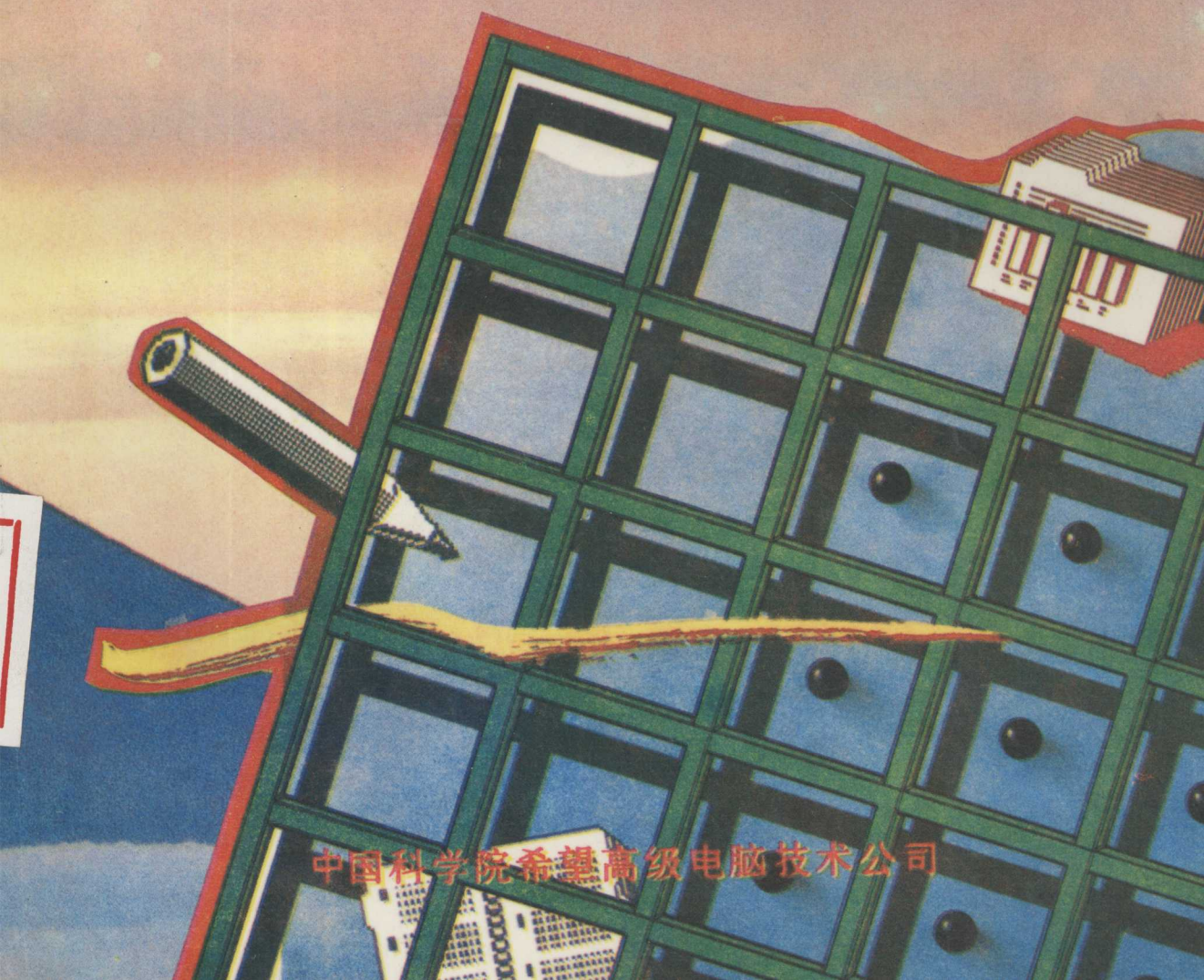
适用于IBM PCAT286.386.486

3.0

Microsoft Windows

最新窗口软件程序员参考手册

（下册）



中国科学院希望高级电脑技术公司

Microsoft Windows (3.0 版)

最新窗口软件程序员参考手册

(下册)

刘京涌　胡达　编译

中国科学院希望高级电脑技术公司

一九九一年四月

目　录

第三部分　其它参考信息

第三部分描述 Windows 应用程序接口的其它参考信息，它是前面内容的补充。

章节

第七章　数据类型和结构

本章描述的是 Microsoft Windows 函数和信息中使用的数据类型和结构，包括两个部分：数据类型表和 Windows 数据结构表，其中每个部分均按字母顺序排列。

