



# 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.

## 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

As example of dual mode program lets clone EXEHDR tools from OS/2 and Windows SDK/Toolkit. First of all consider we writing program for OS/2. Other things will be done later to make dual mode program.

```
#include <os2.h>
#include <newexe.h>

exe_hdr mz_hdr;
new_exe ne_hdr;

void main(void)
{
    DosOpen(FileHandle);
    DosRead(FileHandle, mz_hdr);
    DumpMZ(mz_hdr);
    if (is_extended_exe(mz_hdr))
```

```

{
    DosRead(FileHandle, buffer);
    if (is_NE(buffer)) DumpNE((new_exe)buffer);
}
DosClose(FileHandle);
}

```

## Function Calls

OS/2 1.0 introduced 93(?) function calls that could be used in FAPI programs.

Name	Description	Module (OS/2)	Library (DOS)	Status (OS/2)	Status (DOS)
<a href="#">BadDynLink</a>		-	API/FAPI	-	
<a href="#">DosBeep</a>	Generates sound from the speaker	<a href="#">DOSCALLS</a>	API/FAPI		Done
<a href="#">DosBufReset</a>	Flushes a file cache buffers	<a href="#">DOSCALLS</a>	API/FAPI	Done	Done
<a href="#">DosChDir</a>	Defines the current directory for the requesting process	<a href="#">DOSCALLS</a>	API/FAPI	Done	Done
<a href="#">DosChgFilePtr</a>	Moves the read/write pointer	<a href="#">DOSCALLS</a>	API/FAPI	Done	Done
<a href="#">DosClose</a>	Closes a handle to a file, pipe, or device	<a href="#">DOSCALLS</a>	API/FAPI	Done	Done
<a href="#">DosCreateCSAlias</a>	Create CS alias from data segment	<a href="#">DOSCALLS</a>	API/FAPI		Done
<a href="#">DosDelete</a>	Removes a directory entry associated with a file name	<a href="#">DOSCALLS</a>	API/FAPI	Done	Done
<a href="#">DosDevConfig</a>	Return device configuration	<a href="#">DOSCALLS</a>	API/FAPI		Done
<a href="#">DosDupHandle</a>	Returns a new file handle for an open file	<a href="#">DOSCALLS</a>	API/FAPI	Done	Done
<a href="#">DosFreeSeg</a>	Deallocates a memory segment	<a href="#">DOSCALLS</a>	API/FAPI		Done
<a href="#">DosGetDateTime</a>	Get the current date and time	<a href="#">DOSCALLS</a>	API/FAPI		Done
<a href="#">DosGetEnv</a>	Returns the address of the process environment string for the current process	<a href="#">DOSCALLS</a>	API/FAPI		Done
<a href="#">DosGetHugeShift</a>	Returns a shift count used to derive the selectors that address memory allocated with DosAllocHuge	<a href="#">DOSCALLS</a>	API/FAPI		Done
<a href="#">DosGetMachineMode</a>	Returns the current mode of the processor	<a href="#">DOSCALLS</a>	API/FAPI	Done	Done

