

# Offline Manuals

## Scripts Encryptor (ScrEnc)



### Version: 3.0.3.2

To download your copy [click here](#)

(You can register the Scripts Encryptor from its Help -> Register menu)

### Command Line Operation

Scripts Encryptor can be run from the Windows command line without involving the GUI part of the program. You can also use command line parameters to perform batch conversion of the whole folder(s).

To evoke Scripts Encryptor from command line run it using the following syntax:

SCRENC [Parameters]

The following legacy parameters are supported:

[/s] [/f] [/xl] [/l defLanguage] [/e defExtension] inputfile outputfile

NOTE: Default values from the description below will be used if none specified.

/?	Show help window.
/s	Do silent conversion, or don't show status window (Default is not to be silent). INFO: Keep in mind that conversion of multiple files may take a longer time.
/f	Force overwrite input file(s). If <outputfile> is not used, <inputfile> will be used as output. Default is not to overwrite files. WARNING: This option will make Scripts Encryptor to overwrite the source file(s) specified by the <inputfile> parameter without a prompt! This option cannot be used along with the /isf parameter for safety reasons.
/xl	The @language directive will not be added in ASP files if Windows Script Encoder method was used. (Default: this option is not used.)
/l ScriptLanguage or /l=ScriptLanguage	Default language to be used if none were specified in an HTML/ASP file. (Example: <ScriptLanguage> could be "VBScript", "JScript".) Default: None
/e DefaultExtension or /e=DefaultExtension	Associate input file(s) with a specific file type. <DefaultExtension> could be: asa, asp, htm, js, vbs Default: None Do not specify this parameter if you want the script type to be determined automatically by the input file(s) extensions. (This parameter overrides the /tp parameter.)
<inputfile>	The file(s) to convert. This parameter is mandatory and must specify files only. It can contain a relative, absolute or a network path. If file name/path contains spaces, enclose it into double-quotation marks.

	<p>(Example: "My file.htm")</p> <p>To specify more than one file, separate them with the   symbol. (Example: "file1.js file2.htm")</p> <p>It doesn't matter in what order you place this parameter, except that it has to be the first non-flagged one. Make sure that this parameter doesn't follow any of the legacy flags: /l or /e.</p> <p>The file name(s) may contain the following wildcards:</p> <ul style="list-style-type: none"> <li>* - for any match (Example: "*.js" will match all files with the JS extension)</li> <li>? - for a single character match (Example: "file.htm?" will match files such as "file.htm" and "file.html")</li> </ul>
<outputfile>	<p>The destination location.</p> <p>This parameter is mandatory if /f is not specified.</p> <p>It can contain a relative, absolute or a network path.</p> <p>When &lt;inputfile&gt; specifies more than a single file (by using the   symbol, or by containing wildcards) &lt;outputfile&gt; is interpreted as a folder, and all converted files will be placed into it and will retain their names.</p> <p>If this name/path contains spaces, enclose it into double-quotation marks. (Example: "My folder")</p> <p>If it specifies a folder and it doesn't exist the /cf parameter must be used to create it automatically.</p> <p>If &lt;outputfile&gt; refers to the same location as &lt;inputfile&gt; the /f parameter must be used. In this case the /isf parameter cannot be used for safety reasons.</p> <p>It doesn't matter in what order you place this parameter, except that it has to be the second non-flagged one. Make sure that this parameter doesn't follow any of the legacy flags: /l or /e.</p>

The following advanced parameters are supported:

```
[/op=[s|o|n|rc]] [/tp=[h|a|j|c|m|v]] [/lf=[c|l|r]] [/htm=[g|t|m[o|c|s|u|x|a]]]]]]]
[/sct=[e|q|c|m[a|r]]]]]]] [/isf] [/cf]
[/xtr=[a]] [/ret=N] [/awse=[all|[ScriptType1[,ScriptType2][,...]]]] [/fe=[ANSI|utf-8|utf-16be|utf-16|Unicode]] [/cp="Name"]
[/rpt=["Path"]] [/uid=ID]
```

NOTE: Default values from the description below will be used if none specified.

