# FOXTOOLS

To print these notes, choose Print Topic from the File menu.

FOXTOOLS is a Visual FoxPro API library that exposes:

- Windows DLLs for use in Visual FoxPro for Windows.
- Macintosh system functions for use in Visual FoxPro for Macintosh.

Functions in the FOXTOOLS library allow you to set and retrieve file information, manipulate paths and file names, use system alerts, and perform many other functions.

**Note** The FOXTOOLS functions are not supported by Microsoft Product Support Services (PSS) either electronically or via telephone. Many of these functions are included for backward compatibility with FoxPro version 2.6. Some functions behave differently depending on the platform for which they are used.

# Where FOXTOOLS gets installed

## Visual FoxPro for Windows

When you install Visual FoxPro for Windows, Foxtools.fll is installed in the VFP folder. To use the FOXTOOLS functions, issue the following command:

```
SET LIBRARY TO FOXTOOLS
```

**Note** API routines are documented in the Professional Reference section of Help under API Library Construction. These routines map to editor and window functions and allow you to extend Visual FoxPro using the C programming language.

## Visual FoxPro for Macintosh

When you install Visual FoxPro for Macintosh, Foxtools.mlb is installed in the "System:Extensions" folder and FoxTools.CFM and FoxTools.SLM library files are installed in the VFP folder. FoxTools.CFM is provided for use with PowerPC and 68K Macintosh computers, while FoxTools.SLM works with applications built for only 68k Macintosh computers. To use the FOXTOOLS functions, issue the following command:

```
SET LIBRARY TO FOXTOOLS.SLM  && for 68K Macintosh applications
   -OR-
SET LIBRARY TO FOXTOOLS.CFM  && For PowerMacintosh or 68K Macintosh
applications
```

# Function Reference

# RegFn( ), RegFn32( ), and CallFn( )

The FOXTOOLS API library allows Visual FoxPro programs to call any Windows DLL functions that:

- take the following arguments: integer, long, float, double, string/buffer. These can be passed by reference or by value.
- return the following types: integer, long, float, double, string/buffer. These are returned by value only.

RegFn( ) and RegFn32( ) register a function and the arguments it takes, and CallFn( ) calls a registered function. The Windows DLL functions are documented in the books that come with the Windows Software Development Kit. The Windows SDK is also included with Microsoft C/C++ 7.0 and Visual C++.

**Note**

- FLL32_16.DLL is used for calling 16-bit DLLs under win32s. It uses the universal thunk to call 16 bit DLLs.
- DDEREG.EXE is used for calling 16-bit DLLs under win32. It uses DDE to call 16-bit DLLs.

# RegFn( )

Available in Visual FoxPro for Windows only.

Registers a Windows API function and the arguments that function takes.

**Note**

- On 16-bit Windows platforms, RegFn( ) calls 16-bit DLLs.
- On 32-bit Windows platforms, RegFn( ) calls 32-bit DLLs (if you pass the default number of arguments).

**Syntax**

*RegFn(cFunctionName, cArgTypes, cReturnType, cDLLName)*
*cFunctionName*

Specifies the name or the ordinal value of the function to be registered.

*cArgTypes*

Specifies a string of characters which describe what arguments the function will accept. The following values are allowed:

| Value | Description |
|---|---|
| I | Integer (32-bit) |
| L | Long |
| C | String of characters |
| F | Floating point number |
| D | Double precision floating point number |
| S | Short (16-bit) integer |

Each argument type may be preceded by @ to indicate that the argument is passed by reference. The default is to pass arguments by value.

*cReturnType*

Specifies the return type of the function. The values allowed are the same as given above for the argument except that return by reference is not allowed.

*cDLLName*

Specifies the name of the DLL library that contains the function. This is optional if the function is in one of the libraries on the Win16 (User.exe, Krnl386.exe, and Gdi.exe) or Win32 (Kernel32.dll, Gdi32.dll User32.dll, Advapi32.dll, and Mpr.dll) platforms.

See the Windows SDK function LoadLibrary for details on how the DLL name is searched.

**Return Type**

Numeric

**Remarks**

If successful, RegFn( ) returns a number which can be used to reference the function in future calls to

CallFn( ).

Returns -1 if the library couldn't be opened, and will additionally display a message box.

The same function can be registered more than once. This allows functions that take different arguments to be used by declaring them again with different argument types.

# RegFn32( )

Available in Visual FoxPro for Windows only.

Registers a Windows API function and the arguments that function takes. RegFn32( ) always calls 32-bit DLLs.

### Syntax

RegFn(*cFunctionName, cArgTypes, cReturnType, cDLLName*)
*cFunctionName*

Specifies the name or the ordinal value of the function to be registered.

*cArgTypes*

Specifies a string of characters which describe what arguments the function will accept. The following values are allowed:

| Value | Description |
|---|---|
| I | Integer (32-bit) |
| L | Long |
| C | String of characters |
| F | Floating point number |
| D | Double precision floating point number |
| S | Short (16-bit) integer |

Each argument type may be preceded by @ to indicate that the argument is passed by reference. The default is to pass arguments by value.

*CReturnType*

The return type of the function. The values allowed are the same as given above for the argument except that return by reference is not allowed.

*CDLLName*

Specifies the name of the 32-bit DLL library that contains the function. This is optional if the function is in one of the libraries on Win16 (User.exe, Krnl386.exe, and Gdi.exe) or Win32 (Kernel32.dll, Gdi32.dll, User32.dll, Advapi32.dll, and Mpr.dll) platforms.

