Function Reference

FileWriteLine

Append a line of text to the end of a previously opened text file.

```
FileWriteLine ( "filehandle/filename", "line" )
```

Parameters

filehandle/filename	The handle of a file, as returned by a previous call to FileOpen(). Alternatively, you may use a strinc filename as the first parameter.
line	The line of text to write to the text file. If the line does NOT end in @CR or @LF then a DOS linefeed (@CRLF) will be automatically added.

Return Value

Success:	1.
Failure:	0 if file not opened in writemode, file is read only, or file cannot otherwise be written to.

Remarks

The text file must be opened in write mode or the FileWriteLine() command will fail.

If a filename is given rather than a file handle, the file will be opened and closed during the function call. For parsing large text files this will be much slower than using filehandles. However, filename will be created if it does not already exist.

Note: Do not mix filehandles and filenames, i.e., don't FileOpen() a file and then use a filename in this function. Either use filehandles or filenames in your routines, not both.

When writing text AutoIt will write using UTF8 (without BOM) by default. To write in another mode the file must be opened with FileOpen() and the relevant flags.

The text to write cannot contain Chr(0) characters as the output is truncated. FileWrite() using a file opened in binary mode must be used to write such characters.

Related

FileFlush, FileGetPos, FileOpen, FileRead, FileReadLine, FileSetPos, FileWrite

Example

Copy to clipboard

```
#include <FileConstants.au3>
#include <MsgBoxConstants.au3>
#include <WinAPIFiles.au3>

Example()

Func Example()
    ; Create a constant variable in Local scope of the filepath that will be read
    Local Const $sFilePath = _WinAPI_GetTempFileName(@TempDir)

    ; Create a temporary file to write data to.
    If Not FileWrite($sFilePath, "Start of the FileWriteLine example, line 1. " &
        MsgBox($MB_SYSTEMMODAL, "", "An error occurred whilst writing the tempora
        Return False
    EndIf

    ; Open the file for writing (append to the end of a file) and store the handl
    Local $hFileOpen = FileOpen($sFilePath, $FO_APPEND)
    If $hFileOpen = -1 Then
        MsgBox($MB_SYSTEMMODAL, "", "An error occurred whilst writing the tempora
        Return False
    EndIf

    ; Write data to the file using the handle returned by FileOpen.
    FileWriteLine($hFileOpen, "Line 2")
    FileWriteLine($hFileOpen, "This is line 3 as a new line was appended to the l
    FileWriteLine($hFileOpen, "Line 4" & @CRLF)
    FileWriteLine($hFileOpen, "Line 5")

    ; Close the handle returned by FileOpen.
    FileClose($hFileOpen)

    ; Display the contents of the file passing the filepath to FileRead instead o
    MsgBox($MB_SYSTEMMODAL, "", "Contents of the file:" & @CRLF & FileRead($sFile

    ; Delete the temporary file.
    FileDelete($sFilePath)
EndFunc   ;==>Example
```