



Chapter 2

NIOS for DOS, MS Windows, and Windows95 Design

| | |
|--|----|
| Introduction | 10 |
| DOS and Windows API Calls | 10 |
| DOS Call Services | 11 |
| File Services | 11 |
| Memory Management Services | 11 |
| Miscellaneous Services | 12 |
| Protected-Mode Interrupt Management Services | 12 |
| Screen Management Services | 12 |
| V86 Callback Management Services | 12 |
| V86 Interrupt Management Services | 13 |
| V86 Mode Management Services | 13 |
| Virtual Machine Management Services | 13 |
| Services Available to Real-Mode DOS Applications | 13 |
| Services Available to 16-bit MS Windows Applications | 14 |
| Services Available to 32-bit MS Windows95 Applications | 14 |

Introduction

For each platform it supports, NIOS provides a set of platform-specific APIs for those client modules that are not OS-independent, such as API mappers. Applications that use these functions are not portable to other operating system platforms unless these calls are replaced by calls specific to the new operating system (e.g., DosCall might be replaced by Os2Call).

For the DOS/MS Windows environment, NIOS provides the functionality needed for protected-mode operation in a DOS environment, since DOS does not provide any protected-mode services.

Because these services also function when MS Windows is running, in many cases DOS/MS Windows specific code need not interface to the base OS, whether DOS or the MS Windows VMM directly.

DOS and Windows API Calls

Note that there are four different sets of API functions available. Which set the caller uses depends on the caller's environment.

- Real Mode DOS
- 16-bit Mode MS Windows
- 32-bit Mode MS Windows
- 32-bit NLMs

The following sections summarize these OS-dependent services grouped by functionality.

See Chapters 4 through 7 for detailed descriptions of each API listed below.

DOS Call Services

| | |
|---------------------------------|--|
| DosBeginUseDos | Causes Ctrl-C, Ctrl-Break, and Int 24h to be ignored. |
| DosCall | Assembly language API used to call DOS. |
| DosCallC | C API used to call DOS. |
| DosCallWithDTA | C API used to make DOS function calls that use the Disk Transfer Area. |
| DosCallUseCurrSDA | Calls DOS using the current DOS SDA information. |
| DosCancelDosAvailEvent | Cancel a previously scheduled DOS available event. |
| DosDeregisterUserCmd | Deregisters a previously installed custom DOS command. |
| DosEndUseDos | Restores the Ctrl-C, Ctrl-Break, and Int 24h vectors. |
| DosEnumerateUserCmds | Determines which DOS custom commands have been registered. |
| DosInfoBlock | Global structure containing misc. information about DOS. (In-DOS pointer, list of lists, etc.) |
| DosIsDosBusy | Determines whether DOS is busy. |
| DosRegisterUserCmd | Registers a new or replacement custom DOS command. |
| DosScheduleDosAvailEvent | Schedules an event that fires when DOS is callable. |

File Services

| | |
|-------------------------|--|
| DosClose | Closes a file. |
| DosCreate | Creates a new file. |
| DosDelete | Deletes a file. |
| DosDoesFileExist | Determines if the specified file exists. |
| DosFlush | Flushes all disk buffers. |
| DosGetFileSize | Returns a file's size. |
| DosOpen | Opens a file. |
| DosRead | Reads data from a file. |
| DosRename | Renames a file. |
| DosSearchForFile | Searches for the specified file. |
| DosSeek | Changes the current read/write/seek position for an opened file. |
| DosWrite | Writes data to a file. |

Memory Management Services

| | |
|----------------------------|---|
| DosAMapFlat | Assembly API used to convert a MS-Windows sel:off to a linear address. |
| DosCMapFlat | C API used to convert a MS-Windows sel:off to a linear address. |
| DosConvGetInfo | Returns the size of the largest available block of conventional memory. |
| DosConvMemAlloc | Allocates a memory block below 1MB. |
| DosConvMemFree | Deallocates a previously allocated conventional memory block. |
| DosSharedBufAlloc | Allocates the shared DOS buffer provided by NIOS. |
| DosSharedBufFree | Deallocates the shared DOS buffer. |
| DosSharedBufGetInfo | Gets information about the shared DOS buffer. |
| Win16GetProcAddress | Resolves the sel:off of an exported 16-bit Windows procedure. |