See the Windows SDK function LoadLibrary for details on how the DLL name is searched.

### Return Type

Numeric

### Remarks

If successful, RegFn32( ) returns a number which can be used to reference the function in future calls to CallFn( ).

Returns -1 if the library couldn't be opened, and will additionally display a message box.

The same function can be registered more than once. This allows functions that take different

arguments to be used by declaring them again with different argument types.

# CallFn( )

Available in Visual FoxPro for Windows only.

Calls a registered function and returns the value that the function returned, using the type declared by ReturnType in RegFn call.

### Syntax

CallFN(*nFunctionHandle*, ***Arg1***[, ***Arg2***[, ...]])
*nFunctionHandle*

The function handle from a previous call to RegFn( ) or RegFn32( ).

***Arg1, Arg2, ....***

Arguments required by the function referenced by *FnNum*. You must pass as many arguments as were declared when the function was registered or an error occurs.

All arguments must match their declared type as follows:

- F, D - must be a floating point number
- I, L - must be an integer
- C - must be a string passed by value, or 0 (zero). If 0, a null pointer is passed.

### Return Types

User-defined in RegFn( ) or RegFn32( ).

# FOXTOOLS Functions A - Z

# AddBS( )

Adds a backslash (if needed) to a path expression.

### Syntax

AddBS(*cPath*)
*cPath*

Specifies the path name to which to add the backslash.

### Return Type

Character

# CleanPath( )

Returns a corrected filename (best guess) for an invalid filename and removes spaces, invalid characters, duplicate backslashes, and so on.

### Syntax

CleanPath(*cFilename*)
*cFilename*

Specifies the filename to be cleaned.

### Return Types

Character

# CloseClip( )

Available in Visual FoxPro for Windows only.

Closes the Clipboard opened previously with OpenClip( ).

**Syntax**

CloseClip( )

**Return Type**

Logical

**Remarks**

The return value reports success (.T.) or failure (.F.) of the command. Almost all Clip functions rely on opening the Clipboard before using the function. _CLIPTEXT can be used to access the contents of the Windows Clipboard.

**Note** This function maps to the Windows SDK functions that is similarly named. For more information, see the Windows SDK documentation.

# CountClipF( )

Available in Visual FoxPro for Windows only.

Retrieves the number of different data formats currently on the Windows Clipboard.

**Syntax**

CountClipF( )

**Return Type**

Numeric

**Remarks**

Almost all Clip functions rely on opening the Clipboard before using the function. You can use _CLIPTEXT to access the contents of the Windows Clipboard.

**Note** This function maps to the Windows SDK functions that is similarly named. For more information, see the Windows SDK documentation.

# DefaultExt( )

Returns a file name with a new extension if one doesn't already exist.

**Syntax**

DefaultExt(*cFilename, cDefault*)
*cFilename*

Specifies the filename (with or without a path or extension) to be returned.

*Cdefault*

Specifies the default extension without a period.

**Return Type**

Character

# DriveType( )

Available in Visual FoxPro for Windows only.

Returns the type of the specified drive.

### Syntax

DriveType(*cDrive*)
*cDrive*

The drive designator. The colon in drive names (for example, "C:") is optional.

### Return Types

Numeric

### Remarks

The following table explains the number DriveType( ) returns and the corresponding drive type description.

| Number | Drive type |
|---|---|
| 0 | No type |
| 2 | Floppy disk |
| 3 | Hard disk |
| 4 | Removable drive or network drive |
| 5 | CD-ROM |
| 6 | RAM disk (1) |

(1) Because there are many different types of RAM disks, you might get inconsistent return results.

# EmptyClip( )

Available in Visual FoxPro for Windows only.

Empties the Clipboard and frees handles to data in the Clipboard, then assigns ownership of the Clipboard to the window in which the Clipboard is open.

### Syntax

EmptyClip( )

### Return Type

Logical

### Remarks

Almost all Clip functions rely on opening the Clipboard before using the function. You can use _CLIPTEXT to access the contents of the Windows Clipboard.

**Note** This function maps to the Windows SDK functions that is similarly named. For more information, see the Windows SDK documentation.

# EnumClipFm( )

Available in Visual FoxPro for Windows only.

Enumerates the formats found in a list of available formats that belong to the Clipboard.

**Syntax**

EnumClipFm(*nFormat*)
*nFormat*

Specifies a Clipboard format.

To determine the available clipboard formats, set *nFormat* to 0; EnumClipFm( ) will return the first available Clipboard format. For subsequent calls to EnumClipFm( ), set *nFormat* to the return value of the previous EnumClipFm( ) call.

**Return Type**

Numeric

**Remarks**

Each call to EnumClipFm( ) specifies a known available format; the function returns the format that appears next in the list.

Almost all Clip functions rely on opening the Clipboard before using the function. You can use _CLIPTEXT to access the contents of the Windows Clipboard.

**Note** This function maps to the Windows SDK functions that is similarly named. For more information, see the Windows SDK documentation.

# ForceExt( )

Returns a string with the old file name extension replaced by a new extension.

**Syntax**

ForceExt(*cFilename*, *cExtension*)
*cFilename*

Specifies the filename (with or without a path or extension) which will get a new extension.

*cExtension*

Specifies the new extension (without a period) for *cFilename*.

**Return Type**

Character

# ForcePath( )

Returns a file name with a new path name substituted for the old one.

**Syntax**

ForceExt(*cFilename*, *cPath*)
*cFilename*

Specifies the filename (with or without a path or extension) which will get a new path.

