

This is the Official set of technical criteria for the Windows 95 Logo. It is published and maintained by the Microsoft Windows Logo Department. These criteria will be updated from time to time as needed to clarify frequent questions on the interpretation of these criteria. The most up-to-date version of these criteria will be kept on the WinNews forums on CompuServe, America OnLine, Genie, ftp.microsoft.com, Prodigy, and available by Phone-Fax service at (206) 635-2222 as Doc #136. Logo questions may be emailed to "winlogo@microsoft.com".

There are 3 categories of products addressed in this document:

Section A: Software	<i>page 1</i>
Section B: Computer Systems (PCs)	<i>page 3</i>
Section C: Hardware Peripherals	<i>page 3</i>

Section A: Software

This discussion is divided into 4 categories:

- a) Non file-based applications and applications that run exclusively in full-screen mode*
- b) File-based applications**
- c) Utilities (e.g., virus scanners, disk management)
- d) Development tools (e.g., compilers, linkers)
- e)
- f) **A file-based application is defined as one that provides Open, Save, and Close File menu options
- g) *An application that runs exclusively in full-screen mode is one that cannot e.g., be windowed or minimized. Generally, this refers to applications without min/max buttons or those that use the full-screen APIs.
- h) Further definition of the requirements below may be obtained by reviewing the article, Diving into the Requirements for the Windows 95 Logo¹.
- i)

A. General requirements

To qualify for the Windows 95 Logo, an application must meet the appropriate requirements listed below. These first five requirements apply to all applications.

1. An application must be a Win32® application programming interface executable, compiled with a 32-bit compiler that generates an executable of the PE (Portable Executable) format. OR if your application is not represented in PE format [interpreted code, for example], then the "runtime engine" must be a Win32 executable in the PE format. [e.g., if you develop an application in Microsoft Access, your application is an .mdb file, not an .exe, but access.exe would need to be a Win32 PE format executable.]
2. An application needs Windows 95 UI/Shell support. Following the *Microsoft Windows User Interface Design Guide*, version 4.0², this means:
 - It must register 16x16 and 32x32 pixel icons for each file type and the application.
 - Using the Windows 95 look, as described in the UI Design Guide, version 4.0 is recommended. [The intent is not for all applications to look the same, but that applications with the Windows 3.1 "look" be updated]. Using system-supplied dialogs and controls is also recommended, but not required.
 - It must use the system metrics for sizing.
 - It is recommended that your application use the system colors.

¹ This article may be obtained through Microsoft's Developer Solutions Phone-Fax Service. You can reach the Phone-Fax Service directly at (206) 635-2222. The document you are reading is #136. The article referenced is #131. Request an index of available documents to view a list of other information that is available on the Windows Logo and other topics. These documents are also available on CompuServe in the WinNews forum.

²The UI Design Guide is available through subscription to MSDN [Microsoft Developer Network]. To join the Microsoft Developer Network in the U.S. and Canada, call (800) 759-5474. In Europe, call +31 10 258 88 64. Everywhere else, call (402) 691-0173 for local contact information.

- Using the right mouse button for context menus (and not using the right mouse button for anything else) is recommended.
 - Follow Windows 95 application setup guidelines to make the application properly visible in the shell. At a minimum, this means that you use the registry, not add information to win.ini or system.ini, and provide complete uninstall capability with your application. In addition, the installation process must be automated for the end-customer.
3. It must be tested on Windows NT 3.5 [or the latest version of Windows NT], and, if it uses Windows 95-specific application programming interfaces (APIs), then the Windows 95-only functionality must degrade gracefully on Windows NT. Conversely, if it uses Windows NT-specific APIs, then the Windows NT-only functionality must degrade gracefully on Windows 95. The product must run successfully on both Windows 95 and Windows NT, unless architectural differences between the 2 OSs prevents it.
 - 4. It must use long filenames; i.e., your application must 1) support long filenames; 2) use long filenames for displaying all document and data filenames in the shell, in title bars, in dialogs and controls, and with icons; and 3) hiding the .XXX extension names in the application itself is strongly recommended.
 - 5. Plug and Play event-awareness is recommended, but not required. Some examples of this are being aware of slow links, and reacting to system messages about the insertion or removal of new devices.

