Windows 中 WMF 文件的格式及使用

张俊国　（吉林省土地管理信息中心　130061）
张紫冬　（吉林省工商银行信息处）

在 Windows 中，有两种常见的图象文件格式，一种是位图文件，即 BMP 文件，用于存储光栅图象。另一种就是图元文件（或称为图片文件），一般资料中对这种格式的文件介绍较少，但是在 Windows 中这也是一种常用的图象存储格式，如 WORD 中就带有许多这种格式的图片文件，即扩展名为 WMF 的文件。使用过 WORD 的人都知道，这种图片使用起来非常灵活方便，可以任意改变大小、比列、位置。而且不会象位图文件那样产生失真，只要了解了这种文件的结构格式，也可以在自己的应用程序中灵活地使用它。

Windows 图元文件是通过使用 Windows 的 GDI 函数集来描述图象的，因此，与位图文件相比，可以占用更少的磁盘及内存空间，并且具有更强的设备无关性，而且存储，再现以及在不同应用之间共享等也十分方便。

Windows 中的图元文件常见的有两种格式：

一种是标准的 Windows 图元文件（METAFILE），这是一种临时性的，一般用于在同一应用程序中显示，绘制以及复制图象。

另一种则称为可放置图元文件（PLACEABLE METAFILE）。Windows 中及其他一些 Windows 应用软件所提供的扩展名是 WMF 的文件，就属于这一种格式。

一、标准的 Windows 图元文件

这种图元文件可以是基于全局内存块的内存图元文件，也可以是磁盘图元文件。

文件包括两个部分：一个文件头结构和记录列表。

1. 文件头结构

这个头结构描述了文件的大小和文件中所用绘图对象（drawing object）的数量，这些绘图对象可以是笔、刷、位图或字体。

文件头结构具有如下形式：

```
typedef struct tagMETAHEADER {
    WORD mtType;
    WORD mtHeaderSize;
    WORD mtVersion;
    DWORD mtSize;
    WORD mtNoObjects;
    DWORD mtMaxRecord;
    WORD mtNoParameters;
} METAHEADER;
```

其中：

mtType　指出图元文件是存储在内存中还是在文件中，等于 0：在内存中；等于 1：在文件中

mtHeaderSize　头结构的字节数。

mtVersion　WINDOWS 版本号，对于 Windows 3.0 及以上为 0x300。

mtSize　文件字节数。

mtNoObjects　能同时存在于文件中的绘图对象的最大数量

mtMaxRecord　文件中记录数。

mtNoParameters　没有使用。

2. 图元文件记录

大多数 GDI 函数都可以作为文件记录。

每个记录有以下格式：

```
struct {
    DWORD rdSize;
    WORD rdFunction;
    WORD rdParm[];
}
```

其中：

rdSize　记录的字节数。

rdFunction　GDI 函数代码。

rdParm 包含函数参数的数组（顺序与函数调用顺序相反）。

3. 图元文件应用函数不多，主要包括以下这些：

CreateMetaFile	创建一个图元文件设备句柄，参数为 NULL 则创建内存图元文件，否则是磁盘图元文件。
CloseMetaFile	关闭一个图元文件设备句柄，并返回一个图元文件句柄。
CopyMetaFile	拷贝图元文件。
DeleteMetaFile	删除图元文件句柄。
EnumMetaFile	枚举图元文件中的记录。
GetMetaFile	创建一个特定图元文件句柄。

GetMetaFileBits 获得存储有图元文件数据的内存句柄。
PlayMetaFile 运行图元文件。
PlayMetaFileRecord 运行图元文件的一个记录。
SetMetaFileBits 根据内存块句柄创建一个图元文件句柄
SetMetaFileBitsBetter 作用同上,主要用于 OLE
一般的使用方法是:

1.用 CreateMetaFile 函数创建图元文件设备句柄,如果参数为 NULL 则创建内存图元文件,否则 是磁盘图元文件,参数是文件名;

2.使用 GDI 函数利用图元文件设备句柄对图元文件设备缓冲区绘图,应当尽量避免使用函数 SetViewOrg 和 SetViewExt,否则生成的图象将不能改变大小和位置;

3.用 CloseMetaFile 关闭图元文件设备句柄,并返回图元文件句柄;

4.利用图元文件句柄,调用 PlayMetaFile 函数就可以在输出设备上随意显示图元文件了,显示 之前,可以利用函数 SetViewOrg 和 SetViewExt 改变图象显示的位置和大小;

5.因为图元文件中的绘图单位没有实际意义,因此显示之前可将映象模式设置为 MM-ANISOTROPIC 或其他想采用的模式;

6.用 DeleteMetaFile 删除图元文件句柄,内存图元文件同时被删除如果创建的是磁盘图元文件,文件不会被自动删除;

7.以后要使用磁盘图元文件时,使用 GetMetaFile 读入图元文件,同时自动创建图元文件句柄。

二、可放置的 Windows 图元文件

这种文件更常见而且有许多现成的可供使用。

这种文件与标准 Windows 图元文件的不同之处是在标准图元文件之前又增加了一个 22 字节的文件头,其结构如下:

```
typedef struct {
DWORD          key;
HANDLE         hmf;
RECT           bbox;
WORD           inch;
DWOR           reserved;
WORD           checksum;
} METAFILEHEADER;
```

其中 :

key 用于确定文件类型的关键字,必须等于 0X9AC6CDD7L,可以由此判断是不是图元文件;

hmf 没有使用,必须等于零;

bbox 包围图元文件所绘图象的矩形的坐标,既图片的外边界坐标,使用文件单位,单位变 换关系根据 inch 值确定;

inch 提供文件单位到英寸的转换值,此值应该小于 1440,以防止溢出,一般取值常在 576～1000 之间,由此值可以获得原始图片的大小;

reseved 没有使用,必须为零;

checksum 和校验值,等于异或(XOR)文件的前十个字节,可以用来检验文件是否正确;

一般来说,文件中图片的原始大小并无实际意义,因此,实际应用中可以只利用结构中的 key 来判断文件类型,以及用 bbox 的值确定图象的宽高比。

Windows 中没有提供使用这种格式文件的 API 函数,图元文件的输入输出需要借助于基本的文件输入输出函数编程实现。

三、使用实例

了解了以上情况,就可以显示和生成这两种 Windows 图元文件了。

下面介绍如何编程显示可放置的 WMF 文件的简单方法,以下是使用 C 语言时的具体方法和步骤。

1.打开文件并读入文件头信息,根据 key 值判断文件类型;

2.取得文件长度,分配全局内存块并读入文件(不包括 22 字节的文件头);

3.使用 SetMetaFileBits 函数创建图元文件(MetaFile)句柄;

4.保存当前窗口(Window)及视口(View)的原点及范围值;

5.设置映象模式,如果要按照原图片的宽高比例显示,设置为 MM-ISOTROPIC,如果想按任意比例,大小显示可以把映象模式设置为 MM-ANISOTROPIC;

6.使用 PlayMetaFile 函数显示图片文件,显示之前,可以利用函数 SetViewOrg 和 SetViewExt 改变图象显示的位置和大小;

7.恢复原映象模式;

8.恢复原窗口及视口的原点和范围值;

具体实现见所附源程序(源程序略),本程序使用 BorlandC++3.1 编程,因本程序仅仅显示图元文件,没有其他内容,因此显示图片前后没有对窗口和视口进行保留和恢复操作,程序运行时,用鼠标左键改变大小,右键改变显示位置。

以上显示图元文件的方法也可以用于打印,只要把相应的显示设备句柄改为打印机设备句柄即可。

(来稿时间:1997 年 2 月)