*cPath*

Specifies the new path for *cFilename*.

**Return Type**

Character

# FoxToolVer( )

Returns the version number of the Foxtools library.

### Syntax

FoxToolVer( )

### Return Type

Character

# FxAlert( )

Available in Visual FoxPro for Macintosh only; in Visual FoxPro for Windows, use the MsgBox( ) function.

Displays an alert and waits for a user response.

### Syntax

FxAlert(*nIconNumber*, *nResourceNumber*,
*nHorizontal*, *nVertical* [, cText1 [, cText2] ... ]

*nIconNumber*

Specifies a number that indicates the icon to be displayed on the alert:

| | |
|---|---|
| 0 | Caution |
| 1 | Stop |
| 2 | Note |

*nResourceNumber*

Specifies a resource number, which indicates what buttons appear on the alert and which one is the default button. For example, resource number 259 contains the buttons "Yes" and "No" with the focus initially on the "Yes" button.

The following table shows the buttons displayed on each of the twelve alerts with the default focus button shown in parentheses.

| Resource # | Button 1 | Button 2 | Button 3 | Comment |
|---|---|---|---|---|
| 257 | (Continue) | | | |
| 258 | (OK) | | | |
| 259 | (Yes) | No | | |
| 260 | Yes | (No) | | |
| 261 | (Yes) | No | Cancel | |
| 262 | Yes | (No) | Cancel | |
| 263 | (Continue) | | | Larger alert than 257 |
| 264 | (OK) | | | Larger alert than 258 |
| 265 | (Yes) | No | | Larger alert than 259 |
| 266 | Yes | (No) | | Larger alert than 260 |
| 267 | (Yes) | No | Cancel | Larger alert than 261 |
| 268 | Yes | (No) | Cancel | Larger alert |

than 262

**Note** Foxtools contains the same twelve alerts provided in the FOXUSER file of FoxBASE+ for Macintosh. In Foxtools, the resource numbers are 257 through 268 instead of 1 through 12 as in FoxBASE+ for Macintosh.

*nHorizontal, nVertical*

Specify the coordinates, measured in pixels from the top left of the screen, at which the alert will be placed. Use a value of -1 for either *nHorizontal* or *nVertical* to accept the default position.

*cText1, cText2, ...*

Specifies optional text that will appear on the alert.

**Remarks**

The FxAlert( ) function displays an alert and waits for a user response. One of three icons will be displayed on the alert, based on the type of alert that you specify in *nIconNumber*. FxAlert( ) returns a value representing the user exit action.

# FxCreator( )

Available in Visual FoxPro for Macintosh only.

Returns information about registered applications based on a specified file creator type.

**Syntax**

FxCreator(*cCreator*, *nIndex*, @*cArray*)
*cCreator*

Specifies the four-letter file creator label.

*nIndex*

Specifies the numeric index used to determine which application file, based on creation date, on which to return information. A value of 0 returns info from an application file with the most recent creation date. FxCreator( ) will span multiple mounted volumes to check for creators of the specified type.

*cArray*

Specifies a four-element array in which the function will return a parameter block for the specified creator type. The following table lists the contents of *cArray*:

| Elements | Contents |
|---|---|
| [1] | Application name |
| [2] | Application creation date |
| [3] | Application signature |
| [4] | Application parent directory |

**Return Type**

Numeric

**Remarks**

FxCreator( ) returns information regarding other applications registered in the desktop database. This information is useful when designing interapplication communications (for example, OLE Automation), to determine the currently-installed versions of specific applications, or to specify a path for a file to

automate using AppleScript.

FxCreator( ) returns 0 if successful, or -1 if it cannot read application information for the code passed in *cCreator*. If successful, the function returns information about the corresponding application in the *cArray* argument. To read information about all registered applications, use a loop that increments the value of *nIndex* until the function returns -1.

**Note** The Macintosh Toolbox call is PBDTGetAPPL.

### Example

```
DIMENSION aCreators[4]
nSucess = FxCreator("MSWD",0,@aCreators)
DISPLAY MEMO LIKE aCreators
```

# FxFileType( )

Available in Visual FoxPro for Macintosh only.

Returns type and file creator information for a specified file.

### Syntax

FxFileType(*cFilename*, @*cFileType*, @*cFileCreator*)
*cFilename*

Specifies the file name (including path) for which you want to obtain the file type and file creator. The path and file name should be in Macintosh format (path:filename).

If you want to use MS-DOS conventions for paths and filenames, you need to convert the path and filename for Macintosh use by calling SYS(2027) first. For example, SYS(2027,"Macintosh HD:\Microsoft Visual FoxPro\FoxTools") will return "Macintosh HD:Microsoft Visual FoxPro:FoxTools".

*CFileType*

Specifies an existing variable, passed by reference, to which the function will store the file type. The @ in front of the variable name is required.

*CFileCreator*

Specifies an existing variable passed by reference, to which the function will store the file creator. The @ in front of the variable name is required.

### Return Type

Numeric

### Remarks

FxFileType( ) stores the type of a file to the variable *cFileType* and stores the creator of the file to the variable *cFileCreator*. FxFileType( ) returns 0 for no error or an operating system error number.

# FxGDepth( )

Available in Visual FoxPro for Macintosh only.

Returns the number of colors available with the current graphics device.

### Syntax

FxGDepth( )

### Return Type

Numeric

# FxGestalt( )

Available in Visual FoxPro for Macintosh only.