B. For file-based applications...

The next three requirements apply to those applications that deal with files (those that provide Open, Save, and Close File menu options) and don't run in full-screen mode. Some game and children's software run exclusively in full-screen mode, and would *would not need to follow* these three requirements

6. It must support Universal Naming Conventions (UNC) pathnames.
7. It must have OLE 2.0 Container and/or Object support and OLE 2.0 drag-and-drop support. It is also recommended (but not required) that it have OLE 2.0 compound file support with (document summary information included) and OLE 2.0 Automation support.
8. It needs to support simple-mail-enabling using MAPI or Common Messaging Call (CMC) API; i.e., it must include Send Mail functionality [e.g. a Send Mail command in the File menu].

C. Modified requirements for utilities such as disk optimizers and anti-virus software.

9. Same as #1, above, with the exception of components that must use Exclusive Volume Locking API, soft interrupts, or components that must talk directly to 16-bit drivers. UI and other components of these applications must be 32-bit and use the Windows 95 thinking mechanism to access these 16-bit components.
10. Same as #2, above.
11. Same as #3, above, with the exception of products such as disk utilities that implement platform-specific functionality that does not make sense on Windows NT 3.5.
12. Same as #4 above
13. #5-8 are recommended but not required, though #6 is required if your product accesses network resources.

D. Modified requirements for compilers and other development tools.

14. In addition to the requirements listed below, if Windows is one of the compiler or development tool's target platforms, then it must be capable of generating applications that can meet all of the Windows 95 Logo requirements.
15. Same as #1.
16. Same as #2, above, with the exception that on registering icons for each file type and the application, common source file extensions like .c, .cpp, .h, and .hpp are excluded from the requirement.
17. Same as #3, above.
18. Same as #4, above.
19. Same as #5, above.
20. Same as #6, above.
21. It must support OLE in the following ways:
 - OLE drag-and-drop support is recommended within the tool's design environment
 - OLE Automation support (recommended but not required).
 - It must provide an easy point-and-shoot way (commonly known as wizards or experts) of creating applications with OLE Container and/or Object support, or provide this functionality by default.

13. #8 is recommended, but not required.

Section B: PCs [Personal Computer Systems]

For a PC to qualify for the Windows 95 Logo, it must meet a minimum set of requirements as outlined below, and pass the System Compatibility Test (SCT) for Windows 95. Further details are discussed in the [Hardware Design Guide for Windows 95](#)³, though **wherever there are discrepancies between the Design Guide and this document, this document prevails for Logo purposes**. The SCT tests are included in the Windows 95 DDK, along with instructions for OEM participation. System testing is OEM-administered, and results are sent to MCL [Microsoft Compatibility Labs]. Requirements include:

1. An 80386 architecture (or compatible) CPU [80486-33 or better is highly recommended]
2. 4MB RAM [8 MB recommended]
3. Plug and Play BIOS version 1.0a or later* which reads back all resources [BIOS which soft-sets all resources is recommended]
4. Molded-in or permanently printed icon labels on the computer case for built-in ports - if the same icons appear on the cable connectors, matching of icons will be easier
5. Option ROMs on expansion cards must use the Plug and Play header format documented in the Plug and Play BIOS specification
6. System display adapter must use a packed-pixel frame buffer with at least VGA 640x480x8bpp, color for desktops and 64 gray scale for mobiles [VGA 1024x768x8bpp is recommended for desktops, and 64 color for mobiles]
7. At least one parallel port that supports IEEE-P1284-I mode protocols for compatibility mode and nibble mode - the system must be capable of receiving the parallel device's identifier in nibble mode [ECP P1284-I recommended]
8. One integrated or separate serial port, with 1-16550A required for mobiles [1-16550A recommended for desktops; also additional PS/2 style port and pen devices with barrel button recommended; and serial infrared devices meeting IrDA spec are recommended]
9. APM 1.1 is required for mobile systems [recommended for desktops]
10. If the system ships with expansion cards or peripheral devices integrated onto the motherboard, it is recommended that the cards/devices meet the Logo specifications defined in this document, and use 32-bit Windows 95-based device drivers.

