



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](#)

DosGetInfoSeg

This call returns the address of a global and local information segment, specific to a process.

Syntax

DosGetInfoSeg (GlobalSeg, LocalSeg)

Parameters

- GlobalSeg (PSEL) - output : Address of the global information segment structure, as defined below:
 - time (ULONG): Time in seconds since 1/1/1970.
 - millisecs (ULONG): Time in milliseconds.
 - hours (UCHAR): Current hour.
 - minute (UCHAR): Current minute.
 - seconds (UCHAR): Current second.
 - hundredsec (UCHAR): Current hundredth of a second.
 - timezone (USHORT): Minutes from UTC; if hex FFFFH, timezone is undefined.
 - interval (USHORT): Timer interval in tenths of milliseconds.
 - day (UCHAR): Day.
 - month (UCHAR): Month.
 - year (USHORT): Year.
 - weekday (UCHAR): Day-of-week:

Value	Definition
0	Sunday
1	Monday
2	Tuesday
3	Wednesday
4	Thursday
5	Friday
6	Saturday

- majorversion (UCHAR): Major version number.
- minorversion (UCHAR): Minor version number.

- revision (UCHAR): Revision letter.
- currentsession (UCHAR): Current foreground full-screen session.
- maxnumsessions (UCHAR): Maximum number of full-screen sessions.
- hugeshift (UCHAR): Shift count for huge segments.
- protmodeind (UCHAR): Protect-mode-only indicator:

Value	Definition
0	DOS mode and OS/2 mode.
1	OS/2 mode only.

- lastprocess (USHORT) : Process ID of the current foreground process.
- dynvarflag (UCHAR) : Dynamic variation flag:

Value	Definition
0	Absolute
1	Enabled

- maxwait (UCHAR) : Maximum wait in seconds.
- mintimeslice (USHORT) : Minimum timeslice in milliseconds.
- maxtimeslice (USHORT) : Maximum timeslice in milliseconds.
- bootdrive (USHORT) : Drive from which the system was booted:

Value	Definition
1	Drive A.
2	Drive B.
.	
.	
.	
n	Drive n.

- traceflags (UCHAR): 32 system trace major code flags. Each bit corresponds to a trace major code, hex 00-FFH. The most significant bit (left-most) of the first byte corresponds to major code hex 00H.

Value	Definition
0	Trace disabled.
1	Trace enabled.

- maxtextsessions (UCHAR): Maximum number of VIO windowable sessions.
- maxpmsessions (UCHAR): Maximum number of Presentation Manager sessions.
- LocalSeg (PSEL) - output : Address of the selector for the local information segment structure, as defined below:
 - processid (PID): Current process ID.
 - parentprocessid (PID): Parent process ID.
 - threadprty (USHORT): Priority of current thread.
 - threadid (TID): Current thread ID.
 - sessionid (USHORT): Current session ID.
 - procstatus (UCHAR): Process status.
 - unused (UCHAR): Unused.
 - foregroundprocess (BOOL): Current process is in foreground (has keyboard focus). Value -1 implies yes, 0 implies no.

- typeProcess (UCHAR): Type of process:

Value	Definition
0	Full screen protect mode session.
1	Requires real mode.
2	VIO windowable protect mode session.
3	Presentation Manager protect mode session.
4	Detached protect mode process.

- unused (UCHAR): Unused.
- environmentsel (SEL): Environment selector.
- cmdlineoff (USHORT): Command line offset in the segment addressed by environmentsel.
- dataseglen (USHORT): Length of data segment in bytes.
- stacksize (USHORT): Stack size in bytes.
- heapsize (USHORT): Heap size in bytes.
- hmodule (HMODULE): Module handle.
- dssel (SEL): Data segment selector.

Return Code

rc (USHORT) - return:Return code description is:

- 0 NO_ERROR

Remarks

Items of general interest are kept in the global information segment. Items of information specific to a particular process are kept in the local information segment. This information can be directly read by the application program. Both of these segments are defined as read-only segments. The application program cannot modify this data.