Determines whether a specified extension or control panel is loaded.

### Syntax

FxGestalt(*cGestaltCode, @nReturnValue*)
*cGestaltCode*

Specifies the Gestalt selector as a four-character string using standard codes. The following table lists common gestalt codes for the Macintosh:

| Code | Description |
|---|---|
| ag_v | AppleGuide version |
| ascr | AppleScript attributes |
| ascv | AppleScript version |
| cput | Native CPU type |
| mach | Machine type |
| proc | Processor type |
| qdgx | QuickDraw GX version |
| qtim | QuickTime version |
| ram | Physical RAM size |
| sysa | System architecture |
| sysv | System version |

*nReturnValue*

Specifies a variable in which FxGestalt( ) will return information about the extension specified in *cGestaltCode*.

### Return Type

Numeric

### Remarks

Applications running in Visual FoxPro for Macintosh sometimes include features that require certain extensions to be loaded. Because users might choose not to load certain extensions, the FxGestalt( ) function allows you to test for extensions required in your application. For example, Visual FoxPro for Macintosh generally requires that the extensions for AppleScript, AppleGuide, QuickDraw GX, and QuickTime be loaded.

FxGestalt( ) returns information about the specified extension. The function returns 0 if successful or -1 if the extension could not be found. If the function is successful, it returns additional information about the specified extension in the *nReturnValue* argument.

### Example

Example 1 simply determines whether an extesion is loaded by checking the return value of the function.

```
iData = 0
iSuccess = FxGestalt("ascr",@iData)
IF iSuccess = 0
   * AppleScript is installed - do some code here
ELSE
```

```
   =MESSAGEBOX("You must have AppleScript installed to run this.")
ENDIF
```

Examples 2 and 3 check the value returned in the *nReturnValue* argument for specific information about the extension.

```
iSuccess = FxGestalt("sysa",@iData)
IF m.iData = 2
   * PowerPC architecture
ELSE
   * 68K architecture
ENDIF

iSuccess = FxGestalt("sysv",@iData)
IF m.iData >= 1874
* User has System 7.5.2 or later
ENDIF
```

# FxGetRes( )

Available in Visual FoxPro for Macintosh only.

Retrieves a resource from a file.

### Syntax

FxGetRes(*cFilename, cResourceType, nResourceID, cResourceName, @cResourceValue*)

*cFilename*

Specifies the name of the file containing the resource.

*CResourceType*

Specifies the resource type (for example, "STR "). The resource type specification is case sensitive.

*NResourceID*

Specifies the numeric resource ID of the resource to get. You can specify the resource either by ID or by name; to specify a resource by name, pass false (.F.) in *nResourceID* and pass the name in *nResourceName*.

*cResourceName*

Specifies the resource name of the resource to get.

*@CResourceValue*

Specifies a reference to a variable in which the function should return the value of the specified resource. The value is returned as a Pascal-type string (length byte followed by data).

### Return Type

Numeric

### Remarks

Visual FoxPro for Macintosh stores its registry and options settings in the Visual FoxPro Settings resource fork. The FxGetRes( ) function provides a way for you to to read these settings.

To set resource values, use FxSetRes( )

**Note** You should be familiar with how to handle the resource value returned before calling this function.

FxGetRes( ) returns 0 if successful, or -1 otherwise.

**Example**

```
cFoxFile = SYS(2033,2)+" : Visual FoxPro Settings"
cRes = "STR "
cID = 4680
cValue = ""
IF FILE(m.cFoxFile) AND FxGetRes(m.cFoxFile, m.cRes, m.cID, @cValue) = 0
   sLen = ASC(LEFT(m.cValue, 1))
   ? SUBSTR(cValue, 2, m.sLen)
ENDIF
```

# FxGVolume( )

Available in Visual FoxPro for Macintosh only.

Returns a list of volume names created by the SET VOLUME command.

**Syntax**

FxGVolume( )

**Return Type**

Character

**Remarks**

SET VOLUME maps a DOS drive designator to a Macintosh volume or folder; FxGVolume( ) returns this mapping. For example, the following will return "E Macintosh HD:\Microsoft Visual FoxPro" and any other volume names created by SET VOLUME:

SET VOLUME e TO "Macintosh HD:Microsoft Visual FoxPro"

# FxKeyboard( )

Available in Visual FoxPro for Macintosh only.

Returns the type of keyboard in use.

**Syntax**

FxKeyboard( )

**Return Type**

Numeric

**Remarks**

The FxKeyboard( ) function returns an integer value that indicates the type of keyboard that is currently attached to the system. If the Apple Desktop Bus (ADB) is being used and multiple keyboards are attached, the value returned indicates the keyboard which registered the last keystroke.

| Integer | Keyboard type |
|---|---|
| 0 | unable to determine |
| 1 | MacKbd |
| 2 | MacAndPad |

| | |
|---|---|
| 3 | MacPlusKbd |
| 4 | ExtADBKbd |
| 5 | StdADBKbd |
| 6 | PrtblADBKbd |
| 7 | PrtblISOKbd |
| 8 | StdISOADBKbd |
| 9 | ExtISOADBKbd |
| 10 | ADBKbdII |
| 11 | ADBISOKbdII |
| 12 | PwrBookADBKbd |
| 13 | PwrBookISOADBKbd |

# FxNewFolder( )

Available in Visual FoxPro for Macintosh only.

Creates a new folder.

### Syntax

FxNewFolder(*cFolderName*)
*cFolderName*

Specifies the path and folder name of the new folder. The path and file name should be in Macintosh format (`path:filename`).

