

```

***** W I N D A P . P A S *****
-----
Task      : Determines whether Windows is active, and if so,
            the operating mode.
-----
Author     : Michael Tischer
Developed on : 08/22/91
Last update  : 01/13/92
*****

uses Dos;                                     { Use DOS unit }

const MULTIPLEX = $2F;      { Interrupt number of multiplex interrupt }
      NO_WIN     = $00;      { Windows not active }
      W_386_X    = $01;      { Windows/386 V2.X running }
      W_REAL     = $81;      { Windows running in real mode }
      W_STANDARD = $82;      { Windows running in standard mode }
      W_ENHANCED = $83;      { Windows running in enhanced mode }

*****
{ ***** Determines whether Windows is active ***** }
{ Input   : MVERSION = Integer variable of main version number }
{          : SVERSION = Integer variable of sub version number }
{ Output  : Windows status, from constants NO_WIN, W_386_X, W_REAL, }
{          : W_STANDARD or W_ENHANCED }
{ Info    : Version number can only be passed and returned when }
{          : Windows 3.x is operating in enhanced mode }
{ ***** }

function windows( var MVersion, SVersion : integer ) : integer;

var regs : registers;      { Processor registers for interrupt call }
    Erg : integer;

-- This function replaces intr( $2F, Regs ) with Regs.ax = $1600 ----
-- (installation test for enhanced mode), as the Pascal function ----
-- returns false values ----

function int2fcall : integer;

begin
  inline( $b8 / $00 / $16 /      { mov ax,1600h }
          $cd / $2f /           { int 2Fh }
          $89 / $46 / $FE );    { mov [bp-2], ax }
  { This inline inserts the "mov ax, [bp-2]" instruction, which }
  { places the local function variable in the return register }
end;

begin
  MVersion := 0;      { Initialize version numbers }
  SVersion := 0;

  -- Windows x.y in enhanced mode -----

  erg := int2fcall; { Installation test for enhanced mode }

  case ( lo(Erg) ) of
    $01,
    $FF: begin
      MVersion := 2;      { main version }
      SVersion := 0;      { sub version unknown }
      Windows := W_386_X;
    end;
    $00,
    $80: begin
      regs.ax := $4680; { Identify real mode or standard mode }
      intr( MULTIPLEX, regs );
      if ( regs.al = $80 ) then
        Windows := NO_WIN      { Windows not running }
      else
        begin
          -- Windows in real more or standard mode -----
          regs.ax := $1605; { Emulate installation of DOS extdr }
          regs.bx := $0000;
          regs.si := $0000;
          regs.cx := $0000;
          regs.es := $0000;
          regs.ds := $0000;
          regs.dx := $0001;
          intr( MULTIPLEX, regs );
          if ( regs.cx = $0000 ) then
            begin
              -- Windows in real mode -----

```

```

        regs.ax := $1606;
        intr( MULTIPLEX, regs );
        Windows := W_REAL;
    end
    else
        Windows := W_STANDARD;
    end;
end;

{-- Windows in extended mode, AX contains version number -----}

else
begin
    MVersion := lo(Erg);           { Display Windows version }
    SVersion := hi(Erg);
    Windows := W_ENHANCED;        { Windows in enhanced mode }
end;
end;
end;

{*****}
{      M A I N   P R O G R A M      }
{*****}

var WindowsActive,                { Windows mode }
    MVer,                        { Main version of Windows }
    SVer      : integer;        { Sub version of Windows }

begin
    writeln( 'ÛÛÛÛÛÛÛÛÛ WINDAP  -  (c) 1991 by Michael Tischer ÛÛÛÛ' );
    writeln;
    WindowsActive := windows( MVer, SVer );
    case ( WindowsActive ) of
        NO_WIN:      writeln( 'Windows not active ' );
        W_REAL:      writeln( 'Windows in real mode ' );
        W_STANDARD:  writeln( 'Windows active in standard mode' );
        W_386_X:     writeln( 'Windows/386 V 2.x active' );
        W_ENHANCED:  writeln( 'Windows V ', Mver, '.', SVer,
                               ' active im enhanced mode' );
    end;
    halt( WindowsActive );
end.

```