



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](#)

This call creates a subdirectory that has extended attributes associated with it.

### Syntax

DosMkDir2 (DirName, EABuf, Reserved)

### Parameters

;DirName (PSZ) - input : Address of the ASCIIZ directory path name, which may or may not contain a drive specification. If no drive is specified, the current drive is assumed. :[DosQSysInfo](#) is called by an application during initialization to determine the maximum path length allowed by OS/2. ;EABuf (PEAOP) - input/output: Address of the extended attribute buffer, which contains an [EAOP](#) structure. ;Reserved (ULONG) - input: Reserved and must be set to zero.

### Return Code

;rc (USHORT) - return:Return code descriptions are: \*0 NO\_ERROR \*3 ERROR\_PATH\_NOT\_FOUND \*5 ERROR\_ACCESS\_DENIED \*26 ERROR\_NOT\_DOS\_DISK \*87 ERROR\_INVALID\_PARAMETER \*108 ERROR\_DRIVE\_LOCKED \*206 ERROR\_FILENAME\_EXCED\_RANGE \*254 ERROR\_INVALID\_EA\_NAME \*255 ERROR\_EA\_LIST\_INCONSISTENT

### Remarks

DosMkDir2 allows an application to define extended attributes for a subdirectory at the time of its creation.

If any subdirectory names in the path do not exist, the subdirectory is not created. Upon return, a subdirectory is created at the end of the specified path.

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.

If a program running with the NEWFILES bit set tries to create a directory with blanks immediately preceding the dot on a FAT drive, the system rejects the name. For example, if c: is a FAT drive, the name "file .txt" is rejected and the name "file.txt" is accepted.

## Bindings

### C

```
<PRE> typedef struct _GEA { /* gea */
```

```
BYTE cbName;          /* name length not including NULL */  
CHAR szName[1];      /* attribute name */
```

```
} GEA;
```

```
typedef struct _GEALIST { /* geal */
```

```
ULONG cbList;        /* total bytes of structure including full list */  
GEA list[1];        /* variable length GEA structures */
```

```
} GEALIST;
```

```
typedef struct _FEA { /* fea */
```

```
BYTE fEA;           /* flags */  
BYTE cbName;       /* name length not including NULL */  
USHORT cbValue;    /* value length */
```

```
} FEA;
```

```
typedef struct _FEALIST { /* feal */
```

```
ULONG cbList;        /* total bytes of structure including full list */  
FEA list[1];        /* variable length FEA structures */
```

```
} FEALIST;
```

```
typedef struct _EAOP { /* eaop */
```

```
PGEALIST fpGEOList; /* general EA list */  
PFEALIST fpFEAList; /* full EA list */  
ULONG oError;
```

```
} EAOP;
```

```
#define INCL_DOSFILEMGR
```

```
USHORT rc = DosMkDir2(DirName, EABuf, Reserved);
```

```
PSZ DirName; /* New directory name string */ PEAOP EABuf; /* Extended attribute buffer */ ULONG 0;  
/* Reserved (must be zero) */
```

```
USHORT rc; /* return code */ </PRE>
```

**MASM**

<PRE> GEA struc

```
gea_cbName    db  ?           ;name length not including NULL
gea_szName    db  1 dup (?)   ;attribute name
```

GEA ends

GEALIST struc

```
geal_cbList   dd  ?           ;total bytes of structure including full list
geal_list     db  size GEA * 1 dup (?) ;variable length GEA structures
```

GEALIST ends

FEA struc

```
fea_fEA       db  ? ;flags
fea_cbName    db  ? ;name length not including NULL
fea_cbValue   dw  ? ;value length
```

FEA ends

FEALIST struc

```
feal_cbList   dd  ?           ;total bytes of structure including full list
feal_list     db  size FEA * 1 dup (?) ;variable length FEA structures
```

FEALIST ends

EAOP struc

```
eaop_fpGEAList dd  ? ;general EA list
eaop_fpFEAList dd  ? ;full EA list
eaop_oError    dd  ? ;
```

EAOP ends

EXTRN DosMkDir2:FAR INCL\_DOSFILEMGR EQU 1

PUSH@ ASCIIZ DirName ;New directory name string PUSH@ OTHER EABuf ;Extended attribute buffer  
PUSH DWORD 0 ;Reserved (must be zero) CALL DosMkDir2

Returns WORD </PRE>

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

Permanent link: <https://www.osfree.ru/doku/doku.php?id=en:docs:fapi:dosmkdir2&rev=1629432937>

Last update: **2021/08/20 04:15**

