



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](#)

## Family API

Family API (FAPI) is a subset of [Control Program API](#) which can be used to write binary portable applications. Such applications can be run as on OS/2 as on DOS system without any modifications. It is known 2 versions of original Family API 1.00 and 1.10. Also exists side Family API implementation. FAMAPI by Jonathan de Boyne Pollard and HX DOS Extender API by Andreas Grech. Versions up to 1.10 is a original OS/2 Family API. 1.20 and higher is a osFree extensions.

## Dual OS applications

It is possible to write programs which will run on OS/2, DOS and Windows NT from one binary. Moreover, same source code can be used without any `#ifdef` and other preprocessor statements. Such portability achieved via Family API. Family API is OS/2 API emulation layer on top of DOS. OS/2 executable file is in NE (New Executable) file format. NE file consist of two parts:

1. Legacy DOS MZ EXE format part;
2. NE EXE part.

Using Family API MZ part of file used to provide loading and dynamic linking mechanism to load and link NE. Also Family API file contains emulation library which translates OS/2 API calls to DOS interrupt calls. So, same file can be executed as in OS/2 as in DOS. Windows NT contains OS2 Subsystem (`os2ss.exe`) which provides OS/2 API layer on top of Windows NT kernel. So, Family API allows to support 3 OSes using one binary file.

For current time only 16-bit Family API supported.

## Writing portable tools

[http://www.edm2.com/index.php/Hints\\_for\\_writing\\_simple\\_programs\\_for\\_both\\_OS/2\\_and\\_DOS](http://www.edm2.com/index.php/Hints_for_writing_simple_programs_for_both_OS/2_and_DOS)

# Function Calls

OS/2 1.0 introduced around 80-90 function calls (various information sources differ) that could be used in FAPI programs.

Name	Description	Module (OS/2)	Library (DOS)	Status (OS/2)	Status (DOS)	FAPI Version
<a href="#">BadDynLink</a>	Terminates execution with error message	-	API/FAPI	-	Done	1.00
<a href="#">DosBeep</a>	Generates sound from the speaker	<a href="#">DOSCALLS</a>	API/FAPI		Done	1.00
<a href="#">DosBufReset</a>	Flushes a file cache buffers	<a href="#">DOSCALLS</a>	API/FAPI	Done	Done	1.10
<a href="#">DosChDir</a>	Defines the current directory for the requesting process	<a href="#">DOSCALLS</a>	API/FAPI	Done	Done	1.00
<a href="#">DosChgFilePtr</a>	Moves the read/write pointer	<a href="#">DOSCALLS</a>	API/FAPI	Done	Done	1.00
<a href="#">DosClose</a>	Closes a handle to a file, pipe, or device	<a href="#">DOSCALLS</a>	API/FAPI	Done	Done	1.00
<a href="#">DosCreateCSAlias</a>	Create CS alias from data segment	<a href="#">DOSCALLS</a>	API/FAPI		Done	1.00
<a href="#">DosCLIAccess</a>	Request I/O privilege for disabling and enabling interrupts	<a href="#">DOSCALLS</a>	API/FAPI		Done	1.00
<a href="#">DosPortAccess</a>	Request or release access to ports for I/O privilege	<a href="#">DOSCALLS</a>	API/FAPI		Done	1.00
<a href="#">DosDelete</a>	Removes a directory entry associated with a file name	<a href="#">DOSCALLS</a>	API/FAPI	Done	Done	1.00
<a href="#">DosDevConfig</a>	Return device configuration	<a href="#">DOSCALLS</a>	API/FAPI		Done	1.00
<a href="#">DosDupHandle</a>	Returns a new file handle for an open file	<a href="#">DOSCALLS</a>	API/FAPI	Done	Done	1.00
<a href="#">DosFreeSeg</a>	Deallocates a memory segment	<a href="#">DOSCALLS</a>	API/FAPI		Done	1.00
<a href="#">DosGetDateTime</a>	Get the current date and time	<a href="#">DOSCALLS</a>	API/FAPI		Done	1.00
<a href="#">DosGetEnv</a>	Return process environment for the current process	<a href="#">DOSCALLS</a>	API/FAPI		Done	1.00
<a href="#">DosGetHugeShift</a>	Return a shift count used to derive the selectors that address memory allocated with <a href="#">DosAllocHuge</a>	<a href="#">DOSCALLS</a>	API/FAPI		Done	1.00
<a href="#">DosGetMachineMode</a>	Returns the current mode of the processor	<a href="#">DOSCALLS</a>	API/FAPI	Done	Done	1.00
<a href="#">DosGetMessage</a>	Retrieve a message from the specified system message file	<a href="#">DOSCALLS</a>	API/FAPI			1.00
<a href="#">DosGetVersion</a>	Return the OS version number	<a href="#">DOSCALLS</a>	API/FAPI	Done	Done	1.00
<a href="#">DosInsMessage</a>	Insert variable text string information into the body of a message	<a href="#">DOSCALLS</a>	API/FAPI			1.00

