



The Win32 Preliminary Software Development Kit for Windows NT gives you the tools and information you need to develop powerful, 32-bit Windows-based applications for the rapidly expanding market for Windows-based software.

Overview

The success of the Microsoft® Windows® operating system, as millions of personal computer users have discovered, is that it makes people more productive than ever before. Users choose Windows for its ease-of-use, application availability, productivity, and its smooth integration into existing MS-DOS® environments.

Today's business environment places more demands on computing technology. Line-of-business applications such as inventory management, financial trading and modeling, and online transaction processing demand powerful hardware and a reliable, responsive, and secure operating system. Building these complex line of business solutions requires sophisticated system software.

To create these powerful applications you need the Win32 API with its 32-bit flat memory model, multi-threading, preemptive multitasking, and interprocess communication features. By writing to the Win32 API not only will you be able to take advantage of these features, but your applications will be source compatible between Intel® 386/486 platforms, and RISC

platforms such as MIPS R4000 and DEC® Alpha.

The Win32 SDK for Windows NT is the definitive development environment for creating these powerful, portable 32-bit applications. By purchasing the Win32 SDK (with hard-copy documentation) you will receive preliminary and final versions of the SDK tools and hard-copy documentation, preliminary versions of a C/C++ compiler and documentation, and preliminary and final versions of the Windows NT operating system software.

The Win32 API: Technical Highlights

Take advantage of your

experience with Windows 3.x

You're probably already familiar with the API.

Programmers of Windows 3.x applications will be immediately familiar with the Win32 API and development environment. The Win32 API offers a widened, 32-bit implementation of the Windows 3.1 API, and the development environment is based on Microsoft C/C++ 7.0.

The Win32 SDK makes it easy to move existing Windows 3.x applications to Win32.

Only minimal changes have been made to the syntax of the Win32 API. The API names are the same as Windows 3.x. The semantics are identical. The message order is identical. In fact, in many situations it will be possible to keep a single source code base and compile that

source code into both 16-bit and 32-bit applications.

The Win32 API offers 32-bit implementations of the Windows 3.1 programming features that you're already familiar with, such as:

- n Object linking and embedding (OLE)
- n Drag and Drop
- n TrueType®
- n Dynamic Data Exchange Management Library (DDEML)
- n Dynamic-Link Libraries (DLLs)
- n Windows 3.1 compatible parameter validation (compile and runtime)
- n Multimedia audio support
- n Common dialogs such as File Save, Print, and Select Font

Build powerful applications

The 32-bit flat memory model does away with the burden of managing memory in 64k segments.

- n Provides access up to 2 GB of address space per application.
- n Win32, and Windows NT's full 32-bit internal architecture allows for easier development of programs that deal with large amounts of data.

Multi-threading and preemption make it easy to create simple designs for complex solutions.

- n **Multiple threads** of execution per process allow applications to be more powerful and responsive.
- n **Preemptive multitasking** allows your applications to do computationally intensive tasks in the background while still being responsive to user input. Win32 makes more sophisticated use of system resources and efficiently allocates processing time to each thread and application.
- n **Synchronization objects** such as critical sections, events, mutexes, and semaphores provide the tools necessary to build reliable and robust multi-threaded applications.

The Win32 GDI is a complete set of general purpose drawing APIs.

- n **Bézier curves**- This function combined with the PolyBézier functionality makes it possible to draw any combination of continuous lines and curves.
- n **Paths**- Path APIs make it easy for an application to manage multiple shapes efficiently. These shapes can consist of an arbitrary combination of lines, arcs, ellipses, and Bézier curves.
- n **Complete 2-D transforms**- Combined with TrueType, this API, makes it possible to draw truly device-independent graphics that the system can map to the display surface.

Interprocess communication features make it easy to create network independent client-server applications.

