Macintosh: Determining Video Card Status via Software (1 of 2)

Revised: 12/4/89
Security: Everyone

Macintosh: Determining Video Card Status via Software (1 of 2)

==========

Article Created: 2 November 1989
Article Last Reviewed: 14 July 1992
Article Last Updated:

TOPIC ----------

How can an application detect information about the status of the video display system connected to a Macintosh? For example, how many bits per pixel are selected via the Monitors cdev, and how many bits per pixel is supported by the card in the system? Inside Macintosh Volume V mentions using some high-level Color QuickDraw call. How is this implemented?

DISCUSSION ----------

Refer to the QuickDraw chapters of "Inside Macintosh, Volume V." Page 124 documents a function called "GetMainDevice". This function returns a GDHandle (Graphics Device Handle). This handle has the information you want. Here is its description (page 119):

```
TYPE
   GDHandle = ^GDPtr;
   GDPtr    = ^GDevice;
   GDevice  = RECORD
                gdRefNum:     INTEGER;
                gdID:         INTEGER;
                gdType:       INTEGER;
                gdITable:     ITabHandle;
                gdResPref:    INTEGER;
                gdSearchProc: SProcHndl;
                gdCompProc:   CProcHndl;
                gdFlags:      INTEGER;
                gdPMap:       PixMapHandle;
                gdRefCon:     LONGINT;
                gdNextGD:     GDHandle;
                gdRect:       Rect;
```

```
                gdMode:       LONGINT;
                gdCCBytes:    INTEGER;
                gdCCDepth:    INTEGER;
                gdCCXData:    Handle;
                gdCCXMask:    Handle;
                gdReserved:   LONGINT;
              END;
```

The field in the above Pascal record with is the "gdPMap" field. The "gdPMap" field contains a PixMapHandle. Here is a description of the handle (pages 52 and 53):

```
TYPE
   PixMapHandle = ^PixMapPtr;
   PixMapPtr    = ^PixMap;
   PixMap       = RECORD
                    baseAddr:   Ptr;
                    rowBytes:   INTEGER;
                    bounds:     Rect;
                    pmVersion:  INTEGER;
                    packType:   INTEGER;
                    packSize:   LONGINT;
                    hRes:       Fixed;
                    vRes:       Fixed;
                    pixelType:  INTEGER;
                    pixelSize:  INTEGER;
                    cmpCount:   INTEGER;
                    cmpSize:    INTEGER;
                    planeBytes: LONGINT;
                    pmTable:    CTabHandle;
                    pmReserved: LONGINT;
                  END;
```

This "PixMap" record has a field called "pixelSize" that contains the current depth, in bits, of the pixels for the graphic device.

The Pascal code to retrieve this information looks something like this:

```
 PROCEDURE GetBitsPerPixel;

  VAR
   mainDevice: GDHandle;
   theDepth: INTEGER;

 BEGIN
  mainDevice := GetMainDevice;
  theDepth := mainDevice^^.gdPMap^^.pixelSize;
 END;
```

If more than one video card is installed in the system, and you need to know what the bits per pixel setting is on all devices, not just on the main device, try the following functions (page 124):

GetDeviceList
GetNextDevice

Here is a sample procedure that shows how to use them:

```
PROCEDURE GetGraphicDevs;

  VAR
   DeviceList: ARRAY[1..6] OF GDHandle;
   theDepthList: ARRAY[1..6] OF INTEGER;
   x: INTEGER;

 BEGIN
  DeviceList[1] := GetDeviceList;
  theDepthList[1] := DeviceList[1]^^.gdPMap^^.pixelSize;
  FOR x := 2 TO 6 DO
   BEGIN
    DeviceList[x] := GetNextDevice(DeviceList[x - 1]);
    IF DeviceList[x] <> NIL THEN
     theDepthList[x] := DeviceList[x]^^.gdPMap^^.pixelSize
    ELSE
     x := 6;
   END;
 END;
```

If you need to find which device in the device list array created above is the main device, use the "TestDeviceAttribute" function described on page 124. Here is a code fragment showing how to use the function:

```
IF TestDeviceAttribute(DeviceList[1], mainScreen) THEN...
```

If the "TestDeviceAttribute" function returned "TRUE" or "1" in the example above, then "DeviceList[1]" is the main screen. If the function returned "FALSE" or "0", then the device is not the main screen. You can also use this function to determine if the device is in color model. For example:

```
IF TestDeviceAttribute(DeviceList[1], gdDevType) THEN...
```

If the "TestDeviceAttribute" function in this example returned "TRUE" or "1", then "DeviceList[1]" is in color mode; if it returned "FALSE" or "0", then it is in monochrome mode.

Copyright 1989, Apple Computer, Inc.

Tech Info Library Article Number:4725