介绍了硬件和系统,例如捕获鼠标,获取各种系统信息等。

第 11 章介绍了 Windows 消息,例如消息的获取和发送等。

第 12 章介绍了进程和线程,例如调试进程和线程,获取和设置进程和线程的各种信息,比如优先级、工作集等。

本书为立志掌握 Windows 操作系统编程的人提供了一条有效的捷径,对于不同层次的 Windows 操作系统编程人员来说,本书都极具参考价值,是一本不可多得的参考书。

参与本书的主要编写人员有张鸿超、张宏林等。由于 Windows 操作系统及其 API 函数所涉及的知识面极为广泛,而作者的知识又很有限,所以尽管我们对本书中所涉及的内容一再推敲和仔细调试,仍有可能出现错误和纰漏,希望广大读者批评指正。

编者

前 言

Microsoft Win32 API 是 Microsoft Windows 操作系统提供的应用程序接口函数,它基本上概括了应用程序编程中涉及的每个方面。我们在使用 MFC 编程时,所用的 MFC 函数都是封装在 Win32 API 基础之上的,即 MFC 是 Win32 API 的一个上层架构。学习 Win32 API 能更深刻的理解 MFC 的运作机制,同时使用 Win32 API 函数编程要比使用 MFC 更灵活,能编写出更加高效的程序。Microsoft Win32 API 也可以在 VB 和 Delphi 等语言中使用。

Microsoft Win32 API 函数按照功能划分为窗口处理、设备上下文——DC、绘图函数、位图和图标、菜单处理、文件处理、同步、处理文本和字体、硬件和系统、Windows 消息、进程和线程。

本书按功能对 Win32 API 函数进行了划分,对于每一个功能部分的函数都具体给出了它们的函数原型、功能和用法、注意事项、所在头文件、参数调用和典型示例等信息。本书内容如下:

第 1 章介绍了 Win32 应用程序的一些基本知识,例如系统消息传递机制,图形接口编程等,同时,它也介绍了 Win32 API 函数的一些基本知识。

第 2 章介绍了关于窗口处理部分的 Win32 API 函数,例如窗口创建,撤销等。

第 3 章介绍了设备上下文——DC,例如创建和获取 DC、坐标变换等。

第 4 章介绍了绘图函数,例如路径、图元文件等。

第 5 章介绍了位图和图标,例如从 DLL 中获取图标,设置 DIB 的颜色表等。

第 6 章介绍了菜单的处理,例如上下文菜单显示,设置位图菜单项等。

第 7 章介绍了文件的处理,例如文件查找,加锁和解锁文件等。

第 8 章介绍了同步,例如互斥对象,信号量对象和关键段对象等。

第 9 章介绍了文本和字体的处理,例如获取系统中字体的信息,输出各种形式的文本等。

第 10 章介绍了硬件和系统,例如捕获鼠标,获取各种系统信息等。

第 11 章介绍了 Windows 消息,例如消息的获取和发送等。

第 12 章介绍了进程和线程,例如调试进程和线程,获取和设置进程和线程的各种信息,比如优先级、工作集等。

本书为立志掌握 Windows 操作系统编程的人提供了一条有效的捷径,对于不同层次的 Windows 操作系统编程人员来说,本书都极具参考价值,是一本不可多得的参考书。

参与本书的主要编写人员有张鸿超、张宏林等。由于 Windows 操作系统及其 API 函数所涉及的知识面极为广泛,而作者的知识又很有限,所以尽管我们对本书中所涉及的内容一再推敲和仔细调试,仍有可能出现错误和纰漏,希望广大读者批评指正。

目 录

第 1 章 概述	1
1.1 Win32 程序概述	1
1.2 图形界面编程	1
1.3 消息传递机制	7
1.4 Win32 API 概述	7
1.5 为什么用 API	8
1.6 如何在程序中调用 API 函数	8
1.7 本书示例约定	8
第 2 章 窗口处理	10
2.1 窗口简介	10
2.2 窗口的创建和撤销	10
2.2.1 创建普通窗口 (CreateWindow、CreateWindowEx)	10
2.2.2 关闭窗口 (CloseWindow)	19
2.2.3 销毁窗口 (DestroyWindow)	20
2.3 窗口之间的关系	21
2.3.1 父子关系 (SetParent、GetParent、IsChild)	21
2.3.2 遍历窗口 (EnumWindow、EnumWindowProc、EnumChildWindows、 EnumChildProc)	24
2.3.3 获取指定的子窗口 (ChildWindowFromPoint、ChildWindowFromPointEx)	27
2.3.4 获得特定的窗口 (GetWindow、GetTopWindow、GetNextWindow、 GetDesktopWindow)	29
2.4 窗口的大小、位置和布局	33
2.4.1 获取窗口大小 (GetWindowRect、GetClientRect)	33
2.4.2 调整窗口大小 (AdjustWindowRect、AdjustWindowRectEx)	35
2.4.3 设置和获取窗口位置 (SetWindowPos)	37
2.4.4 移动窗口位置 (MoveWindow)	39
2.4.5 设置一组窗口位置 (BeginDeferWindowPos、DeferWindowPos、 EndDeferWindowPos)	40
2.4.6 设置和获取窗口布局 (SetWindowPlacement、GetWindowPlacement)	43
2.4.7 设置和获得视口坐标起点 (SetViewportOrgEx、GetViewportOrgEx)	44
2.4.8 设置和获取视口坐标范围 (SetViewportExtEx、GetViewportExtEx)	46
2.4.9 设置和获取窗口坐标起点 (SetWindowOrgEx、GetWindowOrgEx)	48
2.4.10 设置和获取窗口坐标范围 (SetWindowExtEx、GetWindowExtEx)	49
2.4.11 坐标起点的偏移操作 (OffsetViewportOrgEx、OffsetWindowOrgEx)	51
2.5 显示窗口	53
2.5.1 显示单个窗口 (ShowWindow、ShowWindowAsync、OpenIcon)	53
2.5.2 弹出式窗口 (ShowOwnedPopups、GetLastActivePopup)	56