Assuming n_1 , n_2 , and n_3 are the maximum number of full-screen sessions, VIO-windowable sessions, and Presentation Manager sessions, the first 0 through (n_1-1) session numbers are assigned to full-screen sessions. The next n_2 session numbers are assigned to VIO-windowable sessions, and the next n_3 session numbers are assigned to Presentation Manager sessions. Session numbers in the range $(n_1+n_2+n_3)$ through 255 are reserved. Applications should use $(n_1+n_2+n_3-1)$ as an upper boundary. Applications should not assume that all session numbers starting with (n_1+n_2) and higher are Presentation Manager sessions.

The application program must be careful when referencing the date or time fields in the global information segment. A timer interrupt can be received by the system in between the instructions that read the individual fields, changing the data in these fields. For example, if the time is currently 23:59:59 on Tuesday, 6/2/87, the program can read the hours field (23). A timer interrupt can now be received, changing the time to 00:00:00 on Wednesday, 6/3/87. The program reads the rest of the time field (0 minutes) and the date field. The program would then think the current time and date is 23:00:00 on Wednesday, 6/3/87, which is incorrect.

The application program should read all time and date fields it uses as quickly as possible. It can then

compare the least significant time field it uses (milliseconds, hundredths, seconds, or minutes) against the current value in the global information segment. If the value has not changed, the rest of the information is valid. If the value has changed, the program time or date information should be read again, as the information is updated while the program reads it.

Bindings

C

```
typedef struct _GINFOSEG {
    ULONG    time;           /* time in seconds */
    ULONG    msec;         /* milliseconds */
    UCHAR    hour;         /* hours */
    UCHAR    minutes;      /* minutes */
    UCHAR    seconds;      /* seconds */
    UCHAR    hundredths;   /* hundredths */
    USHORT   timezone;     /* minutes from UTC */
    USHORT   cusecTimerInterval; /* timer interval (units = 0.0001 seconds) */
    UCHAR    day;          /* day */
    UCHAR    month;        /* month */
    USHORT   year;         /* year */
    UCHAR    weekday;     /* day of week */
    UCHAR    uchMajorVersion; /* major version number */
    UCHAR    uchMinorVersion; /* minor version number */
    UCHAR    chRevisionLetter; /* revision letter */
    UCHAR    sgCurrent;    /* current foreground session */
    UCHAR    sgMax;        /* maximum number of sessions */
    UCHAR    cHugeShift;   /* shift count for huge elements */
    UCHAR    fProtectModeOnly; /* protect mode only indicator */
    USHORT   pidForeground; /* pid of last process in foreground session */
*/
    UCHAR    fDynamicSched; /* dynamic variation flag */
    UCHAR    csecMaxWait;   /* max wait in seconds */
    USHORT   cmsecMinSlice; /* minimum timeslice (milliseconds) */
    USHORT   cmsecMaxSlice; /* maximum timeslice (milliseconds) */
    USHORT   bootdrive;    /* drive from which the system was booted */
    UCHAR    amecRAS[32];  /* system trace major code flag bits */
    UCHAR    csgWindowableVioMax; /* maximum number of VIO windowable sessions */
*/
    UCHAR    csgPMMMax;    /* maximum number of pres. services sessions */
*/
} GINFOSEG;

typedef struct _LINFOSEG {
    PID      pidCurrent;    /* current process id */
    PID      pidParent;    /* process id of parent */
    USHORT   prtyCurrent;  /* priority of current thread */
    TID      tidCurrent;   /* thread ID of current thread */
}
```

```

USHORT  sgCurrent;          /* session */
UCHAR   rfProcStatus;      /* process status */
UCHAR   dummy1;
BOOL    fForeground;       /* current process has keyboard focus */
UCHAR   typeProcess;       /* process type */
UCHAR   dummy2;
SEL     selEnvironment;    /* environment selector */
USHORT  offCmdLine;        /* command line offset */
USHORT  cbDataSegment;     /* length of data segment */
USHORT  cbStack;           /* stack size */
USHORT  cbHeap;            /* heap size */
HMODULE hmod;              /* module handle of the application */
SEL     selDS;             /* data segment handle of the application */
} LINFOSEG;