/op	<p>Operation to perform.</p> <p>Accepted one of the following values: s=scramble, o=Organize; n=No change; rc=Remove C++/JScript/VBScript comments only.</p> <p>Default: 'n'</p> <p>(INFO: 'o' option is available in the Premium Version only.)</p>
/tp	<p>Type of the &lt;inputfile&gt;.</p> <p>Accepted one of the following values: h=HTML; a=ASP; j=JavaScript; c=C/C++; m=MFC/C++; v=VBScript</p> <p>Default: None</p> <p>Do not specify this parameter if you want the script type to be determined automatically by the input file(s) extensions.</p> <p>(INFO: This parameter can be overridden by the /e parameter.)</p>
/lf	<p>Type of end-lines to use for saving.</p> <p>Accepted one of the following values: c=Carriage Return &amp; Line Feed; l=Line Feed; r=Carriage Return</p> <p>(Default: "l" for scrambling, and "c" for other operations.)</p>
/htm	Individual flags for the HTML/ASP files:

There are several types of equation signs that you can use:

= overwrites all default values with the flag(s) specified (Example: `/htm=gm`, will set only 'g' and 'm' flags)

+= adds specified flag(s) to the default ones (Example: `/htm+=g`, will set 'g' flag besides the default ones)

-= removes specified flag(s) from the default ones (Example: `/htm-=m`, will remove the 'm' flag from the default ones)

^= toggles specified default flag(s) (Example: `/htm^=gm`, will set 'g' flag and remove the 'm' flag)

g - Escape tags (scrambling only) (Default: Off)

t - Escape text (scrambling only) (Default: Off)

m - Random escaping (scrambling only) (Default: On)

o - Save for viewing offline (in Internet Explorer only) (Default: Off)

c - Keep comment tags when scrambling (Example: `<!-- ... -->`) (Default: On)

s - Use comment tags on script blocks, if allowed (Example: `<script><!-- ... --></script>`) (Default: On)

u - Allow scrambling of URLs (scrambling only) (Default: On)

x - Allow using short hex escape sequences (i.e. `%xx`) on URLs (scrambling only) (Default: On)

a - Use all-inclusive conversion (Default: On)

(This option will update all HTML/ASP script's `<script src="">` tags that may refer to external script files that will be converted in this operation as well.)

/sct

Individual flags for the JavaScript/JScript/VBScript files or script blocks within HTML/ASP files:

There are several types of equation signs that you can use:

= overwrites all default values with the flag(s) specified (Example: `/sct=eq`, will set only 'e' and 'q' flags)

+= adds specified flag(s) to the default ones (Example: `/sct+=q`, will set 'q' flag besides the default ones)

-= removes specified flag(s) from the default ones (Example: `/sct-=e`, will remove the 'e' flag from the default ones)

^= toggles specified default flag(s) (Example: `/sct^=eq`, will remove 'e' flag and set the 'q' flag)

e - Encode script(s) using Windows Script Encoder method (Default: On)

q - Escape quotes (for JavaScript/JScript and scrambling only) (Default: Off)

c - Escape code (for JavaScript/JScript and scrambling only) (Default: Off)

m - Random escaping (for JavaScript/JScript), or random capping (for VBScript) (scrambling only) (Default: On)

a - Add missing semicolons (for JavaScript/JScript), or colons (for VBScript) (Default: Off)

r - Remove extra semicolons (for JavaScript/JScript only) (Default: Off)

u - Update script(s) file name(s) (Default: Off)