2.5.3	显示子窗口 (CascadeWindows、 TileWindows、 ArrangeIconicWindows)	58
2.5.4	判断各种显示状态 (IsIconic、 IsWindowVisible、 IsZoomed)	62
2.6	窗口属性	63
2.6.1	设置和获取窗口名 (SetWindowText、 GetWindowText、 GetWindowTextLength)	63
2.6.2	设置和获取窗口标记长字 (SetWindowLong、 GetWindowLong)	66
2.6.3	设置和获取前台窗口 (SetForegroundWindow、 GetForegroundWindow)	68
2.7	对话框	69
2.7.1	创建对话框 (DialogBox、 CreateDialog、 CreateDialogIndirect、 Message Box、 MessageBoxEx、 MessageBoxIndirect)	69
2.7.2	获取控件及其 ID (GetDlgItem、 GetDlgItemID)	78
2.7.3	设置获取控件的显示文本 (SetDlgItemText、 SetDlgItemInt、 GetDlgItemText、 GetDlgItemInt)	80
2.7.4	获取下一个 TAB 控件 (GetNextDlgTabItem)	83
2.7.5	对话框过程 (DefDlgProc、 DialogProc)	84
2.7.6	对话框坐标和屏幕坐标间的转换 (MapDialogRect)	85
2.7.7	撤销对话框 (EndDialog)	86
2.8	其他	87
2.8.1	查找窗口 (FindWindow、 FindWindowEx)	87
2.8.2	设置窗口 Z 坐标 (BringWindowToTop)	88
2.8.3	确定窗口属性 (IsWindow、 IsWindowUnicode)	89
2.8.4	设置和获取激活态窗口 (SetActiveWindow、 GetActiveWindow)	91
2.8.5	窗口闪烁 (FlashWindow)	92
2.8.6	获取指定点处的窗口 (WindowFromPoint)	93
2.8.7	遍历线程所有窗口 (EnumThreadWindows、 EnumThreadWndProc)	94
第 3 章	设备上下文——DC	97
3.1	DC 简介	97
3.2	DC 的获取和撤销	97
3.2.1	创建 DC (CreateDC、 CreateCompatibleDC)	97
3.2.2	撤销 DC (ReleaseDC、 DeleteDC)	99
3.2.3	获得窗口 DC (GetWindowDC、 GetDC、 GetDCEX)	100
3.2.4	保存和恢复 DC (SaveDC、 RestoreDC)	103
3.3	设置和获取 DC 的各种属性	105
3.3.1	获得 DC 起始点坐标 (GetDCOrgEx)	105
3.3.2	设置显示配置 (ChangeDisplaySettings、 ChangeDisplaySettingsEx)	106
3.3.3	遍历显示设备和其配置 (EnumDisplayDevices、 EnumDisplaySettings)	108
3.3.4	设置和获取 DC 画刷颜色 (SetDCBrushColor、 GetDCBrushColor)	109
3.3.5	设置和获取 DC 画笔颜色 (SetDCPenColor、 GetDCPenColor)	110
3.3.6	设置和获取 DC 中各种对象 (SelectObject、 GetCurrentObject)	111
3.3.7	获取 DC 中对象的信息 (GetObject、 GetObjectType、 GetStockObject)	114