If you want to use MS-DOS conventions for the path, you need to convert the path name for Macintosh use by calling SYS(2027) first. For example, SYS(2027,"Macintosh HD:\Microsoft Visual FoxPro\FoxTools") will return "Macintosh HD:Microsoft Visual FoxPro:FoxTools".

### Return Type

Numeric

### Remarks

FxNewFolder( ) returns 0 for no error or returns a Macintosh operating system error number.

# FxSetRes( )

Available in Visual FoxPro for Macintosh only.

Sets a resource to a specified value.

### Syntax

FxSetResource(*cFilename, cResourceType, nResourceId,*
*cResourceName, cResourceValue*)
*cFilename*

Specifies the name of the file containing the resource.

*CResourceType*

Specifies the resource type (for example, "STR "). The resource type specification is case sensitive

*nResourceID*

Specifies the numeric resource ID of the resource to be set. To have the function use a resource name (passed in *cResourceName*) instead of a resource ID, or to have the function assign a resource ID, pass false (.F.) in this argument.

*CResourceName*

Specifies the resource name of the resource to be set.

*CResourceValue*

Specifies the value to which to set the resource. The value must be passed as a a Pascal-type string (length byte followed by data).

**Return Type**

Numeric

**Remarks**

Visual FoxPro for Macintosh stores its registry and options settings in the Visual FoxPro Settings resource fork. The FxSetRes( ) function provides a way for you to set resources. FxSetRes( ) creates a file if none already exists, or adds a resource fork if the file exists but did not already have a resource fork

To read resource values, use FxGetRes( ).

FxSetRes( ) returns 0 if successful, or -1 otherwise.

**Example**

```
cFoxFile = SYS(2033,2)+":Visual FoxPro Settings"
cRes = "STR "
cID = 4680
cValue = CHR(3)+"OFF"
IF FILE(m.cFoxFile)
   =FxSetRes(m.cFoxFile,m.cRes,m.cID,m.cValue)
ENDIF
```

# FxSetType( )

Available in Visual FoxPro for Macintosh only.

Sets the type and creator of a file.

**Syntax**

FxSetType(*cFileName, cFileType, cFileCreator*)
*cFileName*

Specifies the path and file name for which you want to set the file type and file creator.

*cFileType*

Specifies the file type. Since the file type is four bytes long, *cFileType* should be four characters long. Strings longer than four characters are truncated. Strings shorter than four characters are padded with 0 (zero).

*cFileCreator*

Specifies the file creator. Since the file creator is four bytes long, *cFileCreator* should be four characters long. Strings longer than four characters are truncated. Strings shorter than four characters are padded with 0 (zero).

**Return Type**

Numeric

**Remarks**

FxSetType( ) returns 0 for no error or returns a Macintosh operating system error number.

# FxStripLF( )

Available in Visual FoxPro for Macintosh only.

Strips line feed characters from a file.

**Syntax**

FxStripLF(*cFileName*)
*cFileName*

Specifies the path and file name for which you want to remove all the line feed characters. The path and file name should be in Macintosh format (path:filename).

If you want to use MS-DOS conventions for the path, you need to convert the path name for Macintosh use by calling SYS(2027) first. For example, SYS(2027,"Macintosh HD:\Microsoft Visual FoxPro\FoxTools") will return "Macintosh HD:Microsoft Visual FoxPro:FoxTools".

**Return Type**

Logical

**Remarks**

The FxStripLF( ) function opens the specified file, removes all the line feed characters (ASCII character 10), and saves the file under its original name. FxStripLF( ) always returns .T., but generates a standard "File Read Error" (#1104) if it cannot read or find the file referenced in *cFileName*, or generates a "File Write Error" (#1105) if it cannot save the file.

# FxSystem( )

Available in Visual FoxPro for Macintosh only.

Returns the path of the System, Extensions, or Preferences folder.

**Syntax**

FxSystem(*nFolder*)
*nFolder*

Specifies the folder for which the path is returned:

| | |
|---|---|
| 0 | System (default) |
| 1 | Extensions |
| 2 | Preferences |

**Return Type**

Character

# FxVInfo( )

Available in Visual FoxPro for Macintosh only.

Returns information about the specified volume.

**Syntax**

FxVInfo(*nVolIndex, nInfoType*)
*iVolIndex*

Specifies the volume number (the index number of mounted volumes). This number must be greater than 0.

*nInfoType*

Specifies the type of information to be returned:

| | |
|---|---|
| 1 | Returns the volume name |
| 2 | Returns the volume creation time and date as a signed 4-byte integer. Add 2^32 to this value to get the number of seconds since midnight of January 1, 1904. |

**Return Type**

Character, Numeric

**Example**

```
? FxVInfo(1,1)   && returns the name of the first volume

* Reads volume date and converts to DateTime expression
nVolDate = FxVInfo(1,2) + int(2^32)
nVolDate = {1/1/1904 12:00AM} + nVolDate
```

# FxVolume( )

Available in Visual FoxPro for Macintosh only.

Returns a list of mounted volume names.

**Syntax**

FxVolume( )

**Return Type**

Character, Numeric

**Remarks**

The FxVolume( ) function returns the list of volumes (hard drives, floppy disks, network connected volumes) delimited with semicolons.

If an error occurs, the function returns an operating system error as a numeric value. Use the Type( ) function to determine the data type of the return value before attempting to manipulate the value.