	(In case a script is in a stand-alone JS or VB file and 'e' flag was specified as well)
/isf	<p>If wildcards were used in the &lt;inputfile&gt;, to include all subfolders in the search as well. (Default: Off)</p> <p>(This parameter cannot be used along with the /f parameter.)</p>
/cf	<p>If &lt;outputfile&gt; specifies a folder, this option creates it first before writing converted files into it. (Default: Off)</p> <p>(The folder is created with all the subfolders specified in the &lt;outputfile&gt; path.)</p>
/xtr	<p>Additional/extra flags:</p> <p>There are several types of equation signs that you can use:</p> <ul style="list-style-type: none"> <li>= overwrites all default values with the flag(s) specified</li> <li>+= adds specified flag(s) to the default ones</li> <li>-= removes specified flag(s) from the default ones</li> <li>^= toggles specified default flag(s)</li> </ul> <p>a - Always escape non-ASCII chars (for JavaScript/JScript only) (Default: On)</p>
/ret	<p>Value for the random escaping/capping threshold. Allowed values are from 1 to 15, inclusive. [1=Less, ... 15=More]</p> <p>Default: 4</p>
/awse	<p>Allow applying Windows Script Encoder method in HTML/ASP files to:</p> <p>all - All script blocks (Default)</p> <p>&lt;ScriptTypes&gt; - Specific script blocks, presented in a comma-separated list of script names. If these names contain spaces, enclose them into double-quotation marks. (Example: "JavaScript1.3, VBScript")</p>
/fe	<p>File encoding to be used for saving &lt;outputfile&gt;:</p> <p>&lt;None&gt; - Use the file encoding from the &lt;inputfile&gt; (Default)</p> <p>ANSI - Single-byte ANSI</p> <p>utf-8 - Various length UTF-8</p> <p>utf-16be - 16-bit Unicode (big-endian)</p> <p>utf-16 - or</p> <p>Unicode - 16-bit Unicode (little-endian)</p>
/cp	<p>Abbreviated code page name to be used for conversion while saving &lt;outputfile&gt;:</p> <p>&lt;None&gt; - Use code page specified in &lt;inputfile&gt; (for HTML), or locally installed one (Default)</p> <p>&lt;Name&gt; - Any of the short code page names listed below.</p>
/rpt	<p>Generate conversion report: (Default: Off)</p> <p>&lt;Path&gt; - Path to a file to save report in. It can contain a relative, absolute or a network path. If file name/path contains spaces, enclose it into double-quotation marks. (Example: "My report.txt") The data is saved as an ANSI text file in a format described below.</p> <p>&lt;None&gt; - If /rpt is used without a file path/name, the report is output to the screen. (Conversion with the output to the screen may be somewhat slower.)</p>
/uid	<p>User ID to be added to the report file (if /rpt parameter was used) (Default: "")</p> <p>If ID value contains spaces, enclose it into double-quotation marks. (Example: "ID #123")</p>

## Additional Info

NOTE 1: Spaces serve as separators between parameters in the command line.

In case you need to include spaces in a parameter's value, enclose it into double-quotation marks:

Example: `/uid="File 1"`

NOTE 2: In case you need to include special characters in a parameter's value, escape them with the backslash symbol:

Example: `/uid="\a\\"b"`, will set UID to `"a\b"`

IMPORTANT: In case of specifying a folder, make sure not to include the last slash:

Example: `"C:\My folder\"` will treat the last slash as an escape sequence for the double-quote.

Use `"C:\My folder"`, or `"C:\My folder\\"` instead.

NOTE 3: It is not recommended to use command lines longer than 256 symbols (including the path for the "screnc" itself).

To specify a longer command line save it in a text file and use the following syntax to run it:

(When used with the `:::` or `*:::` syntax, no other command line parameters should be used when calling "screnc".)

```
screnc :: "Path"
```

- Will read the command line from a file specified by "Path" (Example: `screnc :: "external commands.txt"`)

(The format of the external command line is the same as described here, except that line breaks are interpreted as spaces.)

```
screnc *:: "Path"
```

- Will read the command line from a file specified by "Path" and delete that file when done (Example: `screnc *:: "temp external commands.txt"`)

(The format of the external command line is the same as described here, except that line breaks are interpreted as spaces.)

## Report File Layout

NOTE: This report is generated only if `/rpt` parameter was used.

If this data is saved in a file, the ANSI format is used.

Each value is presented on a separate line.

Empty lines, or lines starting with the equation sign '=' should not be interpreted.