Name	Description	Module (OS/2)	Library (DOS)	Status (OS/2)	Status (DOS)
<a href="#">DosGetMessage</a>		<a href="#">DOSCALLS</a>	API/FAPI		
<a href="#">DosGetVersion</a>	Returns the OS version number	<a href="#">DOSCALLS</a>	API/FAPI	Done	Done
<a href="#">DosInsMessage</a>		<a href="#">DOSCALLS</a>	API/FAPI		
<a href="#">DosMkDir</a>	Creates a subdirectory	<a href="#">DOSCALLS</a>	API/FAPI	Done	Done
<a href="#">DosMove</a>	Moves a file object to another location and changes its name	<a href="#">DOSCALLS</a>	API/FAPI		Done
<a href="#">DosNewSize</a>	Changes the size of a file	<a href="#">DOSCALLS</a>	API/FAPI		Done
<a href="#">DosPutMessage</a>		<a href="#">DOSCALLS</a>	API/FAPI		
<a href="#">DosQCurDir</a>	Returns the full path name of the current directory	<a href="#">DOSCALLS</a>	API/FAPI		Done
<a href="#">DosQCurDisk</a>	Determines the current default drive for the requesting process	<a href="#">DOSCALLS</a>	API/FAPI		Done
<a href="#">DosQFileMode</a>	Queries the mode (attribute) of the specified file	<a href="#">DOSCALLS</a>	API/FAPI		Done
<a href="#">DosQFSInfo</a>		<a href="#">DOSCALLS</a>	API/FAPI		
<a href="#">DosQVerify</a>	Returns the value of the verify flag	<a href="#">DOSCALLS</a>	API/FAPI	Done	Done
<a href="#">DosRmDir</a>	Removes a subdirectory from the specified disk	<a href="#">DOSCALLS</a>	API/FAPI	Done	Done
<a href="#">DosSelectDisk</a>	Selects the drive specified as the default drive	<a href="#">DOSCALLS</a>	API/FAPI	Done	Done
<a href="#">DosSetDateTime</a>		<a href="#">DOSCALLS</a>	API/FAPI		
<a href="#">DosSetFileInfo</a>		<a href="#">DOSCALLS</a>	API/FAPI		
<a href="#">DosSetFileMode</a>		<a href="#">DOSCALLS</a>	API/FAPI		
<a href="#">DosSetVerify</a>	Sets write verification	<a href="#">DOSCALLS</a>	API/FAPI	Done	Done
<a href="#">DosSleep</a>		<a href="#">DOSCALLS</a>	API/FAPI		
<a href="#">DosSubAlloc</a>		<a href="#">DOSCALLS</a>	API/FAPI		
<a href="#">DosSubFree</a>		<a href="#">DOSCALLS</a>	API/FAPI		
<a href="#">DosSubSet</a>		<a href="#">DOSCALLS</a>	API/FAPI		
<a href="#">DosWrite</a>		<a href="#">DOSCALLS</a>	API/FAPI		
<a href="#">DosAllocHuge</a>		<a href="#">DOSCALLS</a>	API/FAPI		
<a href="#">DosAllocSeg</a>		<a href="#">DOSCALLS</a>	API/FAPI		
<a href="#">DosCaseMap</a>		<a href="#">DOSCALLS</a>	API/FAPI		
<a href="#">DosDevIOCtl</a>		<a href="#">DOSCALLS</a>	API/FAPI		
<a href="#">DosError</a>		<a href="#">DOSCALLS</a>	API/FAPI		
<a href="#">DosExecPgm</a>		<a href="#">DOSCALLS</a>	API/FAPI		
<a href="#">DosExit</a>		<a href="#">DOSCALLS</a>	API/FAPI		
<a href="#">DosFileLocks</a>		<a href="#">DOSCALLS</a>	API/FAPI		
<a href="#">DosFindClose</a>		<a href="#">DOSCALLS</a>	API/FAPI		
<a href="#">DosFindFirst</a>		<a href="#">DOSCALLS</a>	API/FAPI		
<a href="#">DosFindNext</a>		<a href="#">DOSCALLS</a>	API/FAPI		