3.4	区域操作	117
3.4.1	新建椭圆区域 (CreateEllipticRgn、CreateEllipticRgnIndirect)	117
3.4.2	新建多边形区域 (CreatePolygonRgn、CreatePolyPolygonRgn)	118
3.4.3	新建矩形区域 (CreateRectRgn、CreateRectRgnIndirect)	121
3.4.4	新建圆矩形区域 (CreateRoundRectRgn)	122
3.4.5	区域之间的操作 (EqualRgn、ExcludeUpdateRgn)	123
3.4.6	区域上绘图操作 (FillRgn、FrameRgn、InvertRgn、PaintRgn)	125
3.5	裁剪操作	128
3.5.1	设置和获取裁剪区域 (SelectClipRgn、SelectClipPath、GetClipRgn、ExtSelectClipRgn)	128
3.5.2	裁剪区域间的运算 (ExcludeClipRect、IntersectClipRect、OffsetClipRgn)	132
3.6	坐标变换	135
3.6.1	客户区坐标与屏幕坐标的变换 (ClientToScreen、ScreenToClient)	135
3.6.2	设备坐标与逻辑坐标的变换 (DPtoLP、LPtoDP)	137
3.6.3	窗口间的坐标变换 (MapWindowPoints)	139
3.7	颜色和调色板	140
3.7.1	创建新调色板 (CreatePalette)	140
3.7.2	启用调色板 (SelectPalette、RealizePalette)	141
3.7.3	设置和获取调色板颜色条目 (SetPaletteEntries、GetPaletteEntries)	143
3.7.4	设置系统调色板的工作方式 (SetSystemPaletteUse)	145
3.7.5	设置和获取颜色调整 (SetColorAdjustment、GetColorAdjustment)	146
3.7.6	获取匹配颜色 (GetNearestColor、GetNearestPaletteIndex)	147
3.8	其他	149
3.8.1	滚动窗口 (ScrollDC)	149
3.8.2	偏移操作 (OffsetRect、OffsetRgn)	150
3.8.3	点、矩形和区域的关系 (PtInRegion、PtVisible、RectInRegion、RectVisible)	152
3.8.4	缩放视口和窗口大小 (ScaleViewportExtEx、ScaleWindowExtEx)	155
3.8.5	由给定 DC 获取相联系的窗口 (WindowFromDC)	157
3.8.6	获取特定设备的信息 (GetDeviceCaps)	158
3.8.7	撤销 DC 中的对象 (DeleteObject)	160
第 4 章	绘图函数	162
4.1	基本画图函数	162
4.1.1	点 (SetPixel、SetPixelV、GetPixel)	162
4.1.2	线 (LineTo、LineDDA、LineDDAProc、MoveToEx、Polyline、PolylineTo)	164
4.1.3	曲线 (PolyBezier、PolyBezierTo、PolyDraw)	170
4.1.4	弧 (AngleArc、Arc、ArcTo、SetArcDirection、GetArcDirection)	173
4.1.5	椭圆 (Ellipse)	177
4.1.6	弦 (Chord)	178
4.1.7	矩形 (Rectangle、Roundrect、DrawEdge、DrawFocusRect、FillRect)	180
4.1.8	多边形 (Polygon、PolyPolygon、PolyPolyline)	185

4.1.9	饼图 (Pie)	188
4.2	路径相关函数	189
4.2.1	启动和终止路径绘图 (BeginPath、EndPath)	189
4.2.2	曲线到直线 (FlattenPath)	191
4.2.3	填充路径 (FillPath、StrokeAndFillPath)	192
4.2.4	边界处理 (StrokePath、WidenPath)	194
4.2.5	封闭路径 (CloseFigure)	195
4.2.6	获取路径区域 (PathToRegion)	196
4.2.7	获取路径 (GetPath)	197
4.2.8	撤销路径 (AbortPath)	198
4.3	处理 DC 对象	199
4.3.1	新建画笔 (CreatePen、CreatePenIndirect、ExtCreatePen)	199
4.3.2	新建画刷(CreateBrushIndirect、CreateHatchBrush、CreateDIBPatternBrushPt、CreatePatternBrush、CreateSolidBrush、GetSysColorBrush)	202
4.3.3	设置和获取背景颜色和模式 (GetBkColor、GetBkMode、SetBkColor、SetBkMode)	209
4.4	图元文件	211
4.4.1	创建图元文件 (CreateEnhMetaFile、CreateMetaFile)	211
4.4.2	复制图元文件 (CopyEnhMetaFile、CopyMetaFile)	213
4.4.3	打开图元文件 (GetEnhMetaFile)	215
4.4.4	关闭图元文件 (CloseEnhMetaFile、CloseMetaFile)	215
4.4.5	撤销图元文件 (DeleteEnhMetaFile、DeleteMetaFile)	216
4.4.6	绘制图元文件 (PlayEnhMetaFile)	217
4.5	其他	218
4.5.1	启动和终止绘图 (BeginPaint、EndPaint)	218
4.5.2	获取需更新区域 (GetUpdateRect、GetUpdateRgn)	220
4.5.3	绘图操作生效(InvalidateRect、InvalidateRgn、ValidateRect、ValidateRgn)	222
4.5.4	设置和获取最大绘图缓冲 (GdiSetBatchLimit、GdiGetBatchLimit)	226
4.5.5	刷新缓冲 (GdiFlush)	227
4.5.6	设置和获取窗口可绘图区域 (SetWindowRgn、GetWindowRgn)	228
4.5.7	禁止在窗口绘图 (LockWindowUpdate)	230
4.5.8	绘制桌面 (PaintDesktop)	231
4.5.9	更新窗口 (RedrawWindow)	231
4.5.10	撤销绘图操作 (CancelDC)	232
4.5.11	描绘标准控件 (DrawFrameControl)	233
4.5.12	设定画图效果 (DrawState、DrawStateProc)	235
4.5.13	设置和获取系统颜色 (SetSysColors、GetSysColor)	238
第 5 章	位图和图标	241
5.1	位图和图标简介	241
5.2	创建和撤销位图、图标	241
5.2.1	创建位图(CreateBitmap、CreateBitampIndirect、CreateCompatible、Bitmap、CreateDIBitmap、CreateDIBSection、LoadBitmap、LoadImage)	241
5.2.2	创建图标 (CreateIcon、LoadIcon、LoadCursorFromFile)	251