Section C: Hardware Peripheral Devices

In order for a device to qualify for the Windows 95 Logo, it must meet the requirements described in the [Hardware Design Guide for Windows 95](#)³, also detailed here, and pass the compatibility tests conducted by Microsoft Compatibility Labs (MCL). **Wherever there are discrepancies between the Design Guide and this document, this document prevails for Logo purposes**. Pre-qualifying test tools and MCL device/driver submission details will be included in the 2nd Windows 95 beta DDK release⁴. Detailed Windows 95-based device driver design information can be found in the current DDK. Detailed hardware design information can be found in the [Hardware Design Guide for Windows 95](#). In general, requirements include:

³The [Hardware Design Guide for Windows 95](#) is published by MS-Press, and is available for purchase at most book stores

* The Plug and Play Specification for Windows 95 is available on CompuServe in the PlugPlay forum.

⁴ The easiest way to get the latest documents and information on Windows 95 is through subscription to MSDN [Microsoft Developer Network]. To join the Microsoft Developer Network in the U.S. and Canada, call 800-759-5474. In Europe, call +31 10 258 88 64. Everywhere else, call (402) 691-0173 for local contact information.

Plug and Play Device Drivers

In order to carry the Windows 95 Logo, device drivers are required to support Plug and Play in Windows 95 as follows:

1. Retrieves configuration information from Configuration Manager
2. Is dynamically loadable
3. Is dynamically reconfigurable
4. Reacts to system messages about the insertion or removal of new devices

An ideal Windows 95-based Plug and Play driver has the following capabilities:

1. Requires minimal user interaction to select the proper driver
2. May need to understand state information; that is, the settings for the device may need to change based on which user is logged in, whether the machine is docked or undocked (in the case of a docking station architecture) or both.

Hardware specifications are divided into the following categories:

- j) Display Adapters
- k) Audio Adapters
- l) Storage Devices
- m) Parallel Port Devices [Printers]
- n) External Communications Devices
- o) Modems
- p) Network Adapters
- q) Note on Keyboards and Mice

i. Display Adapters

1. Support the VGA graphics standard
2. Support at least a 640x480x8bpp display driver - desktops must have capability of displaying at least 256 colors; mobiles must support the 8bpp driver and map colors into at least a 64 gray scale display so that changes to higher-resolution external monitors can be made without restarting Windows 95.
3. Use a packed-pixel frame buffer with at least 8 bits per pixel (bpp)
4. The VGA BIOS, if it exists separately, must have its base address fixed at C000h [an alternate address is recommended]
5. The standard VGA page frame and I/O address resources can be static (i.e., not relocatable)
6. All non-VGA standard display resources (also known as “extended” resources, such as register sets and so on) must provide at least one alternate configuration in case of conflict during IPL boot - the VGA BIOS must know how to use alternate configuration register addresses.
7. Capable of being disabled if a conflicting VGA expansion card is added to the system
8. Support the VESA ergonomic timings
9. On ISA Plug and Play display adapter expansion cards used as a system boot device, the display adapter circuitry should come up active when power is turned on or the system is reset

ii. Audio Adapters

1. Capability to produce 22 kHz, 8-bit, monaural, output-only sound as minimum performance
2. Support of either SoundBlaster or Microsoft Windows Sound System compatibility to use built-in drivers for Windows 95
3. Use of a 1/8” mini-phone jack wired for stereo as the output connector
4. Mapping of the base I/O address to configurations compatible with either SoundBlaster or the Microsoft Sound System
5. Support of at least all IRQ signals used either by SoundBlaster or the Microsoft Windows Sound System
6. Selection of at least three available DMA channels, either 8-bit or 16-bit channels, if DMA is supported
7. Capability of being disabled in case of resource conflicts with other devices

iii. Storage Devices

iii.a. Floppy Disk Controllers

1. Use of at least three static I/O addresses: 3F2h, 3F4h, and 3F5h
2. Support of IRQ6
3. Support of at least DMA 2, if DMA is used - in addition, the controller should be capable of selecting at least two other available DMA channels, either 8-bit or 16-bit