# GetClipDat( )

Available in Visual FoxPro for Windows only.

Retrieves a handle for the Clipboard data of a specified format and passes it directly to the calling application.

**Syntax**

GetClipDat(*nFormat*)
*nFormat*

Contains an identifier for possible Clipboard formats:

| nFormat | Description (define type) |
|---|---|
| 1 | cf_Text |
| 2 | cf_Bitmap |
| 3 | cf_MetaFilePict |
| 4 | cf_SYLK |
| 5 | cf_DIF |
| 6 | cf_TIFF |
| 7 | cf_OEMText |
| 8 | cf_DIB |
| 9 | cf_Palette |

**Return Type**

Logical

**Remarks**

The clipboard controls the handle, not the application. The application (Visual FoxPro itself does) should copy the data immediately. Almost all Clip functions rely on opening the Clipboard before using the function. You can use _CLIPTEXT to access the contents of the Windows Clipboard.

**Note** This function maps to the Windows SDK function that is similarly named. For more information, see the Windows SDK documentation.

# GetClipFmt( )

Available in Visual FoxPro for Windows only.

Retrieves the name of a registered Clipboard format.

**Syntax**

GetClipFmt(*nFormat*)
*nFormat*

Specifies the registered format to retrieve. This argument must not specify any of the predefined clipboard formats.

**Return Type**

Numeric

**Remarks**

Almost all Clip functions rely on opening the Clipboard before using the function. You can use _CLIPTEXT to access the contents of the Windows Clipboard.

**Note** This function maps to the Windows SDK functions that is similarly named. For more information, see the Windows SDK documentation.

# GetFileVersion( )

Returns information about a file.

**Syntax**

GetFileVersion(*cFileName, @ArrayName*)
*cFileName*

Specifies the name of the file for which information is returned.

*ArrayName*

Specifies the name of the array in which the file information is placed. The array must be created before issuing GetFileVersion( ), and must have at least 1 column and 12 rows.

**Return Type**

Numeric

**Remarks**

GetFileVersion( ) is typically used to get information about .EXE and .DLL files.

Zero is returned if the function is successful; otherwise -1 is returned.

The following table lists the file information contained in the twelve array elements:

| Element Number | File Information |
|---|---|
| 1 | Comments |
| 2 | Company Name |
| 3 | File Description |
| 4 | File Version |
| 5 | Internal Name |
| 6 | Legal Copyright |
| 7 | Legal Trademarks |
| 8 | Original File Name |
| 9 | Private Build |
| 10 | Product Name |
| 11 | Product Version |
| 12 | Special Build |

# GetProStrg( )

Available in Visual FoxPro for Windows only.

Retrieves the string associated with an entry within the specified section in the WIN.INI initialization file.

**Syntax**

GetProStrg (lpszSection, lpszEntry, lpszDefault,
    *@lpszReturnBuffer*, *cbReturnBuffer*)
*lpszSection*

Points to a null-terminated string that specifies the section containing the entry.

*LpszEntry*

Pointer to the null-terminated string containing the entry whose associated string is to be retrieved. If this value is NULL, all entries in the section specified by the *lpszSection* argument are copied to the buffer specified by the *lpszReturnBuffer* argument.

*LpszDefault*

Pointer to the default value for the given entry if the entry cannot be found in the initialization file. This argument must never be NULL; it must point to a valid string, even if the string is empty (its first character is zero).

*lpszReturnBuffer*

Pointer to the buffer that will receive the character string.

*cbReturnBuffer*

Specifies the size, in bytes, of the buffer pointed to by the *lpszReturnBuffer* argument.

**Return Value**

Numeric

**Remarks**

An application can use the GetProStrg( ) function to retrieve a string from a specified file.

If the *lpszEntry* argument is NULL, the GetProStrg( ) function copies all entries in the specified section to the supplied buffer. Each string will be null-terminated, with the final string terminating with two null characters. If the destination buffer is too small to hold all the strings, the last string will be truncated and followed by two terminating null characters.

If the string associated with *lpszEntry* is enclosed in single or double quotation marks, the marks are discarded when GetProStrg returns the string.

GetProStrg( ) is not case dependent, so the strings in the *lpszSection* and *lpszEntry* arguments may contain a combination of uppercase and lowercase letters.

The return value is the number of bytes copied to the buffer, not including the terminating zero, if the function is successful.

# IsClipFmt( )

Available in Visual FoxPro for Windows only.

Indicates whether data of the specified format is currently on the Clipboard.

**Syntax**

IsClipFmt(*nFormat*)
*nFormat*

Contains a numeric value indicating the format of the available data.

The following table represents the predefined Windows formats:

| nFormat | Description (define type) |
|---|---|
| 1 | cf_Text |
| 2 | cf_Bitmap |
| 3 | cf_MetaFilePict |
| 4 | cf_SYLK |
| 5 | cf_DIF |
| 6 | cf_TIFF |
| 7 | cf_OEMText |
| 8 | cf_DIB |
| 9 | cf_Palette |

**Return Type**

Logical

**Remarks**

IsClipFmt( ) returns true (.T.) if data of the specified format is on the Clipboard, or false (.F.) otherwise.

**Note** This function maps to the Windows SDK functions that is similarly named. For more information, see the Windows SDK documentation.

# JustDrive( )

Returns the drive letter from a complete path.

**Syntax**

JustDrive(*cPath*)
*cPath*

Specifies the complete path name for which you want only the drive.

**Return Type**

Character

# JustExt( )

Returns the three-letter extension from a complete path.

