

**MICROSOFT®**  
The High Performance Software  
Microsoft Corporation  
16011 NE 36th Way  
Box 97017  
Redmond, WA 98073-9717  
USA

**MICROSOFT.**

Microsoft Ltd  
Reading ENGLAND

QINIX Microsoft  
Seoul KOREA

Microsoft GmbH  
München DEUTSCHLAND

Microsoft SARL  
Paris FRANCE

Microsoft Pty  
Sydney NSW AUSTRALIA

Microsoft Canada Inc  
Toronto CANADA

Microsoft AB  
Stockholm SWEDEN

Microsoft SpA  
Milano ITALIA

Microsoft KK  
Tokyo JAPAN

### Microsoft Windows Software Development Kit System Requirements

- IBM® or COMPAQ® Personal Computer
- 512K memory
- DOS 2.0 or higher
- One double-sided disk drive and one hard disk
- Graphics adapter card (EGA-type card required for color)
- One of the following Microsoft languages:
  - Microsoft C Compiler
  - Microsoft Macro Assembler
  - Microsoft Pascal Compiler

© Copyright 1986, Microsoft Corporation. All rights reserved. Printed in USA.

Please note that the availability and specifications of products described here are subject to change without notice.

Microsoft and MS-DOS are registered trademarks of Microsoft Corporation.

0786 Part No. 098-050-206

## Microsoft® Windows Software Development Kit

The future is here  
for application  
developers.

Microcomputer software.  
The past was then.

There are as many ways to design a software user interface as there are programmers. Each one works. Each one solves a particular set of problems. But instead of simplifying things by making computers easier to use, these multiple approaches can discourage people from using computers at all. Because when it comes to the user interface, there is no commonality. There are no standards.

And as input and output (I/O) device technology advances, developers and users face ever-changing stumbling blocks. Every time a new device like a mouse, screen display, or printer is introduced, developers must decide whether to invest time and money to support it. Users must research if it will work with their favorite applications, as well as if a new application will support their existing devices.

Transferring data from one place to another is easy using pencil and paper. But with traditional software, it can be quite difficult simply to get numbers from a spreadsheet into a word processed document.

Similarly, most people handle unexpected interruptions in their work simply and naturally by switching back and forth between tasks. But when they use a computer and try to switch between applications, they discover that microcomputer software can't always accomplish what the mind performs effortlessly.

Up to now your only real alternative has been to build a complex, multi-function application that may not meet all your users' needs. But clearly there must be a better way.

Now there is a better way. Microsoft Windows represents the foundation of a whole new generation of applications that is changing the course of microcomputer software development.

And the Microsoft Windows Software Development Kit provides the key for you to create these exciting new applications. It contains the tools you need to write the Windows-specific code that will put you in the forefront of software development.

Microsoft Windows provides a rich, graphically oriented environment that features icons, drop-down menus, dialog boxes, fast bit-mapped graphics, and, of course, multiple windows. It standardizes tasks that are common to most applications and makes unique or unfamiliar ones more intuitive for the user to perform.

There are standard ways to make menu selections, to edit text, to scroll the display, and to start and stop individual applications. Any task can be carried out from the keyboard or with a mouse or any other Windows-supported pointing device. This increases the user's comfort level and minimizes learning time and training costs.

Now changing an output device is as simple as plugging it in. The Microsoft Windows Graphic Device Interface (GDI) does all the dirty work using device drivers, so any graphics device can work with any application.

Moving compatible data between your Windows application and another is as easy as select, copy, paste with the Windows Clipboard. In addition, your applications can take advantage of dynamic data exchange to move data directly between them.

And when combined with Windows multi-tasking capabilities, these features give your users unprecedented power. Now they can integrate your application with other Windows applications to build a dynamic and cohesive system.

A picture  
is worth a  
thousand  
words—and  
a lot more  
to the user.

You can get  
anywhere  
from here.

You can't  
get there  
from here.



