



This is part of **Family API** which allow to create dual-os version of program runs under OS/2 and DOS

**Note:** This is legacy API call. It is recommended to use 32-bit equivalent

2021/09/17 04:47 · prokushev · [0 Comments](#)

2021/08/20 03:18 · prokushev · [0 Comments](#)

Name	Description	Cross-platform
<a href="#">BadDynLink</a>	This is DOS specific version function which called during module loading if linking error is detected. It shows message and terminates via int 21h Terminate call	-
<a href="#">DosBeep</a>	Generates sound from the speaker using direct port I/O	No
<a href="#">DosBufReset</a>	Flushes a file cache buffers	
<a href="#">DosChDir</a>	Defines the current directory for the requesting process	
<a href="#">DosChgFilePtr</a>	Moves the read/write pointer	
<a href="#">DosClose</a>	Closes a handle to a file, pipe, or device	
<a href="#">DosCreateCSAlias</a>	Create CS alias from data segment	
<a href="#">DosCLIAccess</a>	Request I/O privilege for disabling and enabling interrupts	
<a href="#">DosPortAccess</a>	Request or release access to ports for I/O privilege	
<a href="#">DosDelete</a>	Removes a directory entry associated with a file name	
<a href="#">DosDevConfig</a>	Return device configuration	
<a href="#">DosDupHandle</a>	Returns a new file handle for an open file	
<a href="#">DosFreeSeg</a>	Deallocates a memory segment	
<a href="#">DosGetDateTime</a>	Get the current date and time	
<a href="#">DosGetEnv</a>	Return process environment for the current process	
<a href="#">DosGetHugeShift</a>	Return a shift count used to derive the selectors that address memory allocated with DosAllocHuge	
<a href="#">DosGetMachineMode</a>	Returns the current mode of the processor	
<a href="#">DosGetMessage</a>	Retrieve a message from the specified system message file	
<a href="#">DosGetVersion</a>	Return the OS version number	
<a href="#">DosInsMessage</a>	Insert variable text string information into the body of a message	
<a href="#">DosMkDir</a>	Create a subdirectory	
<a href="#">DosMkDir2</a>	Create a subdirectory with EA	
<a href="#">DosMove</a>	Move a file object to another location, change its name	
<a href="#">DosNewSize</a>	Changes the size of a file	
<a href="#">DosPutMessage</a>	Output the message	
<a href="#">DosQCurDir</a>	Returns the full path name of the current directory	
<a href="#">DosQCurDisk</a>	Determines the current default drive for the requesting process	
<a href="#">DosQFileMode</a>	Queries the mode (attribute) of the specified file	
<a href="#">DosQFSInfo</a>	Query file system info	
<a href="#">DosQVerify</a>	Returns the value of the verify flag	
<a href="#">DosRmDir</a>	Removes a subdirectory from the specified disk	
<a href="#">DosSelectDisk</a>	Selects the drive specified as the default drive	
<a href="#">DosSetDateTime</a>	Set the date and time	
<a href="#">DosSetFileInfo</a>	Set attribute and extended attribute information for a file	
<a href="#">DosSetFileMode</a>	Changes the mode (attribute) of the specified file	