Miscellaneous Services

| | |
|------------------------------|---|
| DosBeginReentrantExec | Used in special cases where an NLM needs to invoke a service that normally isn't callable at hardware interrupt time. |
| DosEndReentrantExec | Ends a reentrant execution block. |
| DosGetExeContext | Determines if current execution is in the foreground or is in the context of a hardware interrupt. |
| DosWinDebFlag | Global flag set to 1 if Windows is active and a debugger is loaded. |
| DosWinFlag | Global flag that is non-zero when enhanced-mode Windows is active. |

Protected-Mode Interrupt Management Services

| | |
|------------------------------------|--|
| DosHookPMInterrupt | Installs a handler for the specified PM interrupt. |
| DosUnHookPMInterrupt | Deinstalls a PM interrupt handler. |
| DosHookExceptionInterrupt | Installs a handler for a specific processor exception interrupt. |
| DosUnHookExceptionInterrupt | Deinstalls a handler for an exception interrupt. |
| WinCallWhenPMIntReturns | Obtains control on the back end of a current PM interrupt. |
| WinHookPMInt21 | Hooks PM int 21 handlers in Windows. |
| WinUnHookPMInt21 | Unhooks PM int 21 handlers in Windows. |

Screen Management Services

| | |
|--------------------------------|---|
| DosVid16DeregisterGuiCB | Cancels a previously registered GUI callback. |
| DosVid16RegisterGuiCB | Sets the address of the current GUI callback. |
| DosVidCallWhenPopupOk | Schedules an event to fire when the system can display a popup. |
| DosVidCheckKey | Determines if a key is waiting in the keyboard buffer. |
| DosVidCursorSet | Positions the cursor to the specified x,y coordinate in popup. |
| DosVidEmptyTypeAhead | Empties the keyboard typeahead buffer. |
| DosVidGetKey | Waits for a key press and returns the key value. |
| DosVidGetPopupInfo | Obtains information about active popup. |
| DosVidIsPopupOk | Determines if context allows display of popup message. |
| DosVidPopup | Displays a popup message on the screen. |
| DosVidPopupExt | Displays a popup with title, subtitle, prompt, and message text. |
| DosVidRestoreScreen | Restores a portion of the screen from a dynamically allocated buffer. |
| DosVidSaveScreen | Saves a portion of the screen to a dynamically allocated buffer. |
| DosVidSoundBell | Rings the bell once. |
| DosVidStdOut | Displays the specified prefix and message using DOS STDOUT. |
| DosVidWriteToPopup | Writes a string to x,y position in popup. |

V86 Callback Management Services

| | |
|----------------------------|--|
| DosAllocV86Callback | Allocates a V86 callback address. |
| DosFreeV86Callback | Deallocates a previously allocated V86 callback. |

V86 Interrupt Management Services

| | |
|---------------------------------|--|
| DosRegisterV86Int2F | Installs a handler for a specific Int 2F AH value. |
| DosDeregisterV86Int2F | Deinstalls an Int 2F handler. |
| DosHookV86Interrupt | Installs a handler for the specified V86 interrupt. |
| DosUnHookV86Interrupt | Deinstalls a V86 interrupt handler. |
| DosCallWhenV86IntReturns | Used to gain control on the back end of a V86 interrupt. |

V86 Mode Management Services

| | |
|--------------------------------|---|
| DosBeginNestExec | Allocates a new ClientRegStruc for nested V86 execution. |
| DosEndNestExec | Deallocates a ClientRegStruc that was used for nested V86 execution. |
| DosBeginNestExecWithCrs | Sets up a nested V86 execution block using the provided ClientRegStruc. ("C" callable.) |
| DosEndNestExecWithCrs | Ends a nested V86 execution block. ("C" callable.) |
| DosExecuteV86FarCall | Calls a V86 procedure with a far return stack frame. |
| DosExecuteV86Int | Executes the specified V86 interrupt. |
| DosFastExecuteFarRet | Macro that simulates a V86 far return instruction. |
| DosFastExecutePop | Macro that simulates a V86 pop instruction. |
| DosFastExecutePush | Macro that simulates a V86 push instruction. |
| DosExecuteIRet | Simulates a V86 iret instruction. |
| DosExecuteFarRet | Simulates a V86 far return instruction. |
| DosExecutePop | Simulates a V86 pop instruction. |
| DosExecutePush | Simulates a V86 push instruction. |