- n **Remote Procedure Calls (RPCs)**-

The Win32 RPC API is compliant with the Distributed Computing Environment (DCE) specification. Distributed applications written to the Win32 RPC API can communicate and integrate with a wide range of other DCE-compliant systems such as those running Windows NT, UNIX[®], and VMS[®].

- n **Named pipes** allow two or more processes to communicate with each other. Any process that knows the name of a named pipe can access it (subject to security checks); the processes accessing the pipe do not need to be related.
- n **Mailslots** provide a unidirectional interprocess communication mechanism.

More advanced features of the Win32 API.

- n **Structured exception handling** allows for cleaner and more robust code. When an exception is initiated, a systematic search is performed to find an exception handler that will dispose of the exception.
- n **Memory mapped files** allow for faster, easier, and more efficient access to files. This API allows an application to map files into its address space. Data within the file can then be accessed using simpler memory read/write instructions rather than I/O system functions such as rewind and seek.
- n **Unicode** makes it easy to support international markets. Unicode can represent all of the world's characters in modern computer use, including technical symbols and special characters needed in publishing.

Maintain a single code base for applications targeting multiple platforms

Applications written to the Win32 API are source compatible between Intel 386/486 platforms, and RISC platforms such as MIPS R4000, and DEC Alpha.

This makes it easier than ever for you to have a multiple platform strategy.

Win32s[®] allows 32-bit applications to run on Windows 3.1 as well as Windows NT.

A subsequent release of the Win32 SDK for Windows NT will contain Win32s, which allows you to write to a single API and have your application run on both Windows 3.1 (on a 80386 machine or higher) and the Windows NT operating system. Since the executable format for Win32s is identical to that of Windows NT, the same binary can be run interchangeably on Windows 3.1 and

Windows NT.

Write 32-bit applications for the platform of choice

Windows is the graphical user

interface of choice. Since it's debut in 1990, more than 10 million copies of Windows have been licensed, and industry forecasts predict continue strong growth. The increasing availability of Windows-based applications and the rapid acceptance of the Windows operating system have propelled application sales to record numbers. Windows NT will continue this momentum, and the Win32 SDK for Windows NT let's you get started writing great 32-bit Windows-based applications today.

Windows NT: Technical Highlights

Your powerful 32-bit applications can run on the same platform as existing productivity applications.

Windows NT runs a host of today's applications including MS-DOS, Windows 3.x, POSIX, and character-based 16-bit OS/2[®] applications on Intel, MIPS, and DEC Alpha-based systems.

Memory protection and fault tolerance mean your applications and data are safe.

- n Memory protection provides the operating system and applications their own memory space to prevent data corruption and assure data integrity.
- n Under Windows NT, only system code can run in the most privileged execution mode. Applications cannot directly access hardware; applications make hardware requests through the Windows NT operating system.
- n **Fault Tolerance:** The Windows NT file system (NTFS) provides advanced file system features, including hot-fix and a full recovery system to quickly restore file integrity. NTFS maintains a transaction log to insure the integrity of the disk structure even if the system fails unexpectedly. Support for RAID5 (Redundant Array of Inexpensive Disks) and uninterruptible power supplies is also included in Windows NT. Additional fault tolerance capabilities such as disk mirroring, drive duplexing, and striping, will be available in LAN Manager for Windows NT (available separately.)

High Capacity means that you don't

have to worry about your memory usage growing too large for the system to handle.

Windows NT can access up to 4GB of RAM, and multiple terabytes of storage using 64-bit addressing. Applications such as relational databases, which demand large amounts of RAM for optimal performance, can access more virtual memory (2GB per application) than before. The Windows NT 32-bit architecture and 64-bit file and disk volume addressing eliminate any architectural limits on processes, threads, handles and other system resources.

Symmetric multiprocessing support means Windows NT can deliver extraordinary performance by taking advantage of multiple CPUs.

