



National Center on Accessible Information Technology in Education

---

## How does accessibility differ across operating systems?

On any computer, the operating system (OS) is the set of programs that performs basic tasks that are necessary for the computer to be functional. The OS provides a software platform on top of which application programs can run. Early operating systems, including Microsoft's Disk Operating System (DOS) and Unix (originally developed by Bell Labs; now an open specification licensed by The Open Group), posed few accessibility barriers for users of assistive technologies because they were text-based. However, as operating systems became increasingly graphics-oriented, accessibility barriers increased for individuals who couldn't see the graphics or use a mouse.

In education, the most common graphic operating systems today are Microsoft's Windows and Apple's Mac OS. Unix and Linux (an open source UNIX-like operating system) are also widely used, particularly in higher education. The latter two operating systems both have strong command-based traditions that linger today, and assistive technologies that support their command-line interfaces have long been available. However, a variety of graphic desktop environments have been developed for these operating

systems and, consequently, have erected barriers for non-mousers and non-sighted users.

One role of the graphic operating system is to provide an application program interface (API) to programmers so they can write applications consistent with the operating environment. All programs developed using a common API will have a similar interface, which makes applications easier to learn and use. The API provides a set of building blocks, which programmers assemble into an application. It is important that the API provide support for accessibility. For example, all menus and controls in a graphic user interface should be accessible via keyboard, not just mouse, and should be displayed with a font and color scheme that can easily be customized by the user. As long as the API provides the means for delivering these and other accessibility features, applications within that environment can easily be made accessible by software application developers.

To date, there has been a significant disparity in the accessibility of operating systems' APIs. Microsoft addressed many of the accessibility problems of its

Windows API fairly early on and provided developers with the tools to develop applications that were accessible. Most Windows applications, for example, are entirely operable via keyboard (i.e., a mouse isn't required). Other graphic operating systems have failed to deliver comparable accessibility, particularly for non-mousing and/or non-visual users. Microsoft also developed a standard by which applications could effectively communicate with assistive technologies. This standard, called Microsoft® Active Accessibility® (MSAA), has been available since Windows 95. (For more information, see the AccessIT Knowledge Base article [What is MSAA?](http://www.washington.edu/accessit/articles?147) (<http://www.washington.edu/accessit/articles?147>)) These early efforts to support accessibility, combined with the market dominance of Windows, both led to a disproportionate number of assistive technologies being developed for Windows.

With the release of Mac OS X, Apple improved the accessibility of its operating system. For example, it is now possible to access much more of the OS interface via keyboard than had been possible in previous versions. Also, Apple now provides an OS X Accessibility API (<http://developer.apple.com/documentation/Carbon/Accessibility-date.html>). Despite these efforts, however, there still are comparatively few assistive technology products available for Mac OS. For example, the lone developer of a screen reading product for Mac OS discontinued development of this product, as well as its screen magnification product, as of Mac OS 9.x.

Some assistive technology and other accessibility features come bundled with all operating systems, but typically these applications provide only a minimal level of accessibility, not the full set of features that many users require for equal access to the OS and its applications. The following are common built-in accessibility features across all operating systems:

- \* Keyboard customization - Allows users to adjust keyboard behavior so they can (a) press one key at a time in place of multi-key combinations, (b) use the keyboard to control mouse movements, and (c) change the length of time it takes for a keystroke to be registered.
- \* Display customization - Allows users to control the display contrast, font style and size, size of icons, and other display characteristics.
- \* Multi-mode alerts - Provides system alerts visually for users who can't hear auditory alerts.

In addition to these basic accessibility features, both Windows and Mac OS include basic screen magnification software (Magnifier and CloseView, respectively). Windows additionally provides a basic screen reader application called Narrator, and Mac OS includes limited text-to-speech capabilities through its speech recognition feature, PlainTalk. Each of these products provides a basic level of access, but these applications fall far short of the more fully featured screen magnification and screen reading applications that are available for Windows via third party developers.

Linux differs from both Windows and Mac OS in that it is an open source operating system and is supported and advanced by a dedicated community of developers. To date, the Linux developer community has produced a basic core set of accessibility features (as described above), as well as a combined screen reader/screen magnification application, Braille output software, and an on-screen keyboard. Each of these products was developed for the popular GNOME desktop, a graphic interface environment that runs on both Linux and Unix. Additional information about GNOME accessibility is available in the AccessIT Knowledge Base article [Is Linux Accessible?](http://www.washington.edu/accessit/articles?13) (<http://www.washington.edu/accessit/articles?13>)

Additional information about Microsoft Windows accessibility is provided on the Microsoft Accessibility (<http://www.microsoft.com/enable/>) website. This site includes detailed descriptions of accessibility features in current and previous releases of Microsoft Windows, step-by-step tutorials, and guides for users with specific disabilities.

Additional information about Mac OS accessibility is provided on the Apple Accessibility Features (<http://www.apple.com/disability/easyaccess.html>) website. This site includes specific steps for activating and using the accessibility features of Mac OS System 7.x and higher.

---

This document was developed by and is reprinted with the permission of the University of Washington and AccessIT ([www.washington.edu/accessit](http://www.washington.edu/accessit)). Permission is granted to copy these materials for educational, noncommercial purposes provided the source is acknowledged.