Name	Description	Module (OS/2)	Library (DOS)	Status (OS/2)	Status (DOS)	FAPI Version
<a href="#">DosMkDir</a>	Create a subdirectory	<a href="#">DOSCALLS</a>	API/FAPI	Done	Done	1.00
<a href="#">DosMkDir2</a>	Create a subdirectory with EA	<a href="#">DOSCALLS</a>	API/FAPI	Done	Done	????
<a href="#">DosMove</a>	Move a file object to another location, change its name	<a href="#">DOSCALLS</a>	API/FAPI		Done	1.00
<a href="#">DosNewSize</a>	Changes the size of a file	<a href="#">DOSCALLS</a>	API/FAPI		Done	1.00
<a href="#">DosPutMessage</a>	Output the message	<a href="#">DOSCALLS</a>	API/FAPI			1.00
<a href="#">DosQCurDir</a>	Returns the full path name of the current directory	<a href="#">DOSCALLS</a>	API/FAPI		Done	1.00
<a href="#">DosQCurDisk</a>	Determines the current default drive for the requesting process	<a href="#">DOSCALLS</a>	API/FAPI		Done	1.00
<a href="#">DosQFileMode</a>	Queries the mode (attribute) of the specified file	<a href="#">DOSCALLS</a>	API/FAPI		Done	1.00
<a href="#">DosQFSInfo</a>	Query file system info	<a href="#">DOSCALLS</a>	API/FAPI			1.00
<a href="#">DosQVerify</a>	Returns the value of the verify flag	<a href="#">DOSCALLS</a>	API/FAPI	Done	Done	1.00
<a href="#">DosRmDir</a>	Removes a subdirectory from the specified disk	<a href="#">DOSCALLS</a>	API/FAPI	Done	Done	1.00
<a href="#">DosSelectDisk</a>	Selects the drive specified as the default drive	<a href="#">DOSCALLS</a>	API/FAPI	Done	Done	1.00
<a href="#">DosSetDateTime</a>	Set the date and time	<a href="#">DOSCALLS</a>	API/FAPI		Done	1.00
<a href="#">DosSetFileInfo</a>	Set attribute and extended attribute information for a file	<a href="#">DOSCALLS</a>	API/FAPI		Done	1.00
<a href="#">DosSetFileMode</a>	Changes the mode (attribute) of the specified file	<a href="#">DOSCALLS</a>	API/FAPI		Done	1.00
<a href="#">DosSetVerify</a>	Sets write verification	<a href="#">DOSCALLS</a>	API/FAPI	Done	Done	1.00
<a href="#">DosSleep</a>	Suspend the current thread for a specified time	<a href="#">DOSCALLS</a>	API/FAPI		Done	1.00
<a href="#">DosSubAlloc</a>	Suballocate portions of a segment	<a href="#">DOSCALLS</a>	API/FAPI			1.00
<a href="#">DosSubFree</a>	Free memory previously allocated by DosSubAlloc	<a href="#">DOSCALLS</a>	API/FAPI			1.00
<a href="#">DosSubSet</a>	Initialize a segment for suballocation	<a href="#">DOSCALLS</a>	API/FAPI			1.00
<a href="#">DosWrite</a>	Write buffer to file	<a href="#">DOSCALLS</a>	API/FAPI		Done	1.00
<a href="#">DosAllocHuge</a>	Allocate multiple segments as a huge block of memory	<a href="#">DOSCALLS</a>	API/FAPI		Done	1.00
<a href="#">DosAllocSeg</a>	Allocate a data segment up to 64KB in size	<a href="#">DOSCALLS</a>	API/FAPI		Done	1.00
<a href="#">DosCaseMap</a>	Case mapping on a string	<a href="#">DOSCALLS</a>	API/FAPI			1.10
<a href="#">DosDevIOctl</a>	Control functions on a device	<a href="#">DOSCALLS</a>	API/FAPI			1.00
<a href="#">DosDevIOctl2</a>	Control functions on a device	<a href="#">DOSCALLS</a>	API/FAPI			????
<a href="#">DosError</a>	Receive hard error notification	<a href="#">DOSCALLS</a>	API/FAPI		Done	1.00