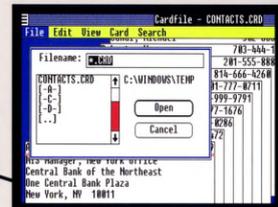
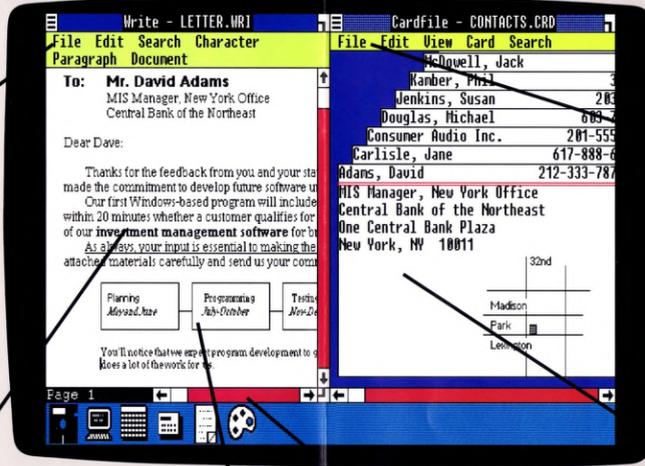
# Microsoft Windows Software Development Kit

## What's in it for you?



Menus provide a consistent way for users to execute commands and select options. Menu bars display primary selections across the top of the screen, and drop-down menus function as sub-menus providing additional choices.

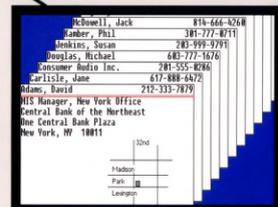
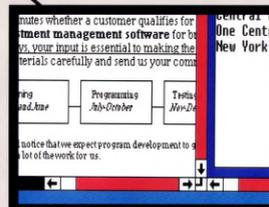
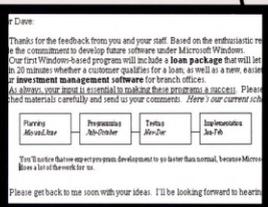
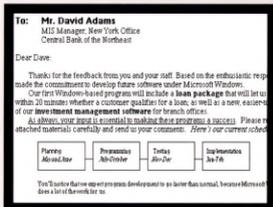
You specify menus as pairs of items: [name,value]. Name is what appears to the user in the menu; it can be either text or a bit-map. Value is the specification of the menu event expressed as an integer. When the user selects a "name," Windows returns the associated value to your application. Windows tracks input from the mouse or other pointing device and the keyboard, draws the menus, and handles housekeeping for you.



Dialog boxes are fill-in forms that let the user supply information in response to a menu choice. Many pre-defined field types can be included in your dialog boxes, such as the following:

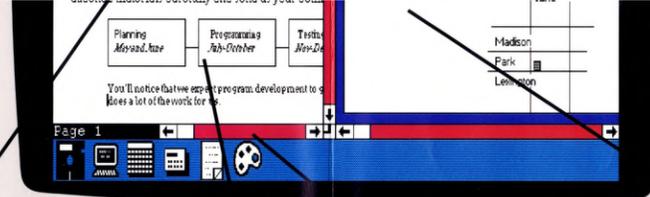
- Static text—for labels and messages
- Edit text—provides text entry fields
- Push buttons—for immediate action
- Radio buttons—allow one choice from many
- Check boxes—toggle for many choices

You create a dialog box by describing the layout and items in the form, and then writing code to handle the information filled in by the user. Windows draws the dialog box when needed, puts all the items in place, helps the user enter text, and returns the filled-in form when the user is finished.

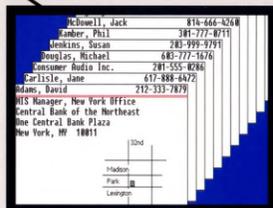
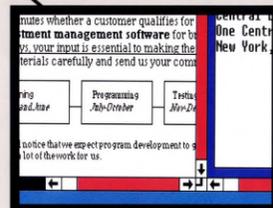
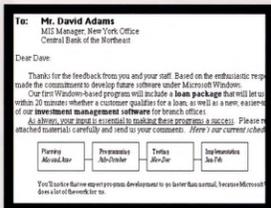


additional choices.