7.1　数据类型

下列表中的数据类型是用来定义大小、参数意义和 Windows 函数返回值的关键字，表中包括字符、整数、布尔型、指针型及指针。字符、整数和布尔型与编译程序大致相同，大部分指针类型都是以 P(短指针)或 LP(长指针)做前缀开头的。短指针指向当前数据段内的数据，长指针包括一个 32 位数据段/偏移量。Windows 应用程序使用指针访问已存内存的资源，Windows 通过一些表提供对这些资源的访问，这些表中包括每个指针的专用项，每个专用项包括资源的地址和标识资源类型的方法。

Windows 数据类型定义如下表:

类型	定义
BOOL	16位布尔值
BYTE	无符号8位整数。
Char	ASCII字符或有符号8位整数。
DWORD	无符号32位整数或一个字段/偏移地址。
FAR	可用于生成长指针的数据类型属性。
FARPROC	指向函数的指针，由调用MakeProcInstance函数获得。
GLOBALHANDLE	整个内存的指针，是由系统堆栈中分配出来的内存块的16位变址。
HANDLE	一般指针，表示一个判定程序数据的表项的16位变址。
HBITMAP	物理位图的指针，是GDI物理绘图目标的16位变址。
HBRUSH	物理刷指针，是GDI物理绘图目标的16位变址。
HCURSOR	光标指针，是资源表项的16位变址。
HDC	显示关联的指针，是GDI设备表的16位变址。
HFONT	物理字指针，是GDI物理绘图目标的16位变址。
HICON	图象资源指针，是资源表目的16位变址。
HMENU	菜单资源指针，是资源表目的16位变址。
HPALETTE	逻辑调色指针，是GDI物理绘图目标的16位变址。
HPEN	物理笔指针，是GDI物理绘图目标的16位变址。
HRGN	物理区域指针，是GDI物理绘图目标的16位变址。
HSTR	串指针，是资源表目的16位变址。
LOCALHANDLE	本内存区指针，是由应用程序堆栈中分配出来的一块内存区的16位变址。
long	有符号的32位整数。

LONG	有符号的32位整数。
LPBITMAP	指向BITMAP数据结构的长指针。
LPBITMAPCOREHEADER	指向BITMAPCOREHEADER数据结构的长指针。
LPBITMAPCOREINFO	指向BITMAPCOREINFO数据结构的长指针。
LPBITMAPFILEHEADER	指向BITMAPFILEHEADER数据结构的长指针。
LPBITMAPINFO	指向BITMAPINFO数据结构的长指针。
LPBITMAPINFOHEADER	指向BITMAPINFOHEADER数据结构的长指针。
LPCOMPAREITEMSTRUCT	指向COMPAREITEMSTRUCT数据结构的长指针。
LPCREATESTRUCT	指向CREATESTRUCT数据结构的长指针。
LPDELETEITEMSTRUCT	指向DELETEITEMSTRUCT数据结构的长指针。
LPHANDLETABLE	指向HANDLETABLE数据结构的长指针。
LPINT	指向有符号的16位整数的长指针。
LPLOGBRUSH	指向LOGBRUSH数据结构的长指针。
LPLOGFONT	指向LOGFONT数据结构的长指针。
LPLOGPALETTE	指向LOGPALETTE数据结构的长指针。
LPLOGPEN	指向LOGPEN数据结构的长指针。
LPMEASUREITEMSTRUCT	指向MEASUREITEMSTRUCT数据结构的长指针。
LPMETAFILEPICT	指向METAFILEPICT数据结构的长指针。
LPMSG	指向MSG数据结构的长指针。
LPOFSTRUCT	指向OFSTRUCT数据结构的长指针。
LPPALETTEENTRY	指向PALETTEENTRY数据结构的长指针。
LPPOINT	指向POINT数据结构的长指针。
LPRECT	指向RECT数据结构的长指针。
LPRESOURCELIST	指向一个或多个RESOURCELIST数据结构的长指针。
LPSTR	指向一个字符串的长指针。
LPTEXTMETRIC	指向TEXTMETRIC数据结构的长指针。
LPVOID	指向一个未定义的数据类型的长指针。
LPWNDCLASS	指向WNDCLASS数据结构的长指针。
NEAR	可用于产生一个短指针的数据类型属性。
NPSTR	指向一个字符串的近指针。
PINT	指向一个带符号的16位整数的指针。
PSTR	指向一个字符串的指针。
PWORD	指向一个无符号的16位整数的指针。
short	带符号的16位整数。
void	空值，用于表示一个函数没有返回值。
WORD	无符号的16位整数。