Name	Description	Module (OS/2)	Library (DOS)	Status (OS/2)	Status (DOS)	FAPI Version
<a href="#">DosErrClass</a>	Receive hard error code information	<a href="#">DOSCALLS</a>	API/FAPI			1.10
<a href="#">DosExecPgm</a>	Execute another program as a child process	<a href="#">DOSCALLS</a>	API/FAPI		Done	1.00
<a href="#">DosExit</a>	End The current thread or process	<a href="#">DOSCALLS</a>	API/FAPI		Done	1.00
<a href="#">DosFileLocks</a>	Locks and unlock a range in an opened file	<a href="#">DOSCALLS</a>	API/FAPI		Done	1.00
<a href="#">DosFindClose</a>	Finish DosFindFirst or DosFindNext directory search function sequence	<a href="#">DOSCALLS</a>	API/FAPI		Done	1.00
<a href="#">DosFindFirst</a>	Finds the first file object or group of file objects whose name(s) match the specification	<a href="#">DOSCALLS</a>	API/FAPI		Done	1.00
<a href="#">DosFindFirst2</a>	Finds the first file object or group of file objects whose name(s) match the specification	<a href="#">DOSCALLS</a>	API/FAPI			???
<a href="#">DosFindNext</a>	Locate the next set of directory entries	<a href="#">DOSCALLS</a>	API/FAPI		Done	1.00
<a href="#">DosGetCtryInfo</a>	Obtain country-dependent formatting information that resides in the country information file	<a href="#">DOSCALLS</a>	API/FAPI			1.10
<a href="#">DosGetDBCSEv</a>	Obtain a DBCS (double byte character set) environmental vector that resides in the country information file	<a href="#">DOSCALLS</a>	API/FAPI		Done	1.10
<a href="#">DosGetCP</a>	Query the current process code page and the prepared system code pages	<a href="#">DOSCALLS</a>	API/FAPI		Done	1.10
<a href="#">DosSetCP</a>	Set process code page and the session's display code page and keyboard code page	<a href="#">DOSCALLS</a>	API/FAPI		Done	1.10
<a href="#">DosGetCollate</a>	Obtain a collating sequence table	<a href="#">DOSCALLS</a>	API/FAPI			1.10
<a href="#">DosHoldSignal</a>	Temporarily disable or enable signal processing for the current process	<a href="#">DOSCALLS</a>	API/FAPI			1.00
<a href="#">DosOpen</a>	Open a file, a named pipe, or a device	<a href="#">DOSCALLS</a>	API/FAPI		Done	1.00
<a href="#">DosOpen2</a>	Open a file with extended attributes	<a href="#">DOSCALLS</a>	API/FAPI			????
<a href="#">DosQFileInfo</a>	Return information for a specific file	<a href="#">DOSCALLS</a>	API/FAPI		Done	1.00
<a href="#">DosRead</a>	Read the specified number of bytes from a file, pipe, or device to a buffer location	<a href="#">DOSCALLS</a>	API/FAPI		Done	1.00