You specify names as pairs of [name,value]. Name is what appears to the user in the menu; it can be either text or a bit-map. Value is the specification of the menu event expressed as an integer. When the user selects a "name," Windows returns the associated value to your application. Windows tracks input from the mouse or other pointing device and the keyboard, draws the menus, and handles housekeeping for you.



- Static text—for labels and messages
  - Edit text—provides text entry fields
  - Push buttons—for immediate action
  - Radio buttons—allow one choice from many
  - Check boxes—toggle for many choices
- You create a dialog box by describing the layout and items in the form, and then writing code to handle the information filled in by the user. Windows draws the dialog box when needed, puts all the items in place, helps the user enter text, and returns the filled-in form when the user is finished.



**Fonts.** The Microsoft Windows GDI lets you build WYSIWYG (what you see is what you get) applications. Windows includes fonts for most popular displays and printers, and GDI can utilize any Windows font the user has installed, including those from third-party suppliers.

Your application's request for a font is answered by GDI selecting the most appropriate one available. The more fonts the user has installed, the closer GDI can match what is on the screen with what is printed out. Windows supports the ANSI character set, which includes the standard set of European characters. This makes providing international versions of your applications even easier.

**Graphics** can make your applications really come alive. Windows GDI is comprehensive, fast, and very powerful. It provides both vector functions (such as circle and polygon) and raster functions (such as bitblt and stretchblt) to allow great flexibility in graphics creation, including the ability to mix both text and graphics.

With the device independence GDI provides, you can easily create hard copy of displayed text or graphics on any GDI-supported output device. Once you've set up the device, you can display text or graphics on the screen or print on the printer with the same sequence of calls.

**Scroll bars** offer an easy and consistent way to move around text or graphics. You specify vertical and/or horizontal scroll bars. Windows draws them and provides functions to help you maintain the screen. These functions let you reappear the portion of the current display that will remain after the screen has scrolled, so you only have to redraw the portion of the screen that will be new.

**Child windows** let you restructure the screen area of the main window of your application. Child windows may be either tiled or overlapped and set up with various parent-child relationships.

You manage the location of child windows and the priority of overlapping child windows. Windows performs many redrawing and housekeeping functions for you such as clipping, which keeps all output inside the child window border.



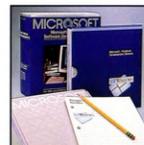
### Utilities

The Windows Software Development Kit provides you with a full set of tools including a set of reference manuals to help you write Windows applications. These tools help you

prepare the various templates and pieces that make up Windows applications and help you compile and debug.

- **Dialog editor** lets you design dialog boxes on the display screen and save a definition of the box in a resource file.

- **Icon editor** helps you create customized icons, cursors, and bitmaps.
- **Font editor** lets you create your own font files.
- **Resource compiler** lets you compile your application resource files and add them to the main resource file.
- **MAKE (Program Maintenance Utility)** automates the process of maintaining your programs. It compares source file modification dates and keeps track of file dependencies.
- **Symbolic debugger** provides a controlled testing environment to help debug your Windows applications. By creating a symbol file from your compiled source code, you can take advantage of the debugger's expanded command set.



### Microsoft support

The most important feature of Microsoft Windows is our dedicated commitment to it and to you, the Windows developer. We know that Windows will be a success only if our support to you is every

bit as good as the Windows product itself. That's why we've established a complete support program that serves you in all critical areas.

We have instituted DIAL, an on-line technical support service to process and administer requests specific

to your applications. The system is designed so your request is reviewed within 24 hours and a response given as soon as possible. The system also provides electronic mail, access to technical bulletin boards, and communication between software developers and Microsoft.