**Syntax**

JustExt(*cPath*)
*cPath*

Specifies the name, which may include the full path, of the file for which you want only the extension.

**Return Type**

Character

# JustFName( )

Returns the file name portion of a complete path and file name.

**Syntax**

JustFName(*cFilename*)
*cFilename*

Specifies the name, which may include the full path, of the file for which you want only the file name.

**Return Type**

Character

# JustPath( )

Returns the path portion of a complete path and file name.

**Syntax**

JustPath(*cFilename)*
*cFilename*

Specifies the full name (including path) of the file for which you want only the path.

**Return Type**

Character

# JustStem( )

Returns the stem name (first eight characters of file name) from a complete path and file name.

**Syntax**

JustStem(*cFilename*)
*cFilename*

Specifies the name (including path) of the file for which you want only the stem.

**Return Type**

Character

# MainHwnd( )

Available in Visual FoxPro for Windows only.

Returns the window handle (HWND) of the main Visual FoxPro window.

**Syntax**

MainHwnd( )

**Return Type**

Numeric

# MkDir( )

Creates a directory or folder.

**Syntax**

MkDir(*cPath*)
*cPath*

Specifies the path to create.

**Return Type**

Numeric

**Remarks**

This function will not check for the valid length and format of a directory string. The return values are:

| | |
|---|---|
| 0 | success |
| 1 | no success |
| 6 | directory already exists |

This function is only available in FoxTools version 1.01 or later.

# MsgBox( )

Available in Visual FoxPro for Windows only. In Visual FoxPro for Macintosh, use the FxAlert( )

function instead.

Displays a modal dialog box centered in the screen.

**Syntax**

MsgBox(*cText*, *cTitle*, *nType*)
*cText*

Specifies the contents in dialog box.

*ctitle*

Specifies the title of dialog box window.

*ntype*

Specifies the type of dialog box as follows:

| nType | Type |
|---|---|
| 0 | MB_OK |
| 1 | MB_OKCANCEL |
| 2 | MB_ABORTRETRYIGNORE |
| 3 | MB_YESNOCANCEL |
| 4 | MB_YESNO |
| 5 | MB_RETRYCANCEL |
| 16 | MB_ICONSTOP |
| 32 | MB_ICONQUESTION |
| 48 | MB_ICONEXCLAMATION |
| 64 | MB_ICONINFORMATION |

**Return Type**

Numeric

**Remarks**

MsgBox( ) returns a value indicating the button that the user clicked:

| Return value | ID of button clicked |
|---|---|
| 1 | idok |
| 2 | idcancel |
| 3 | idabort |
| 4 | idretry |
| 5 | idignore |
| 6 | idyes |
| 7 | idno |

# NextWord( )

Returns the characters between a specified character index and the following word delimiter (or the end of the string).

**Syntax**

NextWord(*cString*, *nPosition*[, *cDelimiter*])
*cString*

Specifies the string to search.

*nposition*

Specifies the numeric position of the character in the string that will be the initial character returned by NextWord( ).

*cdelimiter*

Specifies the optional word delimiters; NextWord( ) returns the string of characters between *nPostition* and this character. If *cDelimiter2* is not used, the default delimiters are spaces, tabs, and carriage return characters (ASCII 13). The delimiter character is not included in the returned string.

**Return Type**

Character

# OpenClip( )

Available in Visual FoxPro for Windows only.

Opens the Clipboard for access by subsequent Clipboard-related functions.

**Syntax**

OpenClip(*nHandle*)
*nHandle*

Specifies the handle of the window to be associated with the Clipboard. 0 is acceptable. To get the handle of the Visual FoxPro main window, call MainHwnd( ).

**Return Type**

Logical

**Remarks**

The return value reports success (.T.) or failure (.F.) of the command. Almost all Clip functions rely on opening the Clipboard before using the function. You can use _CLIPTEXT to access the contents of the Windows Clipboard.

**Note** This function maps to the Windows SDK functions that are similarly named. For more information, see the Windows SDK documentation.

# PutProStrg( )

Available in Visual FoxPro for Windows only.

Copies a string into the specified section of the Windows initialization file (Win.ini).

**Syntax**

PutProsStrg(*lpszSection*, *lpszEntry*, *lpszString*)
*lpszSection*

Pointer to a null-terminated string that specifies the section to which the string is to be copied. If the section does not exist, it is created. The name of the section is case-independent; the string may be any combination of uppercase and lowercase letters.

*LpszEntry*

Pointer to the null-terminated string containing the entry to be associated with the string. If the entry does not exist in the specified section, it is created. If this argument is NULL, the entire section, including all entries within the section, is deleted.

*LpszString*

Pointer to the null-terminated string to be written to the file. If this argument is NULL, the entry specified by the *lpszEntry* argument is deleted.

**Return Type**

Numeric

**Remarks**

The return value is nonzero if the function is successful; otherwise, it is zero.

# Pict2File( )

Available in Visual FoxPro for Macintosh only.

Creates a file out of a picture resource.

**Syntax**

Pict2File(*cFileName*, *cResource*, *nResourceNumber*, *cNewFileName*)
*cFileName*

Specifies the path and file name of the file containing the resource.

*cResource*

Specifies the type of resource to move: "PICT", "ICON" or "ICN#".

*nResourceNumber*

Specifies the number of the resource to move.

*cNewFileName*

Specifies the path and file name where the new image will be saved.