5.2.3	撤销图标 (DestoryIcon)	255
5.3	设置和获得位图和图标的各种属性	256
5.3.1	设置和获取位图的位数据 (SetBitmapBits、GetBitmapBits)	256
5.3.2	设置和获取 DIB (SetDIBits、GetDIBits)	258
5.3.3	设置和获取 DIB 颜色表 (SetDIBColorTable、GetDIBColorTable)	261
5.3.4	设置和获取位图的大小 (GetBitmapDimensionEx、SetBitmapDimensionEx)	263
5.3.5	获取图标信息 (GetIconInfo)	265
5.4	位块操作	266
5.4.1	各种位块操作方式 (BitBlt、PatBlt、PlgBlt、StretchBlt、MaskBlt、TransparentBlt)	266
5.4.2	设置和获取位操作模式 (SetROP2、GetROP2、SetStretchBltMode、GetStretchBltMode)	273
5.5	其他	276
5.5.1	复制位图和图标 (CopyImage、CopyIcon)	276
5.5.2	从文件中搜寻图标 (ExtractIcon、ExtractIconEx、ExtractAssociatedIcon)	278
5.5.3	显示图标 (DrawIcon、DrawIconEx)	281
第 6 章	菜单处理	284
6.1	菜单的简单操作	284
6.1.1	生成新菜单 (CreateMenu、CreatePopupMenu)	284
6.1.2	装载菜单 (LoadMenu、LoadMenuIndirect)	286
6.1.3	撤销菜单 (DestoryMenu、DeleteMenu、RemoveMenu)	288
6.1.4	获取菜单 (GetMenu、GetSubMenu、GetSystemMenu)	290
6.1.5	增添菜单 (AppendMenu)	293
6.2	菜单项处理	295
6.2.1	设置和获取菜单项信息 (SetMenuItemInfo、GetMenuItemInfo)	295
6.2.2	设置和获取缺省菜单项 (SetMenuDefaultItem、GetMenuDefaultItem)	297
6.2.3	获取菜单项的 ID (GetMenuItemID)	299
6.2.4	获取菜单项数 (GetMenuItemCount)	300
6.2.5	获取菜单项的范围大小 (GetMenuItemRect)	301
6.2.6	高亮显示菜单项 (HiliteMenuItem)	302
6.2.7	菜单项属性 (CheckMenuItem、EnableMenuItem、CheckMenuRadioItem)	303
6.2.8	添加菜单项 (InsertMenuItem)	306
6.2.9	获取某点处的菜单项 (MenuItemFromPoint)	307
6.3	其他	308
6.3.1	设置窗口的菜单 (SetMenu)	308
6.3.2	判定句柄是否指向菜单 (IsMenu)	308
6.3.3	上下文菜单 (TrackPopupMenu、TrackPopupMenuEx)	309
6.3.4	显示菜单条 (DrawMenuBar)	312
6.3.5	图形菜单项 (SetMenuItemBitmaps)	313

第 7 章	文件处理	315
7.1	创建、读写和撤销文件	315
7.1.1	创建文件 (CreateFile、lcreat、GetTempFileName)	315
7.1.2	设置文件读写位置 (SetFilePointer、_llseek、_lseek)	319
7.1.3	写入文件 (WriteFile、lwrite、hwrite)	322
7.1.4	异步写入文件 (WriteFileEx、FileIOCompletionRoutine)	324
7.1.5	读入文件 (ReadFile、_lread、_hread)	326
7.1.6	异步读入文件 (ReadFileEx)	329
7.1.7	复制文件 (CopyFile、CopyFileEx、CopyProgressRoutine)	330
7.1.8	更改文件名 (MoveFile、MoveFileEx)	333
7.1.9	关闭文件 (CloseHandle、_lclose)	335
7.1.10	删除文件 (DeleteFile)	336
7.2	设置和获得文件的各种信息	337
7.2.1	获取文件大小 (GetFileSize)	337
7.2.2	获取文件类型 (GetFileType)	338
7.2.3	设置和获取文件时间信息 (SetFileTime、GetFileTime)	339
7.2.4	获取文件版本信息 (GetFileVersionInfo、GetFileVersionInfoSize、VerQueryValue)	341
7.2.5	是否可执行文件 (GetBinaryType)	344
7.2.6	长文件名和短文件名的转换 (GetFullPathName、GetShortPathName)	345
7.2.7	由文件句柄获取文件信息 (GetFileInformationByHandle)	347
7.2.8	设置文件结束标志 (SetEndOfFile)	348
7.2.9	设置和获取文件属性 (SetFileAttributes、GetFileAttributes)	349
7.3	目录处理	351
7.3.1	创建新目录 (CreateDirectory、CreateDirectoryEx)	351
7.3.2	删除目录 (RemoveDirectory)	352
7.3.3	获取系统路径 (GetSystemDirectory)	353
7.3.4	获取临时路径 (GetTempPath)	354
7.3.5	获取 Windows 完整目录 (GetWindowsDirectory)	355
7.3.6	设置和获取当前目录 (SetCurrentDirectory、GetCurrentDirectory)	356
7.4	文件查找	358
7.4.1	查找第一个文件 (FindFirstFile)	358
7.4.2	查找下一个文件 (FindNextFile)	359
7.4.3	查找指定文件 (SearchPath)	359
7.4.4	查找结束 (FindClose)	361
7.5	磁盘信息	361
7.5.1	获取可用逻辑驱动器 (GetLogicalDrives)	361
7.5.2	获取可用逻辑驱动器信息 (GetLogicalDriveStrings)	362
7.5.3	设置和获取卷标信息 (SetVolumeLabel、GetVolumeInformation)	363
7.5.4	获取磁盘空闲空间 (GetDiskFreeSpace、GetDiskFreeSpaceEx)	366
7.5.5	获取磁盘类型 (GetDriveType)	369
7.6	压缩文件	370
7.6.1	初始化内部缓冲区 (LZInit)	370