The Windows NT operating system is designed from the ground up to support symmetric multiprocessing. Each additional processor delivers a linear increase in performance. Windows NT can allocate threads within the same process to different processors; and can allocate not only application threads across processors, but system threads as well.

Once again, it is Windows NT's modern design and layered architecture which make this possible. The micro-kernel (part of the Executive, *see figure 1*) services the rest of the Executive, which is fully preemptible, multithreaded and reentrant. Through this design, not only do applications receive the benefit of multiple processors, so does the operating system.

Win32-based applications run without modification on symmetric multiprocessing systems, and threaded applications automatically take advantage of multiple processors when available. Users benefit from improved performance, and transparent scalability.

The portable design of Windows NT means your applications run on a wide range of computer systems

Unlike previous PC operating systems, which were designed and hardcoded for only one platform, the Microsoft Windows NT operating system is portable across platforms. It will run not only on 32-bit Intel systems, but also on RISC architectures, such as the MIPS R4000, and DEC Alpha.

The layered architecture of Windows NT is crucial to its portability. Within the Windows NT Executive is a compact micro-kernel. The micro-kernel handles low-level, machine-dependent functions, multiprocessor synchronization, thread dispatching and kernel objects. Windows NT code which has

to be changed when porting to a new type of CPU is located here and in the hardware abstraction layer (*see figure 1*) to make it easier to port other processors.

Built-in security means sensitive data is secure.

Security is an important element in today's computing environment. The Windows NT operating system delivers all the security you are likely to need. In fact, it's designed to be certified at US government C2-level for secure environments.

Figure 1: The Windows NT modular operating system design Built-in networking means it's easier to create network independent distributed applications.

Windows NT includes built in file and print-sharing capabilities, along with powerful workgroup application services.

- n Advanced network and systems management facilities available through LAN Manager for Windows NT (available separately.)
- n Open networking interface allows Novell, Banyan, and other network vendors to incorporate Windows NT systems into their network.
- n Support for DCE compliant RPC, named pipes, mailslots, NetBIOS, and sockets.
- n Support for NetBEUI and TCP/IP.

Win32 SDK Specifications

The Win32 SDK contains a complete set of powerful tools-- everything you need to start developing 32-bit Windows applications now. The development environment is equivalent to the toolset included in the Windows 3.1 SDK and C/C++ 7.0. There is no need to learn a new development environment.

As part of the Win32 Preliminary SDK for Windows NT (with hard-copy documentation) you will receive the following SDK tools and documentation in preliminary and final form. A CD only version of the Win32 SDK with PostScript[®] files for the documentation is also available. You may print the these files providing you have a PostScript printer.

SDK tools

- n *Dialog Box Editor* allows you to add, modify, and delete controls as you design and test dialog boxes on screen.
- n *Image Editor* lets you create and modify icons, bitmaps, and cursors.
- n *Hotspot Editor* lets you create and edit hypergraphics, which are bitmaps that include one or more hotspots. A hotspot can cover any portion of the bitmap and include multiple hotspots that link to help topics or run help macros when clicked.

- n *Font Editor* lets you alter existing fonts to create new fonts unique to your application.
- n *DDESpy* lets you monitor Dynamic Data Exchange (DDE) activity.
- n *Spy* is a tool that lets you monitor messages that are sent to one or more windows. Spy also lets you examine message parameter values.
- n *Zoomin* magnifies portions of a Windows screen to help you identify paint problems or other screen-related issues.
- n *Compress* reduces file size 25-45%.
- n *Expand* decompresses files previously compressed by Compress.
- n *Source Profiler* help you optimize the performance of your application by letting you determine the amount of time the operating system spends executing sections of code.
- n *Resource Compiler* is used for compiling your applications resources into an executable.
- n *Help Compiler* supports rich-text format (RTF) tables, secondary windows, and jumps across help files.
- n *Process Viewer*