**Return Type**

Numeric

**Remarks**

The Pict2File( ) function takes a PICT, ICON, or ICN# resource and creates a separate file containing the image. The new file is a PICT type file. Pict2File( ) returns:

| | |
|---|---|
| 0 | no error |
| -2 | not enough memory to complete the operation |
| -3 | resource not found |
| other | operating system error |

# Reduce( )

Replaces specified characters in a string with a space.

**Syntax**

Reduce(*cSearch*, *cReplace*)
*cSearch*

Specifies the character string to change.

*cReplace*

Specifies the characters to search for and replace with a space. If you specify more than one character, the characters are treated separately during the reduction; they are not treated as a single multi-character string.

**Return Type**

Character

**Remarks**

Replaces characters specified in *cReplace* with a space, then removes leading spaces and repeated spaces within the string. The function is commonly used to replace a group of spaces in a string with a single space, or to replace delimiters such as tabs or carriage returns with a space.

# RegClipFmt( )

Available in Visual FoxPro for Windows only.

Registers a new Clipboard format.

**Syntax**

RegClipFmt(*cFormat*)
*cFormat*

Specifies a string that names the new format.

**Return Type**

Numeric

**Remarks**

The registered format can be used in subsequent clipboard functions as a valid format in which to render data, and it will appear in the Clipboard's list of formats.

The return value indicates the newly registered format. If the identical format name has been registered before, even by a different application, the format's reference count is incremented (increased by one) and the same value is returned as when the format was originally registered. The return value is zero if the format cannot be registered.

Almost all Clip functions rely on opening the Clipboard before using the function. You can use _CLIPTEXT to access the contents of the Windows Clipboard.

**Note** This function maps to the Windows SDK functions that is similarly named. For more information, see the Windows SDK documentation.

# RGBComp( )

Returns the red, green, and blue components of a composite RGB color value.

**Syntax**

RGBComp( *nRGBColor*, @*nRedVal*, @*nGreenVal*, @*nBlueVal* )
*nRGBColo*r

Specifies a composite RGB color value ranging from 0 to 16777215.

*@nRedVal*

Specifies a reference to a variable in which the function returns the intensity of the red color component of *nRGBColor*.

@nGreenVal

Specifies a reference to a variable in which the function returns the intensity of the green color component of *nRGBColor*.

*@nBlueVal*

Specifies a reference to a variable in which the function returns the intensity of the blue color component of *nRGBColor*.

**Return Type**

Logical

# RmDir( )

Deletes a directory of folder.

**Syntax**

RmDir(*cPath*)
*cPath*

Specifies the path of the directory or folder to be removed.

**Return Type**

Numeric

**Remarks**

This function is only available in FoxTools version 1.01 or later.

Returns 0 if successful and 1 if not successful.

# SetClipDat( )

Available in Visual FoxPro for Windows only.

Sets the data on the opened Clipboard.

**Syntax**

SetClipDat(*nFormat*, *cData*)
*nFormat*

Specifies an identifier for possible Clipboard formats:

| nFormat | Description (define type) |
|---|---|
| 1 | cf_Text |
| 2 | cf_Bitmap |
| 3 | cf_MetaFilePict |
| 4 | cf_SYLK |
| 5 | cf_DIF |
| 6 | cf_TIFF |
| 7 | cf_OEMText |
| 8 | cf_DIB |
| 9 | cf_Palette |

*cData*

Specifies the data to place on the Clipboard.

**Return Type**

Logical

**Remarks**

The Clipboard must have been opened with OpenClip( ) before you call this function. If the Windows Clipboard is running, it will not update its window to show the data placed in the clipboard by the SetClipDat( ) function until after the CloseClip( ) function is called.

**Note** This function maps to the Windows SDK functions that is similarly named. For more information, see the Windows SDK documentation.

# StrFilter( )

Removes all characters from a string except those specified.

**Syntax**

StrFilter(*cString*, *cSearch*)
*cString*

Specifies the character string to search.

*cSearch*

Specifies the characters to search for and retain in *cString*.

**Return Type**

Character

**Remarks**

StrFilter( ) removes all the characters from *cString* that are not in *cSearch*, then returns the characters that remain. StrFilter( ) is case-sensitive.

# ValidPath( )

Checks for a valid MS-DOS file name or path expression.

**Syntax**

ValidPath(*cName*)
*cName*

Specifies the path or file name to be checked.

**Return Type**

Logical

**Remarks**

ValidPath( ) verifies that a file name or path name is syntactically legal; it does not check for the existence of the specified file or path. The function is not foolproof and can sometimes interpret a file as having a valid name when the file name is actually invalid. However, ValidPath( ) will not reject a valid name.

Not supported in file systems that support spaces in path names.

# WordNum( )

Returns the specified word in a string.

### Syntax

WordNum(*cString*, *nInde*x[, *cDelimiter*])
*cString*

Specifies the string containing the word to be returned.

*nIndex*

Specifies the index position of the word to be returned. For example, if *nIndex* is 3, WordNum( ) returns the third word (if *cString* contains three or more words).

*cDelimiter*

Specifies the character used to delimit words in *cString*. The default delimiters are space, tab, and carriage return.

### Return Type

Character

### Remarks

If *cString* contains fewer words than the number specified with *cIndex*, WordNum( ) returns null.

# Words( )

Counts the number of words in a string.

### Syntax

Words(*cString*[, *cDelimiter*])
*cString*

Specifies the string of words to be counted.

*cdelimiter*

Specifies the character used to delimit words in *cString*. The default delimiters are space and tab.

### Return Type

Numeric