Foxtools Overview

Foxtools is a Visual FoxPro API library that exposes Windows DLLs for use in Visual FoxPro.

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 aren't 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.

What's New

Several file name manipulation functions in Foxtools.fll have been added to Visual FoxPro. It's no longer necessary to use SET LIBRARY TO FOXTOOLS.FLL to call these functions. As a native function in Visual FoxPro, you can call them directly in your Visual FoxPro programs. They are the following:

AddBS( ) Function AGetFileVersion( ) Function (GetFileVersion( ) in FoxTools)
DefaultExt( ) Function
DriveType( ) Function
ForceExt( ) Function
ForcePath( ) Function
JustDrive( ) Function
JustExt( ) Function
JustFName( ) Function
JustPath( ) Function
JustStem( ) Function

Where Foxtools Gets Installed

When you install Visual FoxPro, Foxtools.fll is installed in the main Visual FoxPro directory. 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.

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 (SDK). 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.

AddBS( )

Adds a backslash (if needed) to a path expression. Also available as a native function in Visual FoxPro.

Syntax

AddBS(*cPath*)

cPath

Specifies the path name to which to add the backslash.

Return Type

Character

CallFn( )

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( ).

CleanPath( )

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

Syntax

CleanPath(*cFilename*)

cFilename

Specifies the file name to be cleaned.

Return Types

Character

CloseClip( )

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 function that is similarly named. For more information, see the Windows SDK documentation.

CountClipF( )

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 function 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. Also available as a native function in Visual FoxPro.

Syntax

DefaultExt(*cFilename, cDefault*)

cFilename

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

Cdefault

Specifies the default extension without a period.

Return Type

Character

DriveType( )

Returns the type of the specified drive. Also available as a native function in Visual FoxPro.

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( )

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 function that is similarly named. For more information, see the Windows SDK documentation.

EnumClipFm( )

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 function 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. Also available as a native function in Visual FoxPro.

Syntax

ForceExt(*cFilename*, *cExtension*)

cFilename

Specifies the file name (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. Also available as a native function in Visual FoxPro.

Syntax

ForceExt(*cFilename*, *cPath*)

cFilename

Specifies the file name (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

GetClipDat( )

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 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( )

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. Also available as the native function AGETFILEVERSION( ) in Visual FoxPro.

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( )

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 can't 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( ) isn't 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( )

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. Also available as a native function in Visual FoxPro.

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. Also available as a native function in Visual FoxPro.

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. Also available as a native function in Visual FoxPro.

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. Also available as a native function in Visual FoxPro.

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. Also available as a native function in Visual FoxPro.

Syntax

JustStem(*cFilename*)

cFilename

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

Return Type

Character

MainHwnd( )

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

Syntax

MainHwnd( )

Return Type

Numeric

MkDir( )

Creates a directory.

Syntax

MkDir(*cPath*)

cPath

Specifies the path to create.

Return Type

Numeric

Remarks

This function won't check for the valid length and format of a directory string. The return values are as follows:

0	success
1	no success
6	directory already exists

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

MessageBox( )

Displays a modal dialog box centered in the screen.

Syntax

MessageBox(*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

MessageBox( ) 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* isn't used, the default delimiters are spaces, tabs, and carriage return characters (ASCII 13). The delimiter character isn't included in the returned string.

Return Type

Character

OpenClip( )

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. Zero 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 function that is similarly named. For more information, see the Windows SDK documentation.

PutProStrg( )

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 doesn't 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 doesn't 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.

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 aren't treated as a single multicharacter 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( )

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 can't 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 function 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.

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( )

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 won't 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 function 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 aren't 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 doesn't check for the existence of the specified file or path. The function isn't foolproof and can sometimes interpret a file as having a valid name when the file name is actually invalid. However, ValidPath( ) won't 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 an empty string.

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