7.6.2	打开压缩文件 (LZOpenFile)	371
7.6.3	读入压缩文件 (LZRead)	372
7.6.4	压缩文件指针定位 (LZSeek)	373
7.6.5	获取压缩文件大小 (GetCompressedFileSize)	374
7.6.6	获取压缩文件全名 (GetExpandedName)	375
7.6.7	复制和解压缩文件 (LZCopy)	376
7.6.8	关闭压缩文件 (LZClose)	377
7.7	其他	378
7.7.1	设置和获取字符集 (SetFileApisToANSI、SetFileApisToOEM、AreFileApisANSI)	378
7.7.2	联系文件和 I/O 端口 (CreateIoCompletionPort、PostQueuedCompletionStatus、GetQueuedCompletionStatus)	380
7.7.3	更新文件缓冲区 (FlushFileBuffers)	384
7.7.4	加锁和解锁文件 (LockFile、LockFileEx、UnlockFile、UnlockFileEx)	384
7.7.5	撤销文件 IO (CancelIO)	388
7.7.6	比较文件时间信息 (CompareFileTime)	389
7.7.7	设备 IO 操作 (DeviceIoControl)	390
7.7.8	DOS 时间转换为 Win32 时间 (DosDateTimeToFileTime)	392
7.7.9	Win32 时间转换为 Dos 时间 (FileTimeToDosDateTime)	393
7.7.10	Win32 时间转换本地时间 (FileTimeToLocalFileTime)	394
第 8 章	同步	396
8.1	同步对象	396
8.1.1	生成同步对象 (CreateEvent、CreateMutex、CreateSemaphore、Create WaitableTimer)	396
8.1.2	打开同步对象 (OpenEvent、OpenMutex、OpenSemaphore、Open Waitable Timer)	404
8.1.3	关键段 (EnterCriticalSection、LeaveCriticalSection、DeleteCritiaclSection)	410
8.2	关键区段	413
8.2.1	初始化关键区段 (InitializeCriticalSection、InitializeCriticalSectionAnd SpinCount)	413
8.2.2	非阻塞进入关键区段 (TryEnterCriticalSection)	415
8.2.3	设置关键区段的计数 (SetCriticalSecitonSpinCount)	415
8.3	原子性访问变量	416
8.3.1	原子性比较 (InterlockedCompareExchange)	416
8.3.2	原子性交换 (InterlockedExchange)	417
8.3.3	原子性增加 (InterlockedIncrement、InterlockedExchangeAdd)	418
8.3.4	原子性减少 (InterlockedDecrement)	420
8.4	等待事件发生	421
8.4.1	触发一个事件并等待另一个事件 (SignalObjectAndWait)	421
8.4.2	等待单个事件发生 (WaitForSingleObject、WaitForSingleObjectEx)	422
8.4.3	等待多个事件发生 (WaitForMultipleObjects、WaitForMultipleObjectsEx)	423

8.4.4	等待消息 (MsgWaitForMultipleObjects、MsgWaitForMultipleObjectsEx)	427
8.5	其他	430
8.5.1	设置事件状态 (SetEvent、ResetEvent、PulseEvent)	430
8.5.2	释放互斥量 (ReleaseMutex)	433
8.5.3	释放信号灯 (ReleaseSemaphore)	433
8.5.4	设置可等待定时器状态 (SetWaitableTimer、CancelWaitableTimer)	434
第 9 章	处理文本和字体	436
9.1	文本	436
9.1.1	显示文本 (DrawText、DrawTextEx、TextOut、ExTextOut、TabbedTextOut、GrayString、OutputProc、PolyTextOut)	436
9.1.2	设置和获取文本属性 (SetTextAlign、GetTextAlign、SetTextColor、GetTextColor)	446
9.1.3	调整和获取字符间距 (SetTextCharacterExtra、GetKerningPairs)	451
9.1.4	获得字符串的宽和高 (GetTextExtentPoint32、GetTabbedTextExtent)	452
9.1.5	获取字符串的信息 (GetCharacterPlacement)	455
9.1.6	调整字符串 (SetTextJustification)	456
9.3	字体	458
9.3.1	创建新字体 (CreateFont、CreateFontIndirect、CreateScalableFontResource)	458
9.3.2	在系统中添加和删除字体 (AddFontResource、RemoveFontResource)	464
9.3.3	获取字体的宽度 (GetCharABCWidths、GetCharWidth、GetCharWidth32、GetCharWidthFloat、GetCharABCWidthsFloat)	465
9.3.4	获取字体的各种信息 (GetTextMetric、GetFontLanguageInfo)	470
9.3.5	字体的宽高比 (GetAspectRatioFilterEx、SetMapperFlags)	472
9.3.6	获取字体的字型名 (GetTextFace)	473
9.3.7	遍历系统字体 (EnumFontFamiliesEx、EnumFontFamiExProc)	474
第 10 章	硬件和系统	478
10.1	键盘	478
10.1.1	设置和获取键盘布局 (ActivateKeyboardLayout、GetKeyboardLayout、GetKeyboardLayoutList、GetKeyboardLayoutName)	478
10.1.2	装载和卸载键盘布局 (LoadKeyBoardLayout、UnloadKeyBoardLayout)	483
10.1.3	获得和设置键盘状态 (GetKeyboardState、SetKeyboardState)	485
10.1.4	获得键盘类型 (GetKeyboardType)	488
10.1.5	获得按键名 (GetKeyNameText)	490
10.1.6	获得按键状态 (GetKeyState、GetAsyncKeyState)	491
10.1.7	虚拟按键到扫描码 (MapVirtualKey、MapVirtualKeyEx)	493
10.1.8	虚拟按键到 ASCII 码 (ToAscii、ToAsciiEx)	496
10.1.9	虚拟按键到 Unicode (ToUnicode、ToUnicodeEx)	498
10.1.10	字符到虚拟按键 (VkKeyScan、VkKeyScanEx)	501

