



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](#)

## DosChDir

This call defines the current directory for the requesting process.

### Syntax

```
DosChDir (DirName, Reserved)
```

### Parameters

- DirName ([PSZ](#)) - input : Address of the ASCIIZ directory path name.
- Reserved ([ULONG](#)) - input : Reserved and must be set to zero.

### Return Code

rc ([USHORT](#)) - return

Return code descriptions are:

- 0 NO\_ERROR
- 2 ERROR\_FILE\_NOT\_FOUND
- 3 ERROR\_PATH\_NOT\_FOUND
- 5 ERROR\_ACCESS\_DENIED
- 8 ERROR\_NOT\_ENOUGH\_MEMORY
- 26 ERROR\_NOT\_DOS\_DISK
- 87 ERROR\_INVALID\_PARAMETER
- 108 ERROR\_DRIVE\_LOCKED
- 206 ERROR\_FILENAME\_EXCED\_RANGE

### Remarks

The directory path is not changed if any member of the path does not exist. The current directory changes only for the requesting process.

For FSDs, the case of the current directory is set according to the DirName passed in, not according to the case of the directories on disk. For example, if the directory "c:\bin" is created and DosChDir is called with DirName "c:\bin", the current directory returned by [DosQCurDir](#) will be "c:\bin".

Programs running without the NEWFILES bit set are allowed to DosChDir to a non-8.3 filename format directory.

[DosQSysInfo](#) must be used by an application to determine the maximum path length supported by OS/2. The returned value should be used to dynamically allocate buffers that are to be used to store paths.

## Example Code

### C Binding

```
#define INCL_DOSFILEMGR

USHORT rc = DosChDir(DirName, Reserved);

PSZ      DirName;      /* Directory path name */
ULONG    0;            /* Reserved (must be zero) */

USHORT    rc;          /* return code */
```

This example changes directories to \os2\system.

```
#define INCL_DOSFILEMGR

#define PATH "\\os2\system"
#define RESERVED 0L

USHORT rc;

rc = DosChDir(PATH, RESERVED);
```

### MASM Binding

```
EXTRN DosChDir:FAR
INCL_DOSFILEMGR EQU 1

PUSH@  ASCIIZ  DirName      ;Directory path name string
PUSH   DWORD   0           ;Reserved (must be zero)
CALL   DosChDir
```

Returns WORD

## Note

Text based on <http://www.edm2.com/index.php/DosChDir>

Family API		
DOS	Process Manager	DosBeep DosExit DosSleep DosExecPgm
	File Manager	DosChDir DosChgFilePtr DosClose DosDelete DosDupHandle DosMkDir DosMove DosQCurDir DosQCurDisk DosSetFileMode DosOpen DosQFileInfo DosRead DosQFileMode DosQFSInfo DosQVerify DosRmDir DosSelectDisk DosFindClose DosFindFirst DosFindNext DosSetFileInfo DosSetVerify DosWrite DosFileLocks DosSetFHandState DosNewSize DosBufReset DosQFHandState DosSetFSinfo DosShutdown
	Memory Manager	DosFreeSeg DosSubAlloc DosSubFree DosSubSet DosAllocHuge DosAllocSeg DosReallocHuge DosReallocSeg DosGetHugeShift DosCreateCSAlias
	NLS	DosCaseMap DosGetCtryInfo DosGetDBCSEv DosSetCtryCode DosGetCollate DosGetMessage DosInsMessage DosPutMessage
	Date and Time	DosSetDateTime DosGetDateTime
	Devices	DosDevConfig DosDevIOCtl DosDevIOCtl2
	Signals	DosHoldSignal DosSetSigHandler
	Misc	BadDynLink DosGetEnv DosGetMachineMode DosGetVersion DosError DosErrClass DosSetVec
KBD	KbdCharIn KbdFlushBuffer KbdGetStatus KbdSetStatus KbdStringIn KbdPeek	
VIO	VioGetBuf VioGetConfig VioGetCurPos VioGetCurType VioGetPhysBuf VioReadCellStr VioReadCharStr VioScrollUp VioScrollDn VioScrollLf VioScrollRt VioScrUnLock VioSetCurPos VioSetCurType VioSetMode VioGetMode VioShowBuf VioWrtCellStr VioWrtCharStr VioWrtCharStrAtt VioWrtNAttr VioWrtNCell VioWrtNChar VioWrtTTY VioScrLock VioPopUp	
Tools	BIND	
Modules	DOSCALLS.DLL VIOCALLS.DLL KBDCALLS.DLL MSG.DLL	
Libraries	API.LIB OS2386.LIB FAPI.LIB DOSCALLS.LIB SUBCALLS.LIB	

2018/08/25 15:05 · prokushev · 0 Comments

From:

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

Permanent link:

<http://osfree.ru/doku/doku.php?id=en:docs:fapi:doschdir>

Last update: **2021/09/17 03:42**

