Large 'PICT's How to display large 'PICT's

Early versions of the Macintosh ROMs don't let you use 'PICT' resources over 32K. There are two ways you can work around this:

- Instead of storing a 'PICT' in a resource, you can store it in a PICT file, a file with a 'PICT' stored in its data fork. The example below shows you how to read in a 'PICT' file.
- If your program will run only on Macintoshes with 128K ROMs (or ROMs produced after those), you can store a large 'PICT' in a resource. In those ROMs, QuickDraw reads the 'PICT' until it reaches the end-of-picture opcode and ignores the size word at the beginning of the 'PICT'. Remember, if you put large PICTs in resources, your program won't run on Macintoshes with 64K ROMs.

Example

```
/*replacement for the QuickDraw bottleneck routine*/
pascal void GetPICTData ( Ptr dataPtr, short byteCount)
{ /* GetPICTData */
    OSErr  err;
    long        longCount;

    longCount = byteCount;
    err = FSRead(globalRef,&longCount,dataPtr);
        /*can't check for an error because we don't know how to handle it*/
} /* GetPICTData */

/*error code if DrawPicture aborted*/
#define    abortPICT  128

OSErr GetDrawPICTFile()          /*read in a 'PICT' selected by the user*/
{   /* GetDrawPICTFile */

    Point          wher;          /*where to display dialog*/
    SFReply        reply;         /*reply record*/
    SFTypeList     myFileTypes;   /*see Standard File */
    short          numFileTypes;
    OSErr          err;
    QDProcsPtr     savedProcs;
    QDProcs        myProcs;       /*use CQDProcs for a color window*/
    PicHandle      myPicture;
                       /*we need a picture handle for DrawPicture*/
    long           longCount,myEOF,filePos;

    wher.h = 20;
    wher.v = 20;
    numFileTypes = 1;              /*display 'PICT's*/
    myFileTypes[0] = 'PICT';
    SFGetFile(wher,'',nil,numFileTypes,myFileTypes,nil,&reply);

    if (reply.good)
    {
        SetStdProcs(&myProcs);
            /*use SetStdCProcs for a CGrafPort*/
```

```
        myProcs.getPicProc = GetPICTData;
        savedProcs = thePort->grafProcs;
            /*set the grafProcs to ours*/
        thePort->grafProcs = &myProcs;

        myPicture = (PicHandle)NewHandle(sizeof(Picture));

        err = FSOpen(&reply.fName,reply.vRefNum,&globalRef);
        if (err != noErr) return err;

        err = GetEOF(globalRef,&myEOF);
            /*get EOF for later check*/
        if (err != noErr) return err;

        err = SetFPos(globalRef,fsFromStart,512);/*skip header*/
        if (err != noErr) return err;

            /*read in the (obsolete) size word and the picFrame*/
        longCount = sizeof(Picture);
        err = FSRead(globalRef,&longCount,(Ptr)*myPicture);
        if (err != noErr) return err;

        DrawPicture(myPicture,&(**myPicture).picFrame);
            /*draw the picture*/

        err = GetFPos(globalRef,&filePos);          /*get position for check*/
        if (err != noErr) return err;
        err = FSClose(globalRef);
        if (err != noErr) return err;

        DisposHandle((Handle)myPicture);

        thePort->grafProcs = savedProcs;    /*restore the procs*/

            /*Check for errors. if there wasn't enough room,*/
            /*DrawPicture will abort; the FILE position mark*/
            /*won't be at the end of the FILE.*/
        if (filePos != myEOF) return abortPICT;
        else return noErr;
    } /*if (reply.good) */
}    /* GetDrawPICTFile */
```