10.1.11	OEM 码到扫描码 (OemKeyScan)	503
10.1.12	注册和撤销系统热键 (RegisterHotKey、UnregisterHotKey)	504
10.1.13	插入键盘事件 (SendInput、keybd_event)	506
10.2	鼠标	508
10.2.1	获得鼠标输入点序列 (GetMouseMovePoints)	508
10.2.2	获得和设置双击间隔 (GetDoubleClickTime、SetDoubleClickTime)	509
10.2.3	鼠标左右键设置 (SwapMouseButton)	510
10.2.4	鼠标捕获和释放 (SetCapture、GetCapture、ReleaseCapture)	511
10.2.5	捕获离开和滑过事件 (TrackMouseEvent)	513
10.2.6	插入鼠标事件 (mouse_event)	514
10.2.7	探测拖动 (DragDetect)	516
10.3	光标	517
10.3.1	新建和撤销光标 (CreateCursor、LoadCursorFromFile、DestroyCursor)	517
10.3.2	设置和获取光标 (SetCursor、GetCursor)	521
10.3.3	设置和获取光标位置 (SetCursorPos、GetCursorPos)	522
10.3.4	设置和获取光标活动范围 (ClipCursor、GetClipCursor)	524
10.3.5	显示光标 (ShowCursor)	526
10.3.6	设置系统光标 (SetSystemCursor)	526
10.4	时钟	528
10.4.1	设置和获取系统时间 (SetSystemTime、GetSystemTime、GetSystemTimeAsFileTime)	528
10.4.2	设置和获取当地时间 (SetLocalTime、GetLocalTime)	531
10.4.3	获取自启动以来的微秒数 (GetTickCount)	532
10.4.4	获取和设置系统调整时间 (GetSystemTimeAdjustment、SetSystemTimeAdjustment)	533
10.4.5	设置和获取时区信息 (SetTimeZoneInformation、GetTimeZoneInformation)	535
10.4.6	获取高精度时间 (QueryPerformanceCounter、QueryPerformanceFrequency)	537
10.4.7	设置定时器 (SetTimer)	538
10.4.8	设置定时器回调函数 (TimerProc)	540
10.4.9	撤销定时器 (KillTimer)	540
10.5	系统信息	541
10.5.1	获取系统各种度量信息 (GetSystemMetrics)	541
10.5.2	设置和获取系统参数 (SystemParametersInfo)	544
10.5.3	确定处理器的特性 (IsProcessorFeaturePresent)	549
10.5.4	获取和设置计算机名 (GetComputerName、SetComputerName)	550
10.5.5	获取用户名 (GetUserName)	552
10.5.6	获取操作系统版本信息 (GetVersionEx)	552
10.5.7	获取系统信息 (GetSystemInfo)	553
第 11 章	Windows 消息	555
11.1	消息系统概述	555

11.2	消息处理	555
11.2.1	发送消息 (SendMessage、SendMessageCallback、SendNotifyMessage、SendMessageTimeout、PostThreadMessage、PostMessage、PostQutiMessage、BroadcastSystemMessage)	555
11.2.2	回复消息 (ReplyMessage)	565
11.2.3	获取消息 (GetMessage、WaitMessage、PeekMessage)	567
11.2.4	处理消息 (TranslateMessage、DispatchMessage)	570
11.2.5	设置和获取消息的附加信息 (SetMessageExtraInfo、GetMessageExtraInfo)	571
11.2.6	获取消息的伴随信息 (GetMessagePos、GetMessageTime)	573
11.2.7	注册新窗口消息 (RegisterWindowMessage)	574
11.3	消息队列	576
11.3.1	获取队列状态 (GetQueueStatus)	576
11.3.2	获取队列中输入消息状态 (GetInputState)	577
11.4	其他	578
11.4.1	是否重入 (InSendMessage、InSendMessageEx)	578
第 12 章	进程和线程	580
12.1	进程和线程概述	580
12.2	进程	580
12.2.1	创建新进程 (CreateProcess、CreateProcessAsUser)	580
12.2.2	获取进程对象 (OpenProcess)	585
12.2.3	获取进程起始信息 (GetStartupInfo)	587
12.2.4	设置和获取工作集大小 (SetProcessWorkingSetSize、GetProcessWorkingSetSize)	588
12.2.5	设置和获取进程的优先级 (SetPriorityClass、GetPriorityClass)	589
12.2.6	进程的动态优先级 (SetProcessPriorityBoost、GetProcessPriorityBoost)	591
12.2.7	设置和获取退出参数 (SetProcessShutdownParameters、GetProcessShutdownParameters)	593
12.2.8	获取进程的退出代码 (GetExitCodeProcess)	594
12.2.9	设置和获取环境参数 (GetEnvironmentStrings、SetEnvironmentVariable、GetEnvironmentVariable、FreeEnvironmentVariable)	595
12.2.10	获取进程的时间信息 (GetProcessTime)	598
12.2.11	获取进程的版本信息 (GetProcessVersion)	599
12.2.12	获取当前进程 (GetCurrentProcess、GetCurrentProcessID)	600
12.2.13	等待进程空闲 (WaitForInputIdle)	601
12.2.14	结束进程 (ExitProcess、TerminateProcess)	601
12.2.15	获取命令参数 (GetCommandLine、CommandLineToArgvW)	604
12.3	线程	605
12.3.1	创建新线程 (CreateThread)	605
12.3.2	设置和获得线程优先级 (SetThreadPriority、GetThreadPriority)	607
12.3.3	线程的调度 (SuspendThread、SwitchToThread、ResumeThreadExitThread、TerminateThread、Sleep、SleepEx)	610
12.3.4	设置线程的输入队列 (AttachThreadInput)	616