7.2 数据结构

本节列举了 Windows 使用的数据结构，这些数据结构是按字母顺序介绍的，先给出结构的定义，然后是每个字段的描述。

位图

位图的数据结构:

位图结构定义了逻辑位图的高度、宽度、颜色格式和位值。

```
typedef struct tagBITMAP{
    short  bmType;
    short  bmWIdth;
    short  bmHeight;
    short  bmWidthBytes;
    BYTE   bmPlanes;
    BYTE   bmBitsPixel;
    LPSTR  bmBits;
  }BITMAP;
```

BITMAP 结构有如下字段:

字段名	说明
bmType	规定位图类型，对于逻辑位图，bmType字段必须为0。
bmWidth	规定位图(以象素表示)的宽度。必须大于0。
bmHeight	规定位图的高度(以光栅行表示)。必须大于0。
bmWidthBytes	规定每个光栅行中的字节数。这个数必须是偶数，因为图象设备接口采用的是由位图的位值组成的整数值(二字节)矩阵，换句话说，bmWidthbytes×8必须是16的两倍以上，大于或等于bmWidth字段。
bmPlanes	指向位图中色彩级的数值。
bmBitsPixel	指向每一色彩级相邻颜色位的数目，这些色彩级均需定义象素。
bmBits	指向位图位值的位置，此字段必须是指向字符型矩阵的长指针。

注释: 现在通用的位图格式是单色和彩色的，单色位图用一位、一级格式。扫描都是 16 的倍数。

对于高度为 n 的单色位图可按以下方式组织扫描序列:

```
Scan0
Scan1
.
.
.
.
Scan  n-2
Scan  n-1
```

在单色图案中，象素是黑色或者白色的。如果在位图中相应位是1，则象素接通(为白色)；若在位图中相应位是0，则象素关断(为黑色)。

所有有 RC_BITBLT 位集合功能的设备都支持位图。

每种图案都有各自的色彩格式，为了从一种图案变为另一种图案，可调用 GetDIBits 和 SetDIBits。

参阅第一册第四章“函数目录”中的 CreateBitmapIndirect 和 Getobject 函数。

BITMAPCOREHEADER 3.0 (设备无关的位图格式信息)

BITMAPCOREHEADER 结构包括关于设备无关位图的尺寸和颜色格式，这种位映象与 Microsoft OS/2 Presentation Manager 的 1.1 和 1.2 版本相兼容。

```
typedef struct tagBITMAPCOREHEADER{
    DWORD   cbSize;
    WORD    bcWidth;
    WORD    bcHeight;
    WORD    bcPlanes;
    WORD    bcBitCount;
  } BITMAPCOREHEADER;
```

BITMAPCOREHEADER 结构有如下字段:

字段	说明
bcWidth	规定象素中位图宽度。
bcHeight	规定象素中位图高度。
bcPlanes	规定目标设备级数，必须设置为1。
bcBitCount	规定每个象素的位数，这个值必须是1，4，8或24。

注释: BITMAPCOREINFO 数据结构包括 BITMAPCOREHEADER 结构和颜色表以提供关于设备无关位图的尺寸和颜色的完整定义，阅读 BITMAPCOREINFO 数据结构时，请着重看一看有关规定设备无关位图的信息。

一个应用程序可以用存储在 bcSize 域的信息来设置 BITMAPCOREINFO 数据结构的颜色表，例如:

```
pColor=((LPSTR)pBitmapCoreInfo+(WORD)(pBitmapCoreInfo->bcSize)
```

BITMAPCOREINFO 3.0 (设备无关位图信息)

BITMAPCOREINFO 结构完全定义了设备无关位图的尺寸和颜色，这种位映象与 Microsoft OS/2 Presentation Manager1.1 和 1.2 版本相兼容。

```
typedef struct_BITMAPCOREINFO{
        BITMAPCOREHEADER   bmciHeader;
        RGBTRIPLE          bmciColors[];
      }BITMAPCOREINFO;
```