Virtual Machine Management Services

| | |
|---------------------------|---|
| DosGetCurrVmHandle | Returns a pointer to the current VM control block. |
| DosGetNextVmHandle | Allows enumeration of existing VM control blocks. |
| DosVmIdToVmCbTable | Global table used to obtain the VM control block for a given VM Id. |

Services Available to Real-Mode DOS Applications

| | |
|----------------------------------|---|
| Get NIOS Real Mode API | Returns API entry points used to access NIOS and other NLMs. |
| DosNiosFarCallHandler | Real-mode entry point for invoking NIOS functions. |
| DosInvokeRegNlmApiHandler | Invokes an exported NLM function that uses a register-based calling convention. |
| DosInvokeCNlmApiHandler | Invokes an exported NLM function that uses C calling conventions. |
| RM_NIOS_BEGIN_USE_API | Determines the 32-bit flat linear address of specified server NLM API name. |
| RM_NIOS_COPY_MEM | Copies data from a memory buffer above 1MB to a V86 mode buffer. |
| RM_NIOS_COPY_STRING | Copies string pointed to by pmBuffer into specified 16-bit seg:off buffer. |
| RM_NIOS_END_USE_API | Signals that the DOS application is done using the specified server NLM API. |

| | |
|----------------------|--|
| RM_NIOS_MAP | Converts a seg:off value to a linear address and locks the memory. |
| RM_NIOS_UNMAP | Potentially unlocks a previously-mapped V86 mode buffer. |

Services Available to 16-bit MS Windows Applications

| | |
|---|--|
| Get NIOS Windows 16-bit Mode API | Accesses the NIOS APIs available to 16-bit Windows applications. |
| Win16InvokeRegNlmApiHandler | Invokes an exported NLM function that uses a register-based calling convention. |
| Win16InvokeCNlmApiHandler | Invokes an exported NLM function that uses C calling conventions. |
| Win16LoadModule | Loads an NLM. |
| Win16NiosFarCallHandler | Protected-mode entry point for invoking NIOS functions. |
| Win16UnloadModule | Unloads an NLM. |
| PM16_NIOS_BEGIN_USE_API | Returns the 32-bit flat linear address of the specified server NLM API name. |
| PM16_NIOS_COPY_MEM | Copies data from a linear address into the specified selector's offset buffer. |
| PM16_NIOS_COPY_STRING | Copies string from a linear address into the specified selector's offset buffer. |
| PM16_NIOS_END_USE_API | Signals that the application is done with the specified server NLM API function. |

Services Available to 32-bit MS Windows95 Applications

| | |
|---------------------------------|---|
| Win32InvokeCNlmApi | Calls an exported NLM function that uses the "C" calling conventions. |
| Win32LoadModule | Loads an NLM for Win32 applications. |
| Win32NiosFarCall | Invokes NIOS services for Win32 applications. |
| Win32UnloadModule | Unloads an NLM for Win32 applications. |
| WIN32_NIOS_BEGIN_USE_API | Determines the 32-bit linear address of the specified NLM API name. |
| WIN32_NIOS_COPY_MEM | Copies the contents of the memory at the specified Ring 0 linear address into a Ring 3 buffer. |
| WIN32_NIOS_COPY_STRING | Copies the string pointed to by the Ring 0 pmBuffer address into a Ring 3 application buffer. |
| WIN32_NIOS_END_USE_API | Signals that the application is no longer going to use the specified NLM API function. |
| WIN32_NIOS_MAP | Converts the specified linear address local of a calling Win32 process into a globally accessible linear address. |
| WIN32_NIOS_UNMAP | Unlocks the application memory block and destroys the global linear range alias created through WIN32_NIOS_MAP. |