12.3.5	线程的动态优先级 (SetThreadPriorityBoost、GetThreadPriorityBoost)	618
12.3.6	获取线程的时间信息 (GetThreadTimes)	619
12.3.7	获取线程的退出码 (GetExitCodeThread)	620
12.4	调试	621
12.4.1	调试指定进程 (DebugActiveProcess)	621
12.4.2	设置断点 (DebugBreak)	624
12.4.3	等待异常发生 (WaitForDebugEvent)	625
12.4.4	重新启动线程 (ContinueDebugEvent)	626
12.4.5	设置调试级别 (SetDebugErrorLevel)	627
12.4.6	发送一串字符给调试器 (OutputDebugString)	627
12.4.7	是否处于被调试态 (IsDebuggerPresent)	628
12.4.8	控制转移到调试器 (FatalExit)	629
12.4.9	设置和获取线程上下文 (SetThreadContext、GetThreadContext)	629
12.4.10	获取线程选择子的描述符 (GetThreadSelectorEntry)	631
12.4.11	读写进程内存 (WriteProcessMemory、ReadProcessMemory)	632
12.5	线程局部数据区	635
12.5.1	分配 (TlsAlloc)	635
12.5.2	释放 (TlsFree)	636
12.5.3	设置和获取变量值 (TlsSetValue、TlsGetValue)	637
12.6	进程间通信	638
12.6.1	创建映射文件 (CreateFileMapping)	638
12.6.2	打开映射文件 (OpenFileMapping)	640
12.6.3	映射文件到内存 (MapViewOfFile、MapViewOfFileEx)	641
12.6.4	撤销映射 (UnmapViewOfFile)	644
附录 A	数据结构	645
附录 B	Windows 消息	698
附录 C	类型与宏	746

第 1 章 概述

1.1 Win32程序概述

现在流行的 Windows 操作系统有三个序列：第一个是 32 位和 16 位混合的操作系统系列 Windows 3.x、Windows 9x、Windows Me，其中 Windows Me 是该系列的终结版本；另一个就是基于 NT 技术构建的 Windows NT、Windows 2000、Windows XP 系列；最后一个为嵌入式的操作系统 Windows CE。在这三个不同系列的 Windows 操作系统所提供的 API 函数也有比较大的区别。

一个操作系统的作用就是屏蔽一些复杂的直接硬件操作并提供给用户一个简单明确的应用接口，另外对于一些基本的或常用的操作也以 API 的形式提供给用户，比如内存管理、文件管理等。我们在编制基于 Windows 的应用程序时，就可以直接调用这些 API 函数来实现指定的功能。

在使用 Win32 API 时，我们要注意它的使用平台，很多 API 函数对于嵌入式的 Windows CE 操作系统是没有定义的。同时，一个 API 虽然可以在不同的系统平台上应用，但它们的调用条件是不同的。

1.2 图形界面编程

图形界面编程是典型的事件驱动的程序模式，这种程序启动以后便进入消息循环，它不断处理用户或系统的各种输入，当它处理到 WM_QUIT 消息时才退出该循环，从而结束该应用程序。现在流行的 MFC 编程就是基于此而附加的一层处理，它使程序员能更方便地处理输入事件。因此，理解了 Windows 的底层消息处理机制，便能很好地理解 MFC 机制。下面以一个实例来说明如何使用 Visual C++ 6.0 创建一个 Win32 程序。