Name	Description	Module (OS/2)	Library (DOS)	Status (OS/2)	Status (DOS)	FAPI Version
<a href="#">DosReallocHuge</a>	Change the size of memory originally allocated by DosAllocHuge	<a href="#">DOSCALLS</a>	API/FAPI			1.00
<a href="#">DosReallocSeg</a>	Reallocate a segment after discard or changes the size of a segment already allocated	<a href="#">DOSCALLS</a>	API/FAPI		Done	1.00
<a href="#">DosSetCtryCode</a>		<a href="#">DOSCALLS</a>	API/FAPI			1.00
<a href="#">DosSetFHandState</a>	Set the state of the specified file	<a href="#">DOSCALLS</a>	API/FAPI			1.00
<a href="#">DosSetSigHandler</a>	Set signal handler	<a href="#">DOSCALLS</a>	API/FAPI		Done	1.00
<a href="#">DosAllocShrSeg</a>	Allocate a named shared memory segment to a process	<a href="#">DOSCALLS</a>	API/FAPI		Done	1.20
<a href="#">DosGetShrSeg</a>	Accesses a shared memory segment previously allocated by another process	<a href="#">DOSCALLS</a>	API/FAPI		Done	1.20
<a href="#">DosLoadModule</a>	Load a dynamic link module and returns a handle for the module	<a href="#">DOSCALLS</a>	API/FAPI		Done	1.20
<a href="#">DosFreeModule</a>	Free the reference to a dynamic link module for a process	<a href="#">DOSCALLS</a>	API/FAPI		Done	1.20
<a href="#">DosQHandType</a>	Get handle type	<a href="#">DOSCALLS</a>	API/FAPI		Done	1.00
<a href="#">DosGetProcAddr</a>	Get module procedure address	<a href="#">DOSCALLS</a>	API/FAPI		Done	1.20
<a href="#">DosGetPID</a>	Get process id	<a href="#">DOSCALLS</a>	API/FAPI		Done	1.00
<a href="#">DosSetMaxFH</a>	Set maximum file handlers	<a href="#">DOSCALLS</a>	API/FAPI		Done	1.20
<a href="#">DosGetModHandle</a>	Get module handle	<a href="#">DOSCALLS</a>	API/FAPI		Done	1.20
<a href="#">DosQPathInfo</a>	Get path information	<a href="#">DOSCALLS</a>	API/FAPI			1.??
<a href="#">DosQFSAttach</a>	Query information about an attached file system	<a href="#">DOSCALLS</a>	API/FAPI			1.??
<a href="#">DosQSysInfo</a>	Query system variables values	<a href="#">DOSCALLS</a>	API/FAPI		Done	1.20
<a href="#">DosMemAvail</a>	Query maximum available huge memory block	<a href="#">DOSCALLS</a>	API/FAPI		Done	1.20
<a href="#">DosGetInfoSeg</a>	Query global and local information segments	<a href="#">DOSCALLS</a>	API/FAPI		Done	1.20
<a href="#">KbdCharIn</a>	Return a character data record from the keyboard	<a href="#">KBDCALLS</a>	API/FAPI		Done	1.00
<a href="#">KbdFlushBuffer</a>	Clear the keystroke buffer	<a href="#">KBDCALLS</a>	API/FAPI		Done	1.00
<a href="#">KbdGetStatus</a>	Get the current state of the keyboard	<a href="#">KBDCALLS</a>	API/FAPI		Done	1.00
<a href="#">KbdSetStatus</a>	Set the characteristics of the keyboard	<a href="#">KBDCALLS</a>	API/FAPI			1.00
<a href="#">KbdStringIn</a>	Read a character string (character codes only) from the keyboard	<a href="#">KBDCALLS</a>	API/FAPI			1.00

Name	Description	Module (OS/2)	Library (DOS)	Status (OS/2)	Status (DOS)	FAPI Version
<a href="#">KbdPeek</a>	Return any available character data record from the keyboard without removing it from the buffer	<a href="#">KBDCALLS</a>	API/FAPI		Done	1.00
<a href="#">KbdOpen</a>	Create a new logical keyboard	<a href="#">KBDCALLS</a>	API/FAPI		Done	1.20
<a href="#">KbdClose</a>	Close the existing logical keyboard	<a href="#">KBDCALLS</a>	API/FAPI		Done	1.20
<a href="#">KbdGetFocus</a>	Bind the logical keyboard to the physical keyboard	<a href="#">KBDCALLS</a>	API/FAPI		Done	1.20
<a href="#">KbdFreeFocus</a>	Free the logical-to-physical keyboard bond	<a href="#">KBDCALLS</a>	API/FAPI		Done	1.20
<a href="#">KbdGetCp</a>	Query the code page being used to translate scan codes to ASCII characters	<a href="#">KBDCALLS</a>	API/FAPI			1.20
<a href="#">KbdSetCp</a>	Set the code page used to translate key strokes received from the keyboard	<a href="#">KBDCALLS</a>	API/FAPI			1.20
<a href="#">KbdXlate</a>	Translate scan codes with shift states into ASCII codes	<a href="#">KBDCALLS</a>	API/FAPI			1.20
<a href="#">KbdSetCustXt</a>	Install, on the specified handle, the translate table	<a href="#">KBDCALLS</a>	API/FAPI			1.20
<a href="#">KbdGetHWId</a>	Return the type of keyboard in use	<a href="#">KBDCALLS</a>	API/FAPI			1.20
<a href="#">KbdRegister</a>	Registers a keyboard subsystem within a session	<a href="#">KBDCALLS</a>	API/FAPI			1.20
<a href="#">KbdDeRegister</a>	Deregister a keyboard subsystem within a session	<a href="#">KBDCALLS</a>	API/FAPI			1.20
<a href="#">MouRegister</a>	Register a mouse subsystem within a session	<a href="#">KBDCALLS</a>	API/FAPI			1.20
<a href="#">MouDeRegister</a>	Deregister a mouse subsystem within a session	<a href="#">KBDCALLS</a>	API/FAPI			1.20
<a href="#">MouGetNumButtons</a>	Return the number of buttons supported on the installed mouse driver	<a href="#">MOUCALLS</a>	API/FAPI			1.20
<a href="#">MouGetNumMickeys</a>	Return the number of mickeys in each centimeter for the installed mouse driver	<a href="#">MOUCALLS</a>	API/FAPI			1.20
<a href="#">MouGetDevStatus</a>	Return status flags for the installed mouse device driver	<a href="#">MOUCALLS</a>	API/FAPI			1.20
<a href="#">MouGetNumQueEl</a>	Return the current status for the mouse device driver event queue	<a href="#">MOUCALLS</a>	API/FAPI			1.20
<a href="#">MouReadEventQue</a>	Read an event from the mouse device FIFO event queue	<a href="#">MOUCALLS</a>	API/FAPI			1.20