```

```
#define INCL_DOSINFOSEG
```

```

USHORT  rc = DosGetInfoSeg(GlobalSeg, LocalSeg);

PSEL    GlobalSeg;         /* Address to place global segment (selector) */
PSEL    LocalSeg;         /* Address to place local segment (selector) */

USHORT  rc;                /* return code */

```

MASM

```

GINFOSEG struc
    gis_time          dd ? ;time in seconds
    gis_msecs        dd ? ;milliseconds
    gis_hour          db ? ;hours
    gis_minutes       db ? ;minutes
    gis_seconds       db ? ;seconds
    gis_hundredths    db ? ;hundredths
    gis_timezone      dw ? ;minutes from UTC
    gis_cusecTimerInterval dw ? ;timer interval (units = 0.0001 seconds)
    gis_day           db ? ;day
    gis_month         db ? ;month
    gis_year          dw ? ;year
    gis_weekday       db ? ;day of week
    gis_uchMajorVersion db ? ;major version number
    gis_uchMinorVersion db ? ;minor version number
    gis_chRevisionLetter db ? ;revision letter
    gis_sgCurrent     db ? ;current foreground session
    gis_sgMax         db ? ;maximum number of sessions
    gis_cHugeShift    db ? ;shift count for huge elements
    gis_fProtectModeOnly db ? ;protect mode only indicator
    gis_pidForeground dw ? ;pid of last process in foreground session
    gis_fDynamicSched db ? ;dynamic variation flag
    gis_csecMaxWait   db ? ;max wait in seconds
    gis_cmsecMinSlice dw ? ;minimum timeslice (milliseconds)

```

```
gis_cmsecMaxSlice    dw ? ;maximum timeslice (milliseconds)
gis_bootdrive        dw ? ;drive from which the system was booted
gis_amecRAS          db 32 dup (?) ;system trace major code flag bits
gis_csgWindowableVioMax db ? ;maximum number of VIO windowable sessions
gis_csgPMMMax        db ? ;maximum number of pres. services sessions
```

GINFOSEG ends

LINFOSEG struc

```
lis_pidCurrent      dw ? ;current process id
lis_pidParent       dw ? ;process id of parent
lis_prtyCurrent     dw ? ;priority of current thread
lis_tidCurrent      dw ? ;thread ID of current thread
lis_sgCurrent       dw ? ;session
lis_rfProcStatus    db ? ;process status
lis_dummy1          db ? ;
lis_fForeground     dw ? ;current process has keyboard focus
lis_typeProcess     db ? ;process type
lis_dummy2          db ? ;
lis_selEnvironment  dw ? ;environment selector
lis_offCmdLine      dw ? ;command line offset
lis_cbDataSegment   dw ? ;length of data segment
lis_cbStack         dw ? ;stack size
lis_cbHeap          dw ? ;heap size
lis_hmod            dw ? ;module handle of the application
lis_selDS           dw ? ;data segment handle of the application
```

LINFOSEG ends

EXTRN DosGetInfoSeg:FAR

INCL_DOSINFOSEG EQU 1

PUSH@ WORD GlobalSeg ;Global segment selector (returned)

PUSH@ WORD LocalSeg ;Local segment selector (returned)

CALL DosGetInfoSeg

Returns WORD

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://185.82.219.184/doku/> - **osFree wiki**

Permanent link: <http://185.82.219.184/doku/doku.php?id=en:docs:fapi:dosgetinfoseg&rev=1634451356>

Last update: **2021/10/17 06:15**

