 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

DosSetDateTime

This call is used to set the date and time that are maintained by the operating system.

Syntax

```
DosSetDateTime (DateTime)
```

Parameters

- DateTime (PDATETIME) - input : Address of the date and time structure:
  - hours (UCHAR) - Current hour (0-23)
  - minutes (UCHAR) - Current minute (0-59)
  - seconds (UCHAR) - Current second (0-59)
  - hundredths (UCHAR) - Current hundredth of a second (0-99)
  - day (UCHAR) - Current day (1-31)
  - month (UCHAR) - Current month (1-12)
  - year (USHORT) - Current year (1980-2079)
  - timezone (SHORT) - Minutes west of UTC (Universal Time Coordinate -720 to 720).
  - weekday (UCHAR) - Current day of the week. This value is ignored and is calculated from the other parameter information.

Return Code

rc (USHORT) - return

Return code descriptions are:

- 0 NO_ERROR
- 327 ERROR_TS_DATETIME

Remarks

The value of timezone is the difference in minutes between the current time zone and UTC. This is a positive number if earlier than UTC, and negative number if it is later. For Eastern Standard Time this value is 300 (five hours earlier than UTC).

To get the date and time, issue DosGetDateTime. If the application is executing in the OS/2 environment, it is more efficient to obtain these variables by calling DosGetInfoSeg instead of this function. However, applications written to the family API cannot depend on the availability of DosGetInfoSeg.

Not adhering to the limits on any of the parameters results in the return code being set to rc = 327 (ERROR_TS_DATETIME). Also, OS/2 verifies that the day is possible for the month and the year (even for leap year). If the day is not reasonable, OS/2 will also set rc = 327.

Example Code

C Binding

```
typedef struct _DATETIME {  /* date */
  UCHAR   hours;            /* current hour */
  UCHAR   minutes;          /* current minute */
  UCHAR   seconds;          /* current second */
  UCHAR   hundredths;       /* current hundredths of a second */
  UCHAR   day;              /* current day */
  UCHAR   month;            /* current month */
  USHORT  year;             /* current year */
  SHORT   timezone;         /* minutes of time west of UTC */
  UCHAR   weekday;          /* current day of week */
} DATETIME;

#define INCL_DOSDATETIME

USHORT    rc = DosSetDateTime(DateTime);
PDATETIME DateTime;      /* Date/time structure */
USHORT    rc;            /* return code */
```

The following example obtains and prints date and time information. It then changes the system date to 5/10/1987 and prints the updated information.

```
#define INCL_DOSDATETIME
#include <os2.h>

main()
{
  DATETIME   DateTime;       /* Structure to hold date/time info. */
  USHORT     rc;
```

```
  rc = DosGetDateTime(&DateTime);     /* Address of d/t structure */
  printf("Today is %d-%d-%d; the time is %d:%d\n", DateTime.month,
         DateTime.day, DateTime.year, DateTime.hours, DateTime.minutes);
  DateTime.day = 10;
  DateTime.month = 5;
  DateTime.year = 1987;
  printf("The new date is %d-%d-%d; the time is %d:%d\n", DateTime.month,
         DateTime.day, DateTime.year, DateTime.hours, DateTime.minutes);
  rc = DosSetDateTime(&DateTime);    /* Address of d/t structure */
  printf("rc is %d\n", rc);
 }
```

MASM Binding

```
DATETIME struc
  date_hours      db  ? ;current hour
  date_minutes    db  ? ;current minute
  date_seconds    db  ? ;current second
  date_hundredths db  ? ;current hundredths of a second
  date_day        db  ? ;current day
  date_month      db  ? ;current month
  date_year       dw  ? ;current year
  date_timezone   dw  ? ;minutes of time west of UTC
  date_weekday    db  ? ;current day of week
DATETIME ends

EXTRN  DosSetDateTime:FAR
INCL_DOSDATETIME    EQU 1

PUSH@  OTHER   DateTime      ;Date/time structure
CALL   DosSetDateTime
```

Returns WORD

Note

Text based on http://www.edm2.com/index.php/DosSetDateTime_(Legacy)

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