BITMAPCOREINFO 结构包括以下字段:

字段	说明
bmciHeader	规定了一个包含设备无关位图尺寸和颜色格式信息的BITMAPCOREHEADER数据结构。
bmciColors	规定了定义位图颜色的RGBTRIPLE数据结构的阵列。

注释：一个 OS/2 Presentation Manager 设备无关位图包括两个不同的部分：

一个是 BITMAPCORE 数据结构，它描述了位图的大小、颜色，另一个是定义位图象素的字节阵列，这个阵列的位被压缩在一块，但每一个扫描行必须被 0 填充以便在 LONG 边界结束，段边界可以出现在位图的各个地方，但位图的起始位置是左下角。

BITMAPCOREINFO 的 bcBitCount 字段，规定了定义位图的每个象素和最大颜色数的位数，这个字段必须设置为如下各值：

值	含义
1	位图是单色的，而且bmciColors字段须有两个项。位图阵列的每一位表示一个象素，如果位被清零，则象素用bmciColors表中的第一项的颜色表示，如果位被设置，则用表的第二项颜色表示。
4	位图最多有16种颜色，bmciColors字段有16项，每个位图象素在颜色表中由一个四位变址表示。 例如，如果位图中的第一字节是0X1F，则这个字节表示两个象素，第一个元素包含第二个表项的颜色，第二个元素包含第十六个表项的颜色。
8	位图最多有256种颜色，bmciColors字段包括256项。这样，每个阵列中的字节只表示一个象素。
24	位图最多有2^{24}种颜色，bmciColors字段为空，位图阵列中每三个字节表示一个象素的红、绿、蓝的相对亮度。

bmciColors 表的颜色可按其重要次序显示。或者，对于应用设备无关位图的函数，bmciColors 字段可以是 16 位无符号整数的阵列，用以确定当前已实现的逻辑调色板的变址以取代 RGB 显示值，这样，一个应用程序使用位图时，须调用独立设备位图函数，其参数 wUsage 设置为 DIB_PAL_COLORS。

注意：如果位图将存于文件或转换为另一个应用程序，bmciColors 不要含调色板变址，除非应用程序互斥地使用位图且在它的完全控制下，位图可以包含显式的 RGB 值。

BITMAPFILEHEADER 3.0(位图文件信息)

BITMAPFILEHEADER 数据结构包括有关设备无关位图文件的类型、大小和打印格式等信息。BITMAPFILEHEADER 数据结构包括以下字段：

字段	说明
bfType	规定文件的类型，必须是BM。
bfSize	规定文件大小，以DWORD为单位表示。
bfReserved1	保留值，必须设置为0。
bfReserved2	保留值，必须设置为0。
bfOffBits	规定文件中从实际位图的BITMAPFILEHEADER开始的偏移量，以字节为

单位表示。

注释：在 DIB 文件中，BITMAPINFO 或 BITMAPCOREINFO 数据结构紧随 BITMAPFILEHEADER 结构之后。

BITMAPINFO 3.0 (设备无关位图信息)

BITMAPINFO 结构全面定义了一个 Windows 3.0 设备无关位图的尺寸和颜色信息。

```
typedef struct tagBITMAPINFO{
        BITMAPINEOHEADER bmiHeader;
        RGBQUAD          bmiColors[1];
      }BITMAPINFO;
```

BITMAPINFO 结构包括下列字段:

字段	说明
bmiHeader	规定了一个包含设备无关位图的尺寸和颜色格式信息的 BITMAPINFOHEADER数据结构。
bmiColors	规定了定义位图颜色的RGBTRIPLE数据结构的阵列。

注释：一个 Windows 3.0 设备无关位图包括两个不同的部分:

一个是 BITMAPINFO 数据结构，描述了位图的大小和颜色，另一个是定义位映象象素的字节阵列。这个阵列中的位被压缩在一块，但每一个扫描行必须被 0 填充以便在 LONG 边界结束，段边界可以出现在位图的各个地方，但位图的起始位置是左下角。

BITMAPINFOHEADER 结构的 bitBitCount 字段规定了限定位图的每个象素和最大颜色数的位数，这个字段必须设置为以下各值:

值	含义
1	位图是单色的，而且bmciColors字段须有两个项。位图阵列的每一位代表一个象素，如果位被清零，则象素用bmiColors表中的第一项颜色表示，如果位被设置，则用表的第二项颜色表示。
4	位图最多有16种颜色，bmiColors字段含有16个以上表项，每一个位映象的象素在颜色表中由一个四位变址表示。 例如，如果位图的第一个字节是0X1F，那么这个字节表示两个象素。第一个象素包括在第二个表项中的颜色，第二个象素包括第十六个表项中的颜色。
8	位图最多有256种颜色，bmiColors字段包括256个以上的表项。这样，阵列中的每一字节只表示一个象素。
24	位图最多有2^{24}种颜色，bmiColors字段为空，位图阵列中的每三个字节表示每一象素的红、绿、蓝的相对亮度。

BITMAPINFOHEADER 结构的 biClrUsed 字段规定了位图实际使用的颜色表中的变址数，如果 biClrUsed 字段设置为 0，位图使用与 biBitCound 字段的值相对应的最大颜色数目。

bmiColors 表中的颜色可按重要次序显示。

或者，对于使用设备无关位图的函数，bmiColors 字段可以是 16 位无符号的整数阵列，用以确定当前已实现的逻辑调色板的变址取代 RGB 显示值，这样，一个应用程序使

用位图时必须调用设备无关位图函数，其参数 wUsage 设置为 DIB_PAL_COLORS。

注意：如果位图将存于一个文件或转换为另一个应用程序，bmiColors 字段不要含调色地址。除非应用程序互斥地使用位图且在它的完全控制下，位图可以包含显式的 RGB 值。

BITMAPINFOHEADER 3.0 (设备无关位图格式信息)

BITMAPINFOHEADER 结构包括有关 Windows3.0 设备无关位图的尺寸和颜色格式信息。

```
typedef struct tagBITMAPINFOHEADER{
        DWORD biSize;
        DWORD biWidth;
        DWORD biHeight;
        WORD  biPlanes;
        WORD  biBitCount;
        DWORD biCompression;
        DWORD biSizeImage;
        DWORD biXPelsPerMeter;
        DWORD biYDelsPerMeter;
        DWORD biClrUsed;
        DWORD biClrImportant;
      }BITMAPINFOHEADER;
```

BITMAPINFOHEADER 结构包括以下字段：

字段	说明
biSize	规定BITMAPINFOHEADER结构所需的字节数。
biWidth	规定位图宽度，以象素为单位表示。
biHeight	规定位图高度，以象素为单位表示。
biPlanes	规定目标设备级别数，必须设置为1。
biBitCount	规定每个象素的位数，此值须为1，4，8或24。
biCompression	规定一个压缩位图的压缩类型，可以是如下各值：
值	含义
BI_RGB	规定位图不被压缩。
BI_RLE8	规定一个8位象素的位图的运行长度编码格式。这个压缩格式是一个包含有计数字节和颜色变址字节的两字节格式，请阅读后面的“注释”以获得更多的信息。
BI_RLE4	规定一个4位象素的位图的运行长度编码格式。这个压缩格式是一个包含有计数字节和两个字长颜色变址的两字节格式，阅读后面的“注释”以获得更多的信息。
biSizeImage	规定图象的大小以字节为单位表示。
biXPelsPerMeter	规定位图的目标设备的水平分辨率，以每米象素数计算。一个

应用程序可用这个值从资源中选择最佳匹配于当前设备性能的位映象。

biYPelsPerMeter	规定位图的目标设备的垂直分辨率，是以每米象素数来计算的。
biClrUsed	规定位图实际使用的颜色表中的颜色变址数，如果该值为0，则位图使用对应于biBitCount字段值的颜色最大值。参阅本章前面介绍的BIFMAPINFO数据结构以获得有关颜色表最大尺寸的更多信息。如果biClrUsed不为零，且biBitCount字段小于24，则biClrUsed字段规定了绘图仪或设备驱动器所要使用的彩色数。如果biBitCount被设置为24，biClrUsed段规定了用来提高Windows调色板执行优先级的参考颜色表的大小。如果是一个“压缩”位图(也就是说，一个位图中，BITMAPINFO标题后面紧跟着位图矩阵，而且该位图仅供一个指针访问)，那么biClrUsed字段须是0或是颜色表的实际大小。
biClrImportant	规定显示位图过程中，被认为重要的颜色变址数，如果该值为0，则表示所有颜色都重要。

