 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

DosCaseMap

This call performs case mapping on a string of binary values that represent ASCII characters.

Syntax

```
DosCaseMap (Length, Country, BinaryString)
```

Parameters

- Length (USHORT) - input : Length of the string of binary values to be case mapped.
- Country (PCOUNTRYCODE) - input/output : Address of the country information structure:
  - country (USHORT) : Country code identifier is a binary value of the selected country code. 0 is the default country code.
  - codepage (USHORT) : Code page identifier is a binary value of the selected code page. 0 is the code page of the current process.
- BinaryString (PCHAR) - input/output : Address of a string of binary characters to be case mapped. They are case mapped in place so the results appear in BinaryString and the input is destroyed.

Return Code

rc (USHORT) - return

Return code descriptions are:

- 0 NO_ERROR
- 396 ERROR_NLS_NO_COUNTRY_FILE
- 397 ERROR_NLS_OPEN_FAILED
- 398 ERROR_NO_COUNTRY_OR_CODEPAGE
- 399 ERROR_NLS_TABLE_TRUNCATED

Remarks

DosCaseMap is mainly used to map a lower case character string to an upper case character string. Unless the user replaces the country information file, DosCaseMap only does the conversion from lower case to upper case.

The case map information is taken from the country information file. See the COUNTRY statement in the `IBM Operating System/2 Version 1.2 Command Reference` for information on how to specify the country information file.

If countrycode is 0, the case mapping is performed using the information for the country specified in the COUNTRY statement in CONFIG.SYS.

If countrycode is not 0, the case mapping is performed using the information for that country.

If the code page identifier is 0, the case mapping is performed using the information for the current process code page. Refer to DosSetCp and the CHCP command in the `IBM Operating System/2 Version 1.2 Command Reference` for information on setting the process code page. If codepage is not 0, the case mapping is performed using the information for that code page.

The returned country dependent information may be for the default country and current process code page or for a specific country and code page.

Bindings

C Binding

```
typedef struct _COUNTRYCODE {   /* ctryc */

  USHORT country;               /* country code */
  USHORT codepage;              /* code page */

} COUNTRYCODE;

#define INCL_DOSNLS

USHORT  rc = DosCaseMap(Length, Structure, BinaryString);

USHORT           Length;        /* Length of string to case map */
PCOUNTRYCODE     Structure;     /* Input data structure */
PCHAR            BinaryString;  /* Address of binary string */

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

MASM Binding

```
COUNTRYCODE struc
```

```
  ctryc_country   dw  ? ;country code
  ctryc_codepage  dw  ? ;code page

COUNTRYCODE ends

EXTRN  DosCaseMap:FAR
INCL_DOSNLS         EQU 1

PUSH   WORD    Length        ;Length of string to case map
PUSH@  OTHER   Structure     ;Input data structure
PUSH@  OTHER   BinaryString  ;Binary string
CALL   DosCaseMap
```

</code>

Returns WORD

Example Code

This example case maps a string for the default country and code page 850.

```
#define INCL_DOSNLS

#define CURRENT_COUNTRY 0
#define NLS_CODEPAGE 850

COUNTRYCODE Country;
CHAR        BinString[30];
USHORT      rc;


   Country.country = CURRENT_COUNTRY;          /* Country code */
   Country.codepage = NLS_CODEPAGE;            /* Code page */
   strcpy(BinString,"Howdy");                  /* String to map */
   rc = DosCaseMap(sizeof(BinString),          /* Length of string */
                   &Country,                   /* Input data structure */
                   BinString);                 /* String */
```

Note

Text based on http://www.edm2.com/index.php/DosCaseMap_(FAPI)

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://osfree.su/doku/ - **osFree wiki**

Permanent link:
https://osfree.su/doku/doku.php?id=en:docs:fapi:doscasemap&rev=1631870509

Last update: **2021/09/17 09:21**