

WIN_VER : Programs to display the Windows version.

Copyright © 1994-1995, UK, John M.Howells

This program is freeware, and may be distributed without restriction for non-profit purposes, provided the source code, including the source code of any modified version, is included in any distribution, and that no charge is made, other than a nominal fee to cover the cost of such distribution.

Each of the two versions of the program will report the version of Windows it is running on. The single C source file builds into two versions:

- 1) WIN_V16 : A 16-bit version of the program that reports either the version of Windows 3.xx (16-bit), or that it is running on Windows NT in the Windows on Windows (WOW) emulation, in which case it will report both the version as reported by WOW and the version returned by the underlying Windows NT and whether it is Windows NT Workstation or Windows NT Server and the Service Pack level. If running on Windows 3.xx (16-bit) and Win32s is installed it will report the Win32s version and build.
- 2) WIN_V32 : A 32-bit version of the program that reports either the version and build of Windows NT or Chicago, or that it is running on Win32s if 16-bit Windows is the host and Win32s is present. If running on Win32s it may request that the 16-bit version be run to report the Win32s version and build, if it cannot determine the Win32s version itself. On Windows NT it reports whether it is the Workstation version or a Server version.

When it is run each version displays a dialog box displaying the information. There is no menu, the dialog box is the main window, and the only option is the 'OK' button on the dialog box that closes the program.

You should create a directory called C_DIR\WIN_VER in the root directory of the C: drive and extract the ZIP file using the -d parameter. You can, of course, extract into a different directory, but you may then have to modify the directories in the build files.

The programs can be built using the Microsoft Visual C++ packages using the same source files in each case. There are two make files, WIN_V16.MAK for the Visual C++ 1.51 16-bit version and WIN_V32.MAK for the Visual C++ 2.00 32-bit version, in each case built in its own sub-directory using the relevant Visual C++ package. The EXE files in the distribution are built using the Microsoft Windows hosted tools. If you wish to rebuild the software the MAK files may, of course, have to be updated to reflect the directories used for the build on your target machine.

The programs can also be built using the Borland C++ 4.5 tools. A Multi-Target project IDE file for this purpose is included, and the directory structure this assumes is created. You may need to modify some of the directory entries for the location of items such as Library files. Both programs use the static versions of the run-time libraries, but may be changed to use the dynamic libraries. The EXE files built with the Borland tools are not included, but the distribution does create the two empty directories required to produce the Borland builds.

Because of the slight differences in the way the Microsoft and Borland tools operate a copy of each ICO file, one for 16-bit and one for 32-bit, is created in each of two appropriate directories.

John M.Howells, Wigan, UK
CIS ID: 100031,353
15th February, 1995