注释：BITMAPINFO 数据结构合并了 BITMAPINFOHEADER 结构和一个提供了有关 Windows 3.0 设备无关位图尺寸和颜色完全限定的颜色表。参阅 BITMAPINFO 数据结构说明以获得更多有关规定 Windows 3.0 设备无关位图的信息。

一个应用程序应使用存储在 biSize 字段的信息来设置 BITMAPINFO 数据结构的颜色表，方法如下：

```
pColor = ((LPSTR)pBitmapInfo + (WORD)(pBitmapInfo->biSize)
```

位图压缩格式

Windows 支持那些 8 位或 4 位象表定义自身颜色的压缩位图格式。压缩减少了位映象要使用的磁盘空间和内存空间。下面的段落说明了这些格式。

当 biCompression 字段设置为 BI_RLE8 时，位图可用 8 位位图的运行长度编码格式压缩，这个格式可用下列两种模式之一压缩：

■编码的

■绝对的

通过单一位图，两种模式可用于任何地方。

编码模式由两个字节组成：第一个字节规定了用第二个字节中的颜色变址画出的连续象素的数目，而且，第一个字节可以置为 0 以指明一行结束、位映象结束或 δ 的转义。转义的含义取决于第二个字节的值，下表表示了第二个字节的含义：

转义的第二个字节	含义
0	行末。
1	位图末端。
2	δ，转义后面的两个字节包含了一个无符号数，用来表示下一个象素从当前位置开始的水平和垂直位移。

绝对模式的第一个字节设置为 0，第二个字节在 03H～FFH 之间。在绝对模式中，第二个字节代表后面的字节数，每个字节都包含一个象素的颜色变址。若第二个字节小于等

于 2，转义的含义与编码模式相同。在绝对模式中，每次操作必须根据字边界对齐。

下面的例子表示了一个 8 位压缩位图的十六进制值:

03 04 05 06 00 03 45 56 67 00 02 78 00 02 05 01
02 78 00 00 09 1E 00 01

这个位图可扩展如下: (二位数值表示一个象素的颜色变址)

04 04 04
06 06 06 06 06 06
45 56 67
78 78
move current position 5 right and 1 down
78 78
end of line
1E 1E 1E 1E 1E 1E 1E 1E 1E
end fo FLE bitmap

当 biCompression 字段设置为 BI_RLE4，使用游程长度编码格式的位图被压缩成一个 4 位位图，该位图也使用编码和绝对模式。在编码模式中，第一个字节包含了用第二个字节的颜色变址画出的象素数目。第二个字节包含两个颜色变址，一个在高四位，另一个在低四位。第一个象素按高四位规定的颜色绘图，第二个象素按低四位规定的颜色绘图。第三个按高四位规定的颜色绘图，如此下去，直到第一个字节中所有的象素都被画出来为止。

在绝对模式中，第一个字节包括 0，第二个字节包括后面的颜色变址数，随后的字节包括高四位和低四位的颜色变址，每个变址表示一个象素。在绝对模式中，每次操作必须根据字边界对齐。行结束、位图结束和 δ 转义也使用 BI_RLE4。

下面的例子给出了一个 4 位压缩位图的十六进制值:

03 04 05 06 00 06 45 56 67 00 04 78 00 02 05 01
04 78 00 00 09 1E 00 01

这个位图可扩展如下(一位数值表示一个象素的颜色变址):

0 4 0
0 6 0 6 0
4 5 5 6 6 7
7 8 7 8
move current position 5 right and 1 down
7 8 7 8
end of line
1 E 1 E 1 E 1 E 1
end of RLE bitmap

CLIENTCREATESTRUCT 3.0 (MDI 客户窗口生成结构)

CLIENTCREATESTRUCT 数据结构包含有关菜单和 MDI 客户窗口的第一个多文件接口(MDI)子窗口的信息。当生成一个 MDI 客户窗口时，一个应用程序用长指针指向该结构

作为 CreateWindow 函数的 lpParam 参数。

```
typedef struct tagCLIENTCREATESTRUCT
    {
      HMENU   hWindowMenu;
      WORD    idFirstChild;
    }CLIENTCREATESTRUCT;
```

CLIENTCREATESTRUCT 结构包括以下字段:

字段	说明
hWindowMenu	应用程序的窗口菜单指针，应用程序可调用GetSubMenu函数从MDI结构窗口菜单中寻回该指针。
idFirstChild	是由第一MDI子窗口生成的子窗口，对于每一个由应用程序生成的客户MDI子窗口，Windows将ID加1，而当应用程序消除一个窗口时，为保持标识符区域连续而进行标识符再分配。当一个子窗口从一个窗口菜单中选出来时，这些标识符将在应用程序的MDI框架的WM_COMMAND消息中使用，不应与任何其它标识符相冲突

COLORREF (颜色规定)

一个 COLORREF 颜色数值是一个规定颜色的长整数。申请颜色的 GDI 函数(例如 CreatePen 和 FloodFill)接收一个 COLOREF 值作为参数。根据应用程序使用 COLORREF 值的方式，该数值有三种不同的的形式。可以是下面的一种:

- 红、绿、蓝的显示值(RGB)
- 逻辑调色板的变址。
- 相对调色板的 RGB 值。

显式 RGB

规定一个显式 RGB 值，COLORREF 值有如下的十六进制形式:

0x00bbggrr

低位字节包含红色相对亮度值; 第二个字节包含绿色值，第三个包含蓝色值。高位字节必须为 0。一个字节最大数为 FF(十六进制)，下表给出了产生相应颜色的十六进制数:

值	颜色
0X000000FF	纯红色
0X0000FF00	纯绿色
0X00FF0000	纯蓝色
0X00000000	黑色
0X00FFFFFF	白色
0X00808080	中灰色

RGB 宏指令接收表示红、绿、蓝颜色的值并返回一个显示 RGB COLORREF 值。

调色变址

当确定一个逻辑调色板变址时，COLORREF 值为如下的十六进制形式：

0x0100iiii

两个低位字节由指明逻辑调色变址的 16 位整数组成。第三个字节不使用须置为 0，第四个字节须置为 1。

例如，十六进制值 0X01000000 规定了 0 变址调色项的颜色，0X0100000C 规定了 12 变址调色项的颜色，以此类推。

PALETTEINDEX 宏指令接收一个整数，该数表示一个逻辑调色变址，并返回一个调色变址 COLORREF 值。

相对调色板的 RGB

当规定一个相对调色板的 RGB 值时，COLORREF 值为如下的十六进制形式：

0x02bbggrr

由于使用显示 RGB，三个低字节包括红、绿、蓝三个颜色值且高位字节须置为 2。

对于支持逻辑调色的输出设备，尽管应用程序已经规定了那个调色项的变址，但 Windows 将把一个相对调色的 RGB 值匹配于设备状态调色中最相近的颜色。如果输出设备不支持系统调色，那么尽管它是一个显式 RGB 值，Windows 也将使用相对调色 RGB。

PALETERGB 宏指令接收一个包含红、绿。蓝颜色值的值并返回一个相对调色的 RGB COLORREF 值。

注释：在把一个调色变址或相对调色 RGB COLORREF 值传递给一个还需求设备状态参数的函数之前，使用自身调色板的应用程序必须将自己的调色板用于设备状态(通过调用 SelectPalette)并实现调色功能(通过调用 RealizePalette)。这样确保了该函数正确使用调色项中的颜色。对于生成目标的函数(如 CreatePen)，应用程序必须在为设备状态选出目标之前选择和实现调色功能。

COMPAREITEMSTRUCT 3.0 (所有者绘制的项排序信息)

COMPAREITEMSTRUCT 结构为存储在有序的所有者绘制的组合框和表框中的两个项提供标识和应用程序数据。

每当应用程序要把一个新项加到具有 CBS_SORT 或 LBS_OSRT 型的一上所有者绘制的组合码框或表框中时，Windows 将把 WM_COMPAREITEM 消息传递给主记录。消息中的 lParam 参数包含一个指向 COMPAREITEMSTRUCT 数据结构的长指针。当主记录收到消息时，它将比较两个项并返回一个指示哪个项在前的值。参阅第一册第六章有关“消息目录”中的 WM_COMPAREITEM 消息说明以获得更多的信息。

```
typedef struct tagCOMPAREITEMSTRUCT{
        WORD CtlType;
        WORD CtlID;
        HWND hwrdItem1:
        WORD itemID1;
        DWORD itemData1;
        WORD itemID2;
```