Name	Description	Module (OS/2)	Library (DOS)	Status (OS/2)	Status (DOS)	FAPI Version
<a href="#">MouGetScaleFact</a>	Return a pair of 1-word scaling factors for the current mouse device	<a href="#">MOUCALLS</a>	API/FAPI			1.20
<a href="#">MouGetEventMask</a>	Return the current value of the mouse event queue mask	<a href="#">MOUCALLS</a>	API/FAPI			1.20
<a href="#">MouSetScaleFact</a>	Assign to the current mouse device driver a new pair of 1-word scaling factors	<a href="#">MOUCALLS</a>	API/FAPI			1.20
<a href="#">MouSetEventMask</a>	Assign a new event mask to the current mouse device driver	<a href="#">MOUCALLS</a>	API/FAPI			1.20
<a href="#">MouGetHotKey</a>	Return mouse button or buttons defined as system hot key	<a href="#">MOUCALLS</a>	API/FAPI			1.20
<a href="#">MouSetHotKey</a>	Define mouse button or buttons as system hot key	<a href="#">MOUCALLS</a>	API/FAPI			1.20
<a href="#">MouOpen</a>	Open the mouse device for the current session	<a href="#">MOUCALLS</a>	API/FAPI			1.20
<a href="#">MouClose</a>	Close the mouse device for the current session	<a href="#">MOUCALLS</a>	API/FAPI			1.20
<a href="#">MouGetPtrShape</a>	Get (copy) the pointer shape for the session	<a href="#">MOUCALLS</a>	API/FAPI			1.20
<a href="#">MouSetPtrShape</a>	Set the pointer shape and size for the session	<a href="#">MOUCALLS</a>	API/FAPI			1.20
<a href="#">MouDrawPtr</a>	Notify the mouse device driver that an area is now available	<a href="#">MOUCALLS</a>	API/FAPI			1.20
<a href="#">MouRemovePtr</a>	Notify the mouse device driver that the area is not available	<a href="#">MOUCALLS</a>	API/FAPI			1.20
<a href="#">MouGetPtrPos</a>	Query the current row and column coordinate	<a href="#">MOUCALLS</a>	API/FAPI			1.20
<a href="#">MouSetPtrPos</a>	Set a new row and column coordinate position	<a href="#">MOUCALLS</a>	API/FAPI			1.20
<a href="#">MouInitReal</a>	Initialize mouse pointer draw support for DOS mode	<a href="#">MOUCALLS</a>	API/FAPI			1.20
<a href="#">MouFlushQue</a>	Flush (empty) the mouse event queue	<a href="#">MOUCALLS</a>	API/FAPI			1.20
<a href="#">MouSetDevStatus</a>	Set the mouse device driver status flags	<a href="#">MOUCALLS</a>	API/FAPI			1.20
<a href="#">VioGetBuf</a>	Return the address of the logical video buffer (LVB)	<a href="#">VIOCALLS</a>	API/FAPI			1.20
<a href="#">VioGetCurPos</a>	Return the coordinates of the cursor	<a href="#">VIOCALLS</a>	API/FAPI		Done	1.00
<a href="#">VioGetCurType</a>	Get cursor type	<a href="#">VIOCALLS</a>	API/FAPI		Done	1.00
<a href="#">VioGetPhysBuf</a>	Get addressability to the physical display buffer	<a href="#">VIOCALLS</a>	API/FAPI			1.00