- 第一步 启动 Visual C++，选择【File】菜单中的【New】选项，如图 1-1 所示。

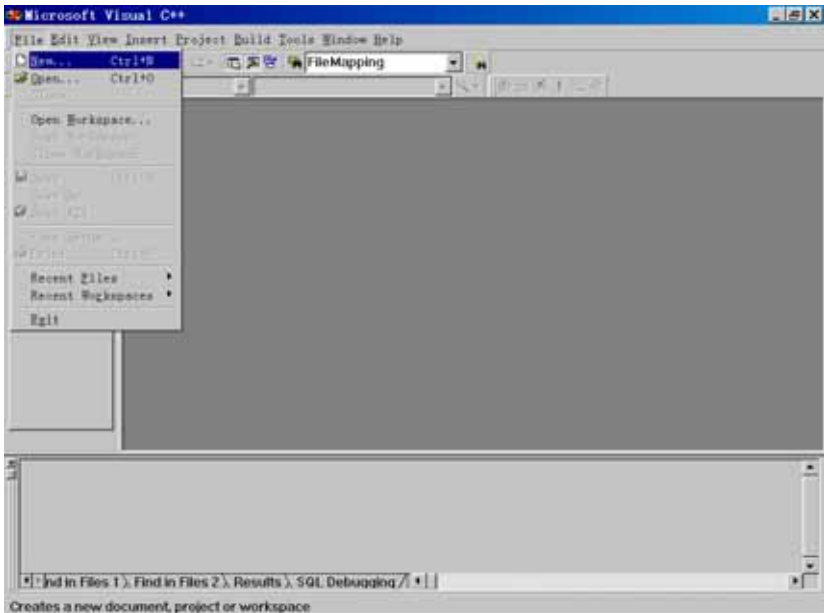


图 1-1 用 Visual C++新建 Win32 程序

- 第二步 在出现的对话框中的【Project】页中选择“Win32 Application”选项，并在“Project name”文本框中键入 Example 作为项目名，然后单击【OK】按钮，如图 1-2 所示。

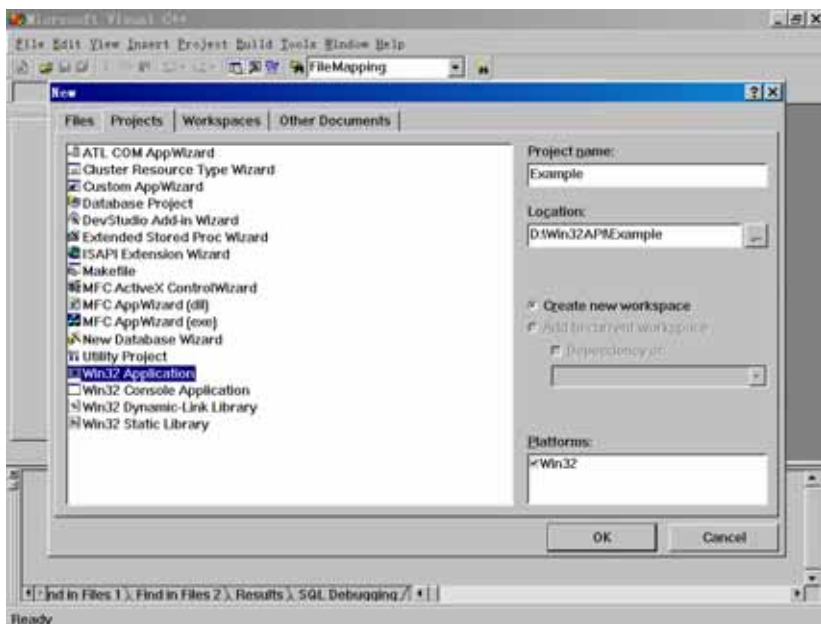


图 1-2 设置程序类型、项目名称和路径

- 第三步 选择“A typical ‘Hello World’ application.”选项，单击【Finish】按钮，如图 1-3 所示。

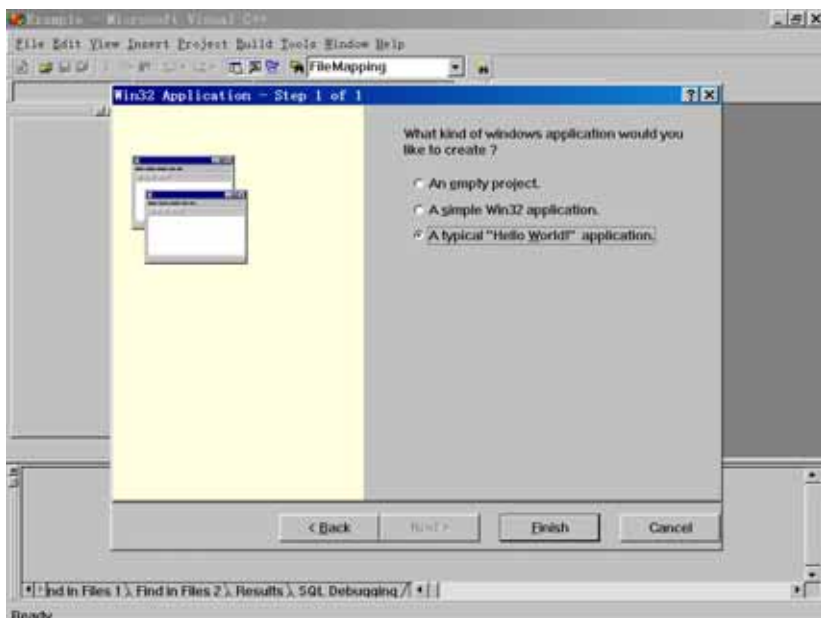


图 1-3 选择创建典型的“Hello World”程序

- 第四步 此时会出现一个对话框，报告项目信息，如图 1-4 所示。单击【OK】按钮完成该程序的创建。