ID: "[Value]";	ID Value specified in the /uid parameter.
TIMESTAMP: <Time>	Time stamp when the conversion began. Format: YYYYMMDDHHmmSS, where YYYY=4-digit year, MM=2-digit month, DD=2-digit day, HH=2-digit hour (in 24-hour format), mm=2-digit minutes, SS=2-digit seconds.
CMD_LINE: <CmdLine>	Full set of command line parameters used for the conversion.
VER: "<Version>"	Version of the Scripts Encryptor this report was generated with.
Collecting Files:	
FOUND: <Number>	Total number of files found suitable for conversion. (Note that the number of files converted can be less than this number.)
Starting Conversion:	Delimits the beginning of the conversion process.
SRC: "<source>"	Source file used for conversion.
DEST: "<destination>"	Location where the destination file was placed after conversion.

STAT: 0x<Code>	<p>Status result code generated after conversion: The code value is presented as a hexadecimal number, preceded by '0x':</p> <p>0 - Success</p> <p>2 - Did not encode script as it was already encoded</p> <p>16 bit - Set if syntax error while parsing the script (value of 0x8000). This value may be combined with positive status codes.</p> <p>15 bit - Set if some Unicode characters were lost during conversion (value of 0x4000). This value may be combined with positive status codes.</p> <p>-1 - Memory fault while reading the source file</p> <p>-2 - File I/O error while reading the source file</p> <p>-3 - General/internal error reading the source file</p> <p>-4 - Memory fault while converting the source file</p> <p>-5 - User aborted the operation</p> <p>-6 - General/internal error converting the source file</p> <p>-7 - This option cannot be applied to this file, operation skipped</p> <p>-8 - Script type cannot be determined for the file</p> <p>-9 - Internal error encoding script</p> <p>-A - The source and destination cannot refer to the same file when /isf is used</p> <p>-B - File I/O error while writing to the destination file</p> <p>-C - Memory fault while writing to the destination file</p> <p>-D - Unsupported code page</p> <p>-E - Unsupported file encoding</p> <p>-F - Requested code page is not installed for conversion</p> <p>-10 - Expired demo version. Please register to continue</p>
TYPE: <Type>	<p>Type of the script used for conversion. Could be: HTML, ASP, JavaScript, C, C++, VBScript, or &lt;empty&gt;</p>
OP: <Type>	<p>Type of the operation performed during conversion. Could be: "Scramble", "Organize", "Remove Comments", "No Change"</p>
FECP SRC: "<Value>"	<p>File encoding and code page of the source file before conversion. Format: "&lt;FileEncoding&gt;/[&lt;ShortCodePageName&gt;] &lt;LongCodePageName&gt;", if code page is available, or "&lt;FileEncoding&gt;", if no code page.</p>
FECP DEST: "<Value>"	<p>File encoding and code page used for saving converted file. See 'FECP SRC:' above for format details.</p>
ENC: <Value>	<p>Specifies whether Windows Script Encoder method was used on this file, or in case of HTML/ASP file, if it was used on any of the &lt;script&gt; blocks in them.</p>

	Could be: Yes or No.
SC_ADD: <Number>	Number of semicolons (or colons) added to the script. (For HTML/ASP file this is a total number for all script blocks.)
SC_RMD: <Number>	Number of semicolons removed from the script. (For HTML/ASP file this is a total number for all script blocks.)
RESULT MSG:	
<Message>	Human readable message signifying the result of conversion.  If present, the <Message> will be placed on the next line after the 'RESULT MSG:' tag. The <Message> can be empty if conversion completed successfully. The <Message> may be preceded by 'WARNING:', 'ERROR:', etc. It normally has a direct correlation to the code reported in the 'STAT:' tag.
	After the last converted file:
Operation(s) Results:	Delimits the end of the conversion process.
CONV_OK: <Number>	Number of files converted successfully.
ERRORS: <Number>	Number of errors that occurred during conversion. (Errors normally mean that conversion was discontinued for that number of files.)
WARNINGS: <Number>	Number of warnings that were issued during conversion. (Warnings normally mean that converted data was saved, but with some unforeseen adjustments.)
SYNTAX ERRS: <Number>	Number of files that contained syntax errors in the JavaScript/JScript blocks/files. (This may mean that conversion produced some erroneous results due to encoder's inability to interpret script(s) correctly.)
RESULT CODE: 0x<Code>	Result code for the overall conversion process:  This value should be evaluated along with status result codes for each converted file if appropriate bits are set for the positive result code (see below).  This value will be returned from the screnc.exe module to a calling process.  The code value is presented as a hexadecimal number, preceded by '0x': (In case of a positive value, only lower 16 bits are used for the code)  0 - Success  1 - No suitable files found for conversion  Bits 17 through 20 may be set individually in case of the following:  17bit - (0x10000) set if at least one error occurred while converting file(s)  18bit - (0x20000) set if at least one warning occurred while converting file(s)  19bit - (0x40000) set if at least one syntax error occurred while converting file(s)  (In case of a negative value, the whole value is used for the code)  -1 - No input file was specified for conversion  -2 - No output file was specified for conversion  -3 - Invalid command line parameter(s)

