 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

DosSysTrace

Brief

Writes a trace record to the system trace buffer. It provides a high speed event recording mechanism which may be used by PM and non-PM threads in ring 3 and ring 2 and by detached processes.

Syntax

```
APIRET APIENTRY16 DosSysTrace
       (USHORT major, USHORT cBuffer, USHORT minor, PCHAR pBuffer)
```

Parameters

- major (USHORT) input : Major code which identifies the trace record. Range reserved for user is 245-255. Valid range 0-255
- cBuffer (USHORT) input : Length of optional buffer. Valid range
  - 0 - 512 (before 4.0 FP 10 and 3.0 FP 35)
  - 0 - 4099 (from 3.0 FP35 and 4.0 FP10 onwards).
- minor (USHORT) input : Minor code which identifies the trace record. Major-minor code pair should uniquely identify the trace record. Valid range 1 - 255
- pBuffer (PCHAR) input : Pointer to optional buffer. If cBuffer is 0, then pBuffer is ignored.

Return Code

ulrc (APIRET) returns

DosSysTrace returns one of the following values

- 0 NO_ERROR
- 150 ERROR_SYSTEM_TRACE

Remarks

DosSysTrace creates a trace record that includes the following items:

- Header Major code, minor code, time stamp, PID of logging process
- Optional System Data Controlled by the TRACE command
- Optional User Data Specified by the pBuffer parameter

To check is tracing enabled use DosGetInfoSeg. GlobalInfoSeg contains 256bit flags area. See DosGetInfoSeg for more info.

Example Code

```
int main(int argc, char *argv[], char *envp[]){
   APIRET rc=0;           /* default return code */
   USHORT major=255;        /* default major code */
   USHORT minor=1;          /* default minor code */
   USHORT cBuffer=0;        /* default buffer length */
   PCHAR  pBuffer=NULL;     /* default buffer address */

   if (argc>1)
   {
      pBuffer = argv[1];
      cBuffer = strlen(argv[1]);
   }

   if (argc>2) major = atol(argv[2]);
   if (argc>3) minor = atol(argv[3]);

   rc = Dos16SysTrace(major, cBuffer, minor, pBuffer);

   if (rc) printf("DosSysTrace retuned rc=%u\n", rc);

   return rc;
}
```

Related Functions

- DosDumpProcess
- DosForceSystemDump

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