Name	Description	Module (OS/2)	Library (DOS)	Status (OS/2)	Status (DOS)
<a href="#">DosGetCtryInfo</a>		<a href="#">DOSCALLS</a>	API/FAPI		
<a href="#">DosGetDBCSEv</a>		<a href="#">DOSCALLS</a>	API/FAPI		
<a href="#">DosHoldSignal</a>		<a href="#">DOSCALLS</a>	API/FAPI		
<a href="#">DosOpen</a>		<a href="#">DOSCALLS</a>	API/FAPI		
<a href="#">DosQFileInfo</a>		<a href="#">DOSCALLS</a>	API/FAPI		
<a href="#">DosRead</a>		<a href="#">DOSCALLS</a>	API/FAPI		
<a href="#">DosReallocHuge</a>		<a href="#">DOSCALLS</a>	API/FAPI		
<a href="#">DosReallocSeg</a>		<a href="#">DOSCALLS</a>	API/FAPI		
<a href="#">DosSetCtryCode</a>		<a href="#">DOSCALLS</a>	API/FAPI		
<a href="#">DosSetFHandState</a>		<a href="#">DOSCALLS</a>	API/FAPI		
<a href="#">DosSetSigHandler</a>		<a href="#">DOSCALLS</a>	API/FAPI		
<a href="#">KbdCharIn</a>		<a href="#">KBDCALLS</a>	API/FAPI		
<a href="#">KbdFlushBuffer</a>		<a href="#">KBDCALLS</a>	API/FAPI		
<a href="#">KbdGetStatus</a>		<a href="#">KBDCALLS</a>	API/FAPI		
<a href="#">KbdSetStatus</a>		<a href="#">KBDCALLS</a>	API/FAPI		
<a href="#">KbdStringIn</a>		<a href="#">KBDCALLS</a>	API/FAPI		
<a href="#">KbdPeek</a>		<a href="#">KBDCALLS</a>	API/FAPI		
<a href="#">VioGetBuf</a>		<a href="#">VIOCALLS</a>	API/FAPI		
<a href="#">VioGetCurPos</a>		<a href="#">VIOCALLS</a>	API/FAPI		
<a href="#">VioGetCurType</a>		<a href="#">VIOCALLS</a>	API/FAPI		
<a href="#">VioGetPhysBuf</a>		<a href="#">VIOCALLS</a>	API/FAPI		
<a href="#">VioReadCellStr</a>		<a href="#">VIOCALLS</a>	API/FAPI		
<a href="#">VioReadCharStr</a>		<a href="#">VIOCALLS</a>	API/FAPI		
<a href="#">VioScrollDn</a>		<a href="#">VIOCALLS</a>	API/FAPI		
<a href="#">VioScrollLf</a>		<a href="#">VIOCALLS</a>	API/FAPI		
<a href="#">VioScrollRt</a>		<a href="#">VIOCALLS</a>	API/FAPI		
<a href="#">VioScrUnLock</a>		<a href="#">VIOCALLS</a>	API/FAPI		
<a href="#">VioSetCurPos</a>		<a href="#">VIOCALLS</a>	API/FAPI		
<a href="#">VioSetCurType</a>		<a href="#">VIOCALLS</a>	API/FAPI		
<a href="#">VioSetMode</a>		<a href="#">VIOCALLS</a>	API/FAPI		
<a href="#">VioShowBuf</a>		<a href="#">VIOCALLS</a>	API/FAPI		
<a href="#">VioWrtCellStr</a>		<a href="#">VIOCALLS</a>	API/FAPI		
<a href="#">VioWrtCharStr</a>		<a href="#">VIOCALLS</a>	API/FAPI		
<a href="#">VioWrtCharStrAtt</a>		<a href="#">VIOCALLS</a>	API/FAPI		
<a href="#">VioWrtNAttr</a>		<a href="#">VIOCALLS</a>	API/FAPI		
<a href="#">VioWrtNCell</a>		<a href="#">VIOCALLS</a>	API/FAPI		
<a href="#">VioWrtNChar</a>		<a href="#">VIOCALLS</a>	API/FAPI		
<a href="#">VioWrtTTY</a>		<a href="#">VIOCALLS</a>	API/FAPI		
<a href="#">VioScrLock</a>		<a href="#">VIOCALLS</a>	API/FAPI		

## Limitations

## Real Mode

- max. 640K memory
- no virtual address space
- no multitasking
- no undocumented OS services
- If the filename of an executable produced by BIND is changed, then it will not run under DOS 2.1.

## Protected Mode

- 16 MB memory
- 1GB virtual address space

# Notes

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

From:

<http://www.osfree.org/doku/> - **osFree wiki**

Permanent link:

<http://www.osfree.org/doku/doku.php?id=en:docs:fapi&rev=1606230462>

Last update: **2020/11/24 15:07**