	-4 - Cannot enumerate requested input file(s)
	-5 - Operation aborted by user
	-6 - Internal error
	-7 - The <source> and <destination> cannot refer to the same folder when /isf parameter is used
	-8 - Could not open the external command line file (error message will be shown to the user)
	-9 - Could not parse the external command line file (error message will be shown to the user)
	-10 - Could not save the report in a specified file

## Supported File Encodings

The following file encodings are recognized and supported by Scripts Encryptor:

ANSI	American National Standard Institute encoding using one byte per character. This is a default encoding. (Using this encoding for non-English pages may cause the loss of non-English characters)
UTF-8	Unicode Transformation Format-8 encoding using variable number of bytes per character - lossless encoding of Unicode characters.
UTF-16	Unicode Transformation Format-16 encoding using two bytes per character.
UTF-16BE	Unicode Transformation Format-16 (big-endian) encoding using two reversed bytes per character.

## Supported Code Pages

Scripts Encryptor supports many code pages for the foreign languages. Here is the list of the currently supported code pages:

<Code Pages List>

To get an up-to-date list of supported code pages go to Help -> Command Line Help in the Scripts Encryptor menu.

INFORMATION: Although Scripts Encryptor has support for many international code pages you might not be able to use all of them by default as some of them might not be installed on your system. To check code pages installed in your Operating System go to Control Panel, and then to Regional and Language Options.

## Deprecated Command Line Parameters

In case you were using command line parameters with a previous version of the Scripts Encryptor refer to the manual for the deprecated command line parameters [here](#).

## Examples of Using Command Line Parameters

1. `screnc test.htm encode.htm`

Encode "test.htm" into "encode.htm" using Windows Script Encoder method.

2. `screnc /f test.htm`



Encode "test.html" using Windows Script Encoder method and overwrite it.

```
3. screnc /e html test.txt test1.txt
```

Treat "text.txt" as an HTML file, and encode it into "text1.txt" using Windows Script Encoder method.

```
4. screnc test.asp "c:\myDir\test.asp"
```

Encode "test.asp" into "c:\myDir\test.asp" using Windows Script Encoder method.

```
5. screnc "*.asp" "C:\Users\John Doe\Documents\Test"
```

Encode all ASP files in the current folder using Windows Script Encoder method and place them into "C:\Users\John Doe\Documents\Test" folder.

```
6. screnc -s -f "*.js|*.vbs" /op=s /sct=""
```

Scramble all JavaScript and VBScript files in the current folder with minimum scrambling options and overwrite them. No status window is displayed, and Windows Script Encoder method is not used.

```
7. screnc -e asp *.* "C:\Users\John Doe\Documents\Test" /op=o
```

(Available in the Premium Version only)

Organize ALL the files in the current folder as ASP files. Place them into "C:\Users\John Doe\Documents\Test" folder.

```
8. screnc -e asp -xl *.inc "C:\Users\John Doe\Documents\Test"
```

Encode all the files with .inc extension as ASP files. Do not add the @LANGUAGE directive at the top of each file. Place the results into "C:\Users\John Doe\Documents\Test" folder.

```
9. screnc -l vbscript test.html encode.html
```

Encode "test.html" into "encode.html". When a script tag with no language attribute is found, VBScript is assumed to be the default language.

```
10. screnc /op=s /lf=1 /htm="" /sct=ar /xtr="" /fe=utf-8 /rpt "*.htm?" "Compacted HTML"
```