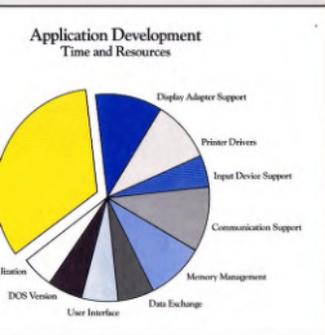
In addition, classes and seminars are provided on various Windows related topics such as how to write efficient Windows code, how to design applications to best take advantage of the Windows environment, and how to implement communication between programs.

Our commitment to Microsoft Windows and you is absolute. We know the time for Windows has come, and are doing everything we can to guarantee its success.

## Why your applications should be Microsoft Windows applications.

As a typical application developer, you spend significant amounts of time adapting your applications to new I/O devices. And it takes even more time to port applications to different environments and maintain utility subroutine libraries.

But with Microsoft Windows, you can spend more time writing applications and less time reinventing the wheel and doing maintenance. Windows provides over 300 new functions that handle everything from menus, dialog boxes, and graphics to data exchange and memory management. Instead of writing and maintaining these subroutines yourself, you can rely on Windows to provide them for you.



Microsoft Windows lets you devote more time and resources to your application code by handling many tasks for you.

## Microsoft Windows brings the age of the virtual device.

The Windows Graphic Device Interface allows your applications to take advantage of new technology automatically. When your application needs graphic output, it calls GDI, which, in turn, calls the device driver supplied for each device being used. If the device can directly produce the desired result, GDI and the device driver will take advantage of that capability. If not, GDI will simulate the necessary action and send it to the device.

Although GDI handles today's devices easily, it was really designed with tomorrow's devices in mind. When a smarter device—like a laser printer or video coprocessor board—is added to a system, Windows runs faster because GDI lets the device do more of the work.

Applications specifically designed for the Windows interface will run on MS-DOS® systems that run Windows. So instead of investing time and money in writing many versions of each application, you can write many applications, each in one standard version.

And when it comes to providing international versions of your applications, Microsoft Windows backs you up with discrete resource files that dramatically cut the time needed for localization. Each application is linked to its own resource file that can hold all the information to be translated or modified. So a non-technical person can translate text and change sizes and layouts without having to modify the application's source code.

Windows allows you to determine international formats for dates, time, currency, and numbers. And there is support for local keyboards and alternate character sets.

## Microsoft Windows is flexible.

Even with its highly standardized user interface, Windows still permits a great deal of developer customization. Any of the default interface features can be tailored to give your applications an individual look. And you have complete control over the screen area of your main window with many options for fonts, menus, dialog boxes, and more.

## Microsoft Windows is evolving.

Windows is a critical part of Microsoft's systems strategy. As MS-DOS grows and evolves, Windows—as an extension of the operating system—will grow and evolve with it. This strategy will assure that Windows and MS-DOS will always work together.



Microsoft Windows resource files provide the key to quick localization of your applications.

## Acceptance of Windows means opportunities for you.

Microsoft Windows is steadily gaining momentum as the preferred graphical operating environment for microcomputers. It has been selected as the supported operating environment by many hardware manufacturers—AT&T, Digital Equipment Corporation, Hewlett-Packard, Olivetti, Tandy Corporation, Texas Instruments, Zenith, and others. In addition, many corporations and software developers have selected Windows as their foundation for application development including Aldus, DCA, Living Videotext, Lotus Development Corporation, Micrograf, MicroPro, Micromim, Palantir, and Reuters. And, of course, Microsoft is leading the way in Windows application development.

As more Windows applications appear on the market, more people will become familiar with the simplicity and standardization the Windows environment offers. They can feel more comfortable about starting to use computers as well as about learning new applications. And as this happens there will be greater and greater demand for Windows applications, opening more opportunities for you.

Microsoft Windows heralds a new age in microcomputer software. Now is the time to take hold of the future. We have the product, the tools, and the support to help you get there.