Name	Description	Module (OS/2)	Library (DOS)	Status (OS/2)	Status (DOS)	FAPI Version
<a href="#">VioReadCellStr</a>	Read a string of character-attribute pairs from the screen	<a href="#">VIOCALLS</a>	API/FAPI		Done	1.00
<a href="#">VioReadCharStr</a>	Read a string of characters from the display	<a href="#">VIOCALLS</a>	API/FAPI		Done	1.00
<a href="#">VioScrollDn</a>	Scroll the entire display buffer (or area specified within the display buffer) down	<a href="#">VIOCALLS</a>	API/FAPI		Done	1.00
<a href="#">VioScrollLf</a>	Scroll the entire display buffer (or area specified within the display buffer) to the left	<a href="#">VIOCALLS</a>	API/FAPI			1.00
<a href="#">VioScrollRt</a>	Scroll the entire display buffer (or area specified within the display buffer) to the right	<a href="#">VIOCALLS</a>	API/FAPI			1.00
<a href="#">VioScrUnLock</a>	Release ownership of (unlocks) the physical display buffer	<a href="#">VIOCALLS</a>	API/FAPI		Done	1.00
<a href="#">VioSetCurPos</a>	Set the cursor's coordinates on the screen	<a href="#">VIOCALLS</a>	API/FAPI		Done	1.00
<a href="#">VioSetCurType</a>	Set the cursor type	<a href="#">VIOCALLS</a>	API/FAPI		Done	1.00
<a href="#">VioSetMode</a>	Set the mode of the display	<a href="#">VIOCALLS</a>	API/FAPI		Done	1.00
<a href="#">VioShowBuf</a>	Update the physical display buffer with the logical video buffer (LVB)	<a href="#">VIOCALLS</a>	API/FAPI			1.20
<a href="#">VioWrtCellStr</a>	Write a string of character-attribute pairs (cells) to the display	<a href="#">VIOCALLS</a>	API/FAPI		Done	1.00
<a href="#">VioWrtCharStr</a>	Write a character string to the display	<a href="#">VIOCALLS</a>	API/FAPI		Done	1.00
<a href="#">VioWrtCharStrAtt</a>	Write a character string with repeated attribute to the display	<a href="#">VIOCALLS</a>	API/FAPI		Done	1.00
<a href="#">VioWrtNAttr</a>	Write an attribute to the display a specified number of times	<a href="#">VIOCALLS</a>	API/FAPI		Done	1.00
<a href="#">VioWrtNCell</a>	Write a cell (character-attribute pair) to the display a specified number of times	<a href="#">VIOCALLS</a>	API/FAPI		Done	1.00
<a href="#">VioWrtNChar</a>	Write a character to the display a specified number of times	<a href="#">VIOCALLS</a>	API/FAPI		Done	1.00
<a href="#">VioWrtTTY</a>	Write a character string to the display starting at the current cursor position	<a href="#">VIOCALLS</a>	API/FAPI		Done	1.00
<a href="#">VioScrLock</a>	Request ownership of (locks) the physical display buffer	<a href="#">VIOCALLS</a>	API/FAPI		Done	1.00



