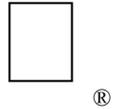


# New Technical Notes

## Macintosh



### Developer Support

## Finding Drivers in the Unit Table Devices

Revised by:  
Written by: Rick Blair

March 1988  
February 1986

This note will explain how code can be written to determine the reference number of a previously installed driver when only the name is known.

You should already be familiar with The Device Manager chapter of *Inside Macintosh* before reading this technical note.

The Pascal code at the end of this note demonstrates how to obtain the reference number of a driver that has been installed in the Unit Table. The reference number may then be used in subsequent calls to the Device Manager such as `Open`, `Control` and `Prime`.

One thing to note is that the `dRAMBased` bit really only tells you whether `dCtlDriver` is a pointer or a handle, not necessarily whether the driver is in ROM or RAM. SCSI drivers, for instance, are in RAM but not relocatable; their DCE entries contain pointers to them.

From MPW Pascal:

```

PROCEDURE GetDrvrRefNum(driverName: Str255; VAR drvrRefNum: INTEGER);

TYPE
    WordPtr      = ^INTEGER;

CONST
    UTableBase   = $11C;      {low memory globals}
    UnitNtryCnt  = $1D2;

    dRAMBased    = 6;        {bit in dCtlFlags that indicates ROM/RAM}
    drvrName     = $12;      {length byte and name of driver [string]}

VAR
    negCount     : INTEGER;
    DCEH         : DCtlHandle;
    drivePtr     : Ptr;
    s            : Str255;

BEGIN
    UpString(driverName, FALSE); {force same case for compare}

    negCount := - WordPtr(UnitNtryCnt)^; {get -(table size)}

```

```
{Check to see that driver is installed, obtain refNum.}
{Assumes that an Open was done previously -- probably by an INIT.}
{Driver doesn't have to be open now, though.}

drvRefNum := - 12 + 1; {we'll start with driver refnum = -12,
                       right after .ATP entry}

{Look through unit table until the driver found or reach the end.}

REPEAT
  drvRefNum := drvRefNum - 1; {bump to next refnum}
  DCEH := GetDctlEntry(drvRefNum); {get handle to DCE}

  s := ''; {no driver, no name}

  IF DCEH <> NIL THEN
    WITH DCEH^ DO BEGIN {this is safe -- no chance of heap
                        moving before dCtlFlags/dCtlDriver
                        references}

      IF (dCtlDriver <> NIL) THEN BEGIN
        IF BTST(dCtlFlags, dRAMBased) THEN
          drivePtr := Handle(dCtlDriver)^ {zee deréference}
        ELSE
          drivePtr := Ptr(dCtlDriver);

        IF drivePtr <> NIL THEN BEGIN
          s := StringPtr(ORD4(drivePtr) + drvName)^;
          UprString(s,FALSE); {force same case for compare}
        END;
      END; {IF}
    END; {WITH}
  UNTIL (s = driverName) OR (drvRefNum = negCount);

  {Loop until we find it or we've just looked at the last slot.}

  IF s <> driverName THEN drvRefNum := 0; {can't find driver}
END;
```

From MPW C:

```
short      GetDrvRefNum(driverName)
char       *driverName[256];

{ /* GetDrvRefNum */

#define      UnitNtryCnt  0x1d2

/*bit in dCtlFlags that indicates ROM/RAM*/
#define      dRAMBased    6
/*length byte and name of driver [string]*/
#define      drvName      0x12

short      negCount,dRef;
DctlHandle DCEH;
char       *drivePtr,*s;

negCount = -(short *) (UnitNtryCnt); /*get -(table size)*/
```

```
/*Check to see that driver is installed, obtain refNum.*/
/*Assumes that an Open was done previously -- probably by an INIT.*/
/*Driver doesn't have to be open now, though.*/

dRef = -12 + 1; /*we'll start with driver refnum == -12,
               right after .ATP entry*/

/*Look through unit table until we find driver or reach the end.*/

do {
    dRef -= 1; /*bump to next refnum*/
    DCEH = GetDctlEntry(dRef); /*get handle to DCE*/

    s = "";

    if ((DCEH != nil) && ( (**DCEH).dCtlDriver != nil) )
    {
        if (((**DCEH).dCtlFlags >> dRAMBased) & 1)
            /* test dRamBased bit */
            drivePtr = *(Handle) (**DCEH).dCtlDriver;
            /*zee deréference*/
        else
            drivePtr = (**DCEH).dCtlDriver;

        if (drivePtr != nil)
            s = drivePtr + drvrName;
    }
} while (EqualString(s,driverName,0,0) && (dRef != negCount));
/*Loop until we find it or we've just looked at the last slot.*/

if (EqualString(s,driverName,0,0))
    return dRef;
else
    return 0; /*can't find driver*/
}/* GetDrvRefNum */
```

That's all there is to locating a driver and picking up the reference number.

**Further Reference:**

---

- The Device Manager