Compact all HTML files in the current folder by scrambling them with minimum settings (only adding and removing of extra semicolons is required, do not escape non-ASCII characters). Save results with UTF-8 file encoding into "Compacted HTML" folder. Use only Line Feeds at the end of each line while saving files. Generate on-screen report.

```
11. screnc /op=s /htm=gtmcux /sct=qcmæ /ret=6 "C:\Users\John Doe\Documents\*.htm"
"C:\Users\John Doe\Documents\Test" /isf
```

Scramble all the .htm files in the "C:\Users\John Doe\Documents" folder (including subfolders) and place the results into "C:\Users\John Doe\Documents\Test" folder. Use the following options: Escape HTML tags, escape HTML text, use random escaping, keep HTML comment tags, allow scrambling of URLs in the HTML, with the use of short escape sequences. Also escape quotes and code at random in the JavaScript blocks, add missing semicolons. Set the random escaping threshold to 6 (middle value). Encode the result using Windows Script Encoder method.

```
12. screnc /cp="windows-1250" /f "*.js?|*.vb?" /rpt /sct==e
```

Rewrite each file with the .js, .jse, .vbs, .vbe, etc. extension in the current folder with the "Central European (Windows)" code page. Do not perform encoding with the Windows Script Encoder method. Generate on-screen report to see the results.

13. `screnc /op=s /awse="JavaScript, JavaScript1.2, JavaScript1.3" /htm=a /uid="1" /rpt="C:\Users\John Doe\Documents\results.txt" "*.htm?|*.asp|*.js" Test`

Encode all HTML, ASP and JavaScript files in the current folder and place the results into "Test" folder. Apply Windows Script Encoder method only to the HTML and ASP script blocks with the JavaScript, JavaScript1.2, or JavaScript1.3 language tag. Do the "all-inclusive conversion" to update all <script> tags within HTML and ASP pages if corresponding .JS file was encoded. Generate a report in the "C:\Users\John Doe\Documents\results.txt" file, and use ID equal to 1 in it.

14. `screnc /op=s /tp=v /lf=c /sct=mau "*.vbs" "c:\My folder\VB files\Encoded" /cf`

Scramble all VBScript files in the current folder by doing random capping and adding end-line colons. Then encode the result using the Windows Script Encoder method. Create "c:\My folder\VB files\Encoded" folder and save resulting files into it after changing the file extensions to reflect the encoding.

15. `screnc /op=s /htm+=gt /sct+=qcaru "test.htm" "test_scrambled.htm"`

Scramble and encode the "test.htm" file in the current folder with the maximum scrambling options on and save it as "test\_scrambled.htm" file

16. `screnc /op=rc /tp=m /lf=c /isf "*.c|*.cpp|*.h" "C:\Users\John Doe\Documents\Test C++"`

Remove comments from all .c, .cpp, and .h files in the current folder (including all subfolders), treat these files as (advanced) C++/MFC files and place results into "C:\Users\John Doe\Documents\Test C++" folder. Use Microsoft-specific Line Feed & Carriage Return at the end of each line.

17. `screnc /op=o /htm-=s "C:\My files\*.htm?|C:\My files\*.asp" "C:\My files\Converted Files\Organized" /isf`

(Available in the Premium Version only)

Organize all HTML and ASP files in the "C:\My files" folder, make sure not to use comment tags on script blocks within them though. Include all subfolders as well, and place the results into the "C:\My files\Converted Files\Organized" folder (note that this folder must exist before the conversion).

### Bug Report And Feedback

Please let us know about any glitches that you may find in Scripts Encryptor, also feel free to post your comments and suggestions through feedback. This will improve any future versions of this software. Plus if we find your report/suggestion useful we will give you a new improved version absolutely free of charge.

Thank you for choosing Scripts Encryptor!

---

**New!** Check other latest software [FREE](#) to download!

**Report Errors and Glitches** (IMPORTANT: Include detailed description of the glitch and the part of the script you're referring

to.)

**Send Comment/Suggestion**

www.dennisbabkin.com Copyright (c) 2003-2011  
Last updated: August 29, 2011