Name	Description	Module (OS/2)	Library (DOS)	Status (OS/2)	Status (DOS)	FAPI Version
<a href="#">VioGetMode</a>	Return the mode of the display	<a href="#">VIOCALLS</a>	API/FAPI		Done	1.20
<a href="#">VioGetConfig</a>	Return the video display configuration	<a href="#">VIOCALLS</a>	API/FAPI		Done	1.??
<a href="#">VioGetAnsi</a>	Return the current ANSI status On/Off state	<a href="#">VIOCALLS</a>	API/FAPI		Done	1.20
<a href="#">VioSetAnsi</a>	Activate or deactivate ANSI support	<a href="#">VIOCALLS</a>	API/FAPI		Done	1.20
<a href="#">VioScrollUp</a>	Scroll the entire display buffer (or area specified within the display buffer) up	<a href="#">VIOCALLS</a>	API/FAPI		Done	1.20
<a href="#">VioDeRegister</a>	Deregister alternate video system	<a href="#">VIOCALLS</a>	API/FAPI		Done	1.20
<a href="#">VioRegister</a>	Register alternate video system	<a href="#">VIOCALLS</a>	API/FAPI		Done	1.20
<a href="#">VioGetState</a>	Return the current settings of adapter	<a href="#">VIOCALLS</a>	API/FAPI			1.??
<a href="#">VioSetState</a>	Set the current settings of adapter	<a href="#">VIOCALLS</a>	API/FAPI			1.??
<a href="#">VioGetCP</a>	Query the code page for display	<a href="#">VIOCALLS</a>	API/FAPI			1.20
<a href="#">VioSetCP</a>	Set the code page for display	<a href="#">VIOCALLS</a>	API/FAPI			1.20
<a href="#">VioGetFont</a>	Get current font	<a href="#">VIOCALLS</a>	API/FAPI			1.??
<a href="#">VioSetFont</a>	Set current font	<a href="#">VIOCALLS</a>	API/FAPI			1.??
<a href="#">VioModeWait</a>	Notify process about it must restore its video mode	<a href="#">VIOCALLS</a>	API/FAPI			1.20
<a href="#">VioModeUndo</a>	Allow one thread within a process to cancel a VioModeWait	<a href="#">VIOCALLS</a>	API/FAPI			1.20
<a href="#">VioPopUp</a>	Show temporary screen to display message	<a href="#">VIOCALLS</a>	API/FAPI			1.20
<a href="#">VioEndPopUp</a>	Return from temporary screen	<a href="#">VIOCALLS</a>	API/FAPI			1.20
<a href="#">VioSavRedrawWait</a>	Notifies application when it must save/redraw its screen	<a href="#">VIOCALLS</a>	API/FAPI			1.20
<a href="#">VioSavRedrawUndo</a>	Allow one thread within a process to cancel a VioSavRedrawWait	<a href="#">VIOCALLS</a>	API/FAPI			1.20
<a href="#">VioPrtSc</a>	Dump screen to printer	<a href="#">VIOCALLS</a>	API/FAPI			1.20
<a href="#">VioPrtScToggle</a>	Toggle VioWrtTty also write to printer	<a href="#">VIOCALLS</a>	API/FAPI			1.20
<a href="#">DosSysTrace</a>	Write trace entry	<a href="#">DOSCALLS</a>	API/FAPI			1.20
<a href="#">DosGetSTDA</a>	Return Trace buffer	<a href="#">DOSCALLS</a>	API/FAPI			1.20

# Compatibility

Feature	DOS 16-bit Real Mode	DOS 16-bit Protected Mode <sup>1)</sup>	OS/2 16-bit Protected Mode
Max memory	640KB	16MB <sup>2)</sup>	16MB
Virtual memory	No	No	1Gb
Multitasking	No	No	Yes
Multithreading	No	No	Yes
Long filenames	Yes <sup>3)</sup>	Yes <sup>4)</sup>	Yes <sup>5)</sup>
Extended attributes	Yes <sup>6)</sup>	Yes <sup>7)</sup>	Yes <sup>8)</sup>
App EXE name change	Yes <sup>9)</sup>	Yes <sup>10)</sup>	Yes

## Remarks

For implementation details refer to [Implementation details](#) section.

## Notes

This text based on [http://www.edm2.com/index.php/Family\\_API](http://www.edm2.com/index.php/Family_API)

- <sup>1)</sup> Requires DPML host with 16-bit client support
- <sup>2)</sup> Actually depends on DPML host and current CPU
- <sup>3)</sup> <sup>4)</sup> Supported since version 1.20 Requires LFN driver under DOS or run in Windows 9x VDM.
- <sup>5)</sup> <sup>8)</sup> Starting from OS/2 1.??
- <sup>6)</sup> <sup>7)</sup> Under OS/2 BOX only
- <sup>9)</sup> <sup>10)</sup> Under DOS 3 and higher

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

Permanent link: <https://cocorico.osfree.org/doku/doku.php?id=en:docs:fapi&rev=1639806094>

Last update: **2021/12/18 05:41**