Name	Description	Cross-platform
<a href="#">DosSetVerify</a>	Sets write verification	
<a href="#">DosSleep</a>	Suspend the current thread for a specified time	
<a href="#">DosSubAlloc</a>	Suballocate portions of a segment	
<a href="#">DosSubFree</a>	Free memory previously allocated by DosSubAlloc	
<a href="#">DosSubSet</a>	Initialize a segment for suballocation	
<a href="#">DosWrite</a>	Write buffer to file	
<a href="#">DosAllocHuge</a>	Allocate multiple segments as a huge block of memory	
<a href="#">DosAllocSeg</a>	Allocate a data segment up to 64KB in size	
<a href="#">DosCaseMap</a>	Case mapping on a string	
<a href="#">DosDevIOctl</a>	Control functions on a device	
<a href="#">DosDevIOctl2</a>	Control functions on a device	
<a href="#">DosError</a>	Receive hard error notification	
<a href="#">DosErrClass</a>	Receive hard error code information	
<a href="#">DosExecPgm</a>	Execute another program as a child process	
<a href="#">DosExit</a>	End The current thread or process	
<a href="#">DosFileLocks</a>	Locks and unlock a range in an opened file	
<a href="#">DosFindClose</a>	Finish DosFindFirst or DosFindNext directory search function sequence	
<a href="#">DosFindFirst</a>	Finds the first file object or group of file objects whose name(s) match the specification	
<a href="#">DosFindFirst2</a>	Finds the first file object or group of file objects whose name(s) match the specification	
<a href="#">DosFindNext</a>	Locate the next set of directory entries	
<a href="#">DosGetCtryInfo</a>	Obtain country-dependent formatting information that resides in the country information file	
<a href="#">DosGetDBCSEv</a>	Obtain a DBCS (double byte character set) environmental vector that resides in the country information file	
<a href="#">DosGetCP</a>	Query the current process code page and the prepared system code pages	
<a href="#">DosSetCP</a>	Set process code page and the session's display code page and keyboard code page	
<a href="#">DosGetCollate</a>	Obtain a collating sequence table	
<a href="#">DosHoldSignal</a>	Temporarily disable or enable signal processing for the current process	
<a href="#">DosOpen</a>	Open a file, a named pipe, or a device	
<a href="#">DosOpen2</a>	Open a file with extended attributes	
<a href="#">DosQFileInfo</a>	Return information for a specific file	
<a href="#">DosRead</a>	Read the specified number of bytes from a file, pipe, or device to a buffer location	1.00
<a href="#">DosReallocHuge</a>	Change the size of memory originally allocated by DosAllocHuge	
<a href="#">DosReallocSeg</a>	Reallocate a segment after discard or changes the size of a segment already allocated	1.00
<a href="#">DosSetCtryCode</a>		
<a href="#">DosSetFHandState</a>	Set the state of the specified file	
<a href="#">DosSetSigHandler</a>	Set signal handler	
<a href="#">DosAllocShrSeg</a>	Allocate a named shared memory segment to a process	
<a href="#">DosGetShrSeg</a>	Accesses a shared memory segment previously allocated by another process	

Name	Description	Cross-platform
<a href="#">DosLoadModule</a>	Load a dynamic link module and returns a handle for the module	
<a href="#">DosFreeModule</a>	Free the reference to a dynamic link module for a process	
<a href="#">DosQHandType</a>	Get handle type	
<a href="#">DosGetProcAddr</a>	Get module procedure address	
<a href="#">DosGetPID</a>	Get process id	
<a href="#">DosSetMaxFH</a>	Set maximum file handlers	
<a href="#">DosGetModHandle</a>	Get module handle	
<a href="#">DosQPathInfo</a>	Get path information	
<a href="#">DosQFSAttach</a>	Query information about an attached file system	
<a href="#">DosQSysInfo</a>	Query system variables values	
<a href="#">DosMemAvail</a>	Query maximum available huge memory block	
<a href="#">DosGetInfoSeg</a>	Query global and local information segments	
<a href="#">KbdCharIn</a>	Return a character data record from the keyboard	
<a href="#">KbdFlushBuffer</a>	Clear the keystroke buffer	
<a href="#">KbdGetStatus</a>	Get the current state of the keyboard	
<a href="#">KbdSetStatus</a>	Set the characteristics of the keyboard	
<a href="#">KbdStringIn</a>	Read a character string (character codes only) from the keyboard	
<a href="#">KbdPeek</a>	Return any available character data record from the keyboard without removing it from the buffer	
<a href="#">KbdOpen</a>	Create a new logical keyboard	
<a href="#">KbdClose</a>	Close the existing logical keyboard	
<a href="#">KbdGetFocus</a>		
<a href="#">KbdFreeFocus</a>		
<a href="#">KbdGetCp</a>		
<a href="#">KbdSetCp</a>		
<a href="#">KbdXlate</a>		
<a href="#">KbdSetCustXt</a>		
<a href="#">KbdGetHWId</a>		
<a href="#">KbdRegister</a>		
<a href="#">KbdDeRegister</a>		
<a href="#">MouRegister</a>		
<a href="#">MouDeRegister</a>		
<a href="#">MouGetNumButtons</a>		
<a href="#">MouGetNumMickeyes</a>		
<a href="#">MouGetDevStatus</a>		
<a href="#">MouGetNumQueEI</a>		
<a href="#">MouReadEventQue</a>		
<a href="#">MouGetScaleFact</a>		
<a href="#">MouGetEventMask</a>		
<a href="#">MouSetScaleFact</a>		
<a href="#">MouSetEventMask</a>		
<a href="#">MouGetHotKey</a>		
<a href="#">MouSetHotKey</a>		
<a href="#">MouOpen</a>		
<a href="#">MouClose</a>		