4. Capability of being independently disabled

iii.b. IDE/ATA Adapters

1. Use of the first device attached to the adapter as the boot device
2. Use of the standard I/O addresses 1F0h through 1F7h and 3F6h
3. Support at least IRQ14
4. Capability to be disabled if an ATA (IDE) expansion card is added to the system - in addition, if a single adapter card contains a floppy disk drive controller, the adapter must be capable of independently disabling the floppy drive controller if a conflict occurs

iii. c. IDE/ATA Peripherals

1. Support the ATA Packet Interface protocol for CD-ROMs defined in SFF-8020, version 1.2.
2. Are compliant with requirements specified in the ATA - 2 specification.
3. Set the Signature after an ATA Read or ATA Identify Command is received.
4. Implement the SEEK command and set the DSC bit when the ATAPI seek is complete, but do not change the drive select bit.
5. Return CANNOT READ MEDIUM - INCOMPATIBLE FORMAT additional sense code qualifier when a READ is received on an audio track.
6. Support CD-DA.
7. Support the READ_CD command sector types mode 2 form 1, mode 2 form 2, mode 1 form 1, and mode 1 form 2.
8. Support the Test_Unit_Ready command.

iii.d. SCSI Host Adapters

1. Meeting the standards described in the current version of the Plug and Play SCSI Specification*
2. Support of the SCSI Configured Auto-Magically (SCAM) Level 1 protocol for automatic SCSI identifier assignment
3. External SCSI peripheral subsystems must use the 50-pin, high-density shielded device connector defined in the SCSI-2 Standard
4. Selection of at least three available DMA channels, either 8-bit or 16-bit, if DMA is supported
5. Capability of being disabled in the event of resource conflicts with other devices
6. Support automatic switchable termination for PnP operation of internal, external or mixed SCSI configurations.

iii.e. SCSI Devices

1. Meeting the standards described in the current version of the Plug and Play SCSI Specification*
2. Support of the SCSI Configured Auto-Magically (SCAM) Level 1 protocol for automatic SCSI identifier assignment
3. External SCSI peripheral subsystems must use the 50-pin, high-density shielded device connector defined in the SCSI-2 Standard
4. Use of the drivers and receivers that meet the specifications defined in the single-ended alternative of the SPI
5. Cables must conform to the cable requirements defined in clause 6 of the SPI specification
6. External SCSI peripherals must contain two connectors for the SCSI cable: a SCSI in connector and a SCSI out connector - the last peripheral in the chain uses a terminator on the SCSI out connector
7. Attachment of a permanent terminator to the end of the cable, for internal SCSI peripherals
8. Internal SCSI peripherals must not terminate the SCSI bus
9. Terminations must conform to the terminator requirements in the SPI specification over the TERMPWR voltage range of 4.0 to 5.25 VDC
10. Terminators must be powered from the TERMPWR line on the SCSI bus
11. Provision of overcurrent protection for the TERMPWR line or lines
12. Only terminators can draw power from TERMPWR
13. Implementation of the SCSI Bus Parity signal defined in the SCSI-2 Specifications.

* The Plug and Play Specification for Windows 95 is available on CompuServe in the PlugPlay forum.

iv. Parallel Port Devices [Printers]

1. Meeting the standards described in the current version of the Plug and Play Parallel Port Device Specification*
2. Compliant with IEEE P1284-I
3. Support of the Compatibility and Nibble mode protocols to read the device ID from the peripheral

v. External Communications Devices

1. Ability to identify themselves using the identification method described in the Plug and Play External COM Device Specification*

vi. Modems

1. Support at least 9600 bps V.32 with V42/V42bis protocol for data modems
2. Support the TIA-602 (Hayes-compatible) AT command set, with extensions for flow control, V.42/V.42bis
3. Support fax capabilities of at least 9600 bps V.29 with class 1 (TIA-578A)
4. Support Plug and Play device identification, using the appropriate Plug and Play specification (e.g. ISA bus, COM port, PCMCIA slot, LPT port).
5. 16550A compatible UART interface

vii. Network Adapters

1. Support of the NDIS 3.1 network device driver, which supports dynamically starting and stopping the network card
2. Provide a means of automatically enabling the adapter as a boot device or enabling the adapter as a non-bootable device, if the network adapter is designed with RIPL capability
3. No hooking of INT 18 and INT 19 on ISA bus systems - this is a requirement for an ISA Plug and Play card
4. Support of at least seven IRQ signals and enable/disable
5. Selection of at least three available DMA channels, either 8-bit or 16-bit, if DMA is supported
6. Capability of being disabled in case of resource conflicts with other devices

viii. Note on Keyboards and Mice

Specifications for Keyboards and Mice are not yet available, but will be by March, 1995.

* The Plug and Play Specification for Windows 95 is available on CompuServe in the PlugPlay forum