SDK Documentation

- n *Win32 Programmer's Reference- Overview*
- n *Win32 Programmer's Reference- API Part 1*
- n *Win32 Programmer's Reference- API Part 2*
- n *The Windows Interface: An Application Design Guide*
- n *RPC Programmer's Guide and Reference*

Compilers and other tools

Since compilers and tools are not yet commercially available for Win32, you will also receive preliminary versions of the following tools and documentation. (the CD only option does not contain printed documentation)

- n *Microsoft C/C++*: Builds upon MS C/C++ 7.0 and generates 32-bit code. Includes complete edit, build, debug and compile tools, and includes Win32-compatible Microsoft Foundation Class (MFC) libraries.
- n *WinDbg* is a Win32-based debugger
- n *Microsoft Editor*
- n *Linker*- links object files created by the compiler into executable images.
- n *Microsoft Macro Assembler* for both Intel and R4000 architectures.

Compiler documentation

- n *C Language Reference*
- n *C++ Language Reference*
- n *C/C++ Tutorial*
- n *Class libraries Users Guide*
- n *Class Libraries Reference*
- n *Runtime Library Reference*
- n *Tools*
- n *Programming Techniques*

Windows NT operating system

You will also receive preliminary and final versions of the Windows NT operating system software.

System Requirements

Windows NT operation system

The first preliminary release of the Windows NT operating system requires an Intel 80386 system or higher, or ACE/ARC compatible R4000 systems with at least 8MB of RAM and 55MB available hard disk space (includes 20MB for a swap file). *This does not include disk space for applications.* See the Hardware Compatibility List for a complete list of systems on which preliminary testing has been done. The most recent hardware compatibility list is in the MSWIN32 forum on CompuServe®.

Win32 SDK development environment

For Win32 SDK development, an 80386 33 MHz CPU (or higher) is recommended, and 12MB total of RAM is required (16MB RAM is recommended.) A total of 100MB available hard disk space (includes 20MB for a swap file) is required for complete installation of the operating system, SDK tools, and C/C++ compiler.

The Win32 SDK is only available on CD-ROM and requires a CD-ROM drive and SCSI adapter. For a list of computer systems, CDROM drives, SCSI cards and other peripherals on which preliminary testing has been done, see the Hardware Compatibility List.

Device Driver Development

A preliminary version of the Microsoft Windows NT Device Driver Kit will also be available during the fall of 1992. It will provide all the documentation, tools, and sample source code you need to develop drivers for printers, displays, pointing devices etc.

Product Support

Twenty-four hour online technical information about Microsoft products is available through CompuServe. As part of the

Microsoft Connection on CompuServe you can exchange messages with Microsoft support engineers and experienced users of Microsoft products. Plus, you can download free software (such as drivers, sample code, utilities, and add-ons) provided by Microsoft and CompuServe members. Through the Microsoft Connection, users can now access the Microsoft Developer Services area. This area offers:

n ***Developer Forums.*** These forums cover information on the Windows operating system, languages, tools, and utilities from a developer's perspective. For example, a special forum, MSWIN32, has been set up to support this preliminary version of the Win32 SDK.

n ***Developer Knowledge Base.*** This up-to-date reference tool, compiled by Microsoft Product Support engineers,

contains developer-specific technical information about Microsoft products.

n ***Software Library.*** This collection of text and graphics files, sample code and utilities is keyword searchable, and these files can be downloaded for local use.

To connect to the Microsoft Connection, type GO MICROSOFT at the CompuServe "!" prompt. For more information about signing up for a CompuServe account, call (800) 848-8199, 8:00 A.M. to 10:00 P.M. eastern time, and ask for operator 230 to receive a \$15 connect time usage credit.

Ordering Information

Order the Win32 Preliminary SDK for Windows NT today! Call Microsoft Developer Services at (800) 227-4679. In Canada, call the Microsoft Canada Customer

Support Centre at (800) 563-9048. Outside the 50 United States and Canada, please contact your local Microsoft representative.