Name	Description	Cross-platform
<a href="#">MouGetPtrShape</a>		
<a href="#">MouSetPtrShape</a>		
<a href="#">MouDrawPtr</a>		
<a href="#">MouRemovePtr</a>		
<a href="#">MouGetPtrPos</a>		
<a href="#">MouSetPtrPos</a>		
<a href="#">MouInitReal</a>		
<a href="#">MouFlushQue</a>		
<a href="#">MouSetDevStatus</a>		
<a href="#">VioGetBuf</a>	Return the address of the logical video buffer (LVB)	
<a href="#">VioGetCurPos</a>	Return the coordinates of the cursor	
<a href="#">VioGetCurType</a>	Get cursor type	
<a href="#">VioGetPhysBuf</a>	Get addressability to the physical display buffer	
<a href="#">VioReadCellStr</a>	Read a string of character-attribute pairs from the screen	
<a href="#">VioReadCharStr</a>	Read a string of characters from the display	
<a href="#">VioScrollDn</a>	Scroll the entire display buffer (or area specified within the display buffer) down	
<a href="#">VioScrollLf</a>	Scroll the entire display buffer (or area specified within the display buffer) to the left	
<a href="#">VioScrollRt</a>	Scroll the entire display buffer (or area specified within the display buffer) to the right	
<a href="#">VioScrUnLock</a>	Release ownership of (unlocks) the physical display buffer	
<a href="#">VioSetCurPos</a>	Set the cursor's coordinates on the screen	
<a href="#">VioSetCurType</a>	Set the cursor type	
<a href="#">VioSetMode</a>	Set the mode of the display	
<a href="#">VioShowBuf</a>	Update the physical display buffer with the logical video buffer (LVB)	
<a href="#">VioWrtCellStr</a>	Write a string of character-attribute pairs (cells) to the display	
<a href="#">VioWrtCharStr</a>	Write a character string to the display	
<a href="#">VioWrtCharStrAtt</a>	Write a character string with repeated attribute to the display	
<a href="#">VioWrtNAttr</a>	Write an attribute to the display a specified number of times	
<a href="#">VioWrtNCell</a>	Write a cell (character-attribute pair) to the display a specified number of times	1.00
<a href="#">VioWrtNChar</a>	Write a character to the display a specified number of times	
<a href="#">VioWrtTTY</a>	Write a character string to the display starting at the current cursor position	
<a href="#">VioScrLock</a>	Request ownership of (locks) the physical display buffer	
<a href="#">VioGetMode</a>	Return the mode of the display	
<a href="#">VioGetConfig</a>	Return the video display configuration	
<a href="#">VioGetAnsi</a>	Return the current ANSI status On/Off state	
<a href="#">VioSetAnsi</a>	Activate or deactivate ANSI support	
<a href="#">VioScrollUp</a>	Scroll the entire display buffer (or area specified within the display buffer) up	
<a href="#">VioDeRegister</a>	Deregister alternate video system	
<a href="#">VioRegister</a>	Register alternate video system	
<a href="#">VioGetState</a>	Return the current settings of adapter	

Name	Description	Cross-platform
<a href="#">VioSetState</a>	Set the current settings of adapter	
<a href="#">VioGetCP</a>	Query the code page for display	
<a href="#">VioSetCP</a>	Set the code page for display	
<a href="#">VioGetFont</a>	Get current font	
<a href="#">VioSetFont</a>	Set current font	
<a href="#">VioModeWait</a>	Notify process about it must restore its video mode	
<a href="#">VioModeUndo</a>		
<a href="#">VioPopUp</a>	Show temporary screen to display message	
<a href="#">VioEndPopUp</a>	Return from temporary screen	
<a href="#">VioSavRedrawWait</a>	Notifies application when it must save/redraw its screen	
<a href="#">VioSavRedrawUndo</a>		
<a href="#">VioPrtSc</a>	Dump screen to printer	
<a href="#">VioPrtScToggle</a>	Toggle VioWrtTty also write to printer	

From:

<https://cocorico.osfree.org/doku/> - **osFree wiki**

Permanent link:

[https://cocorico.osfree.org/doku/doku.php?id=en:docs:fapi:implementation\\_details&rev=1634464214](https://cocorico.osfree.org/doku/doku.php?id=en:docs:fapi:implementation_details&rev=1634464214)

Last update: **2021/10/17 09:50**

