

Cweed4 V4.04 2022 Jan 14

Cweed4 is a C source file weeder for Windows, written in Win32Forth.

Download Cweed4 here : <https://www.inventio.co.uk/Cweedexe.htm>

It only changes whitespace (formatting) and does not otherwise change the content of the file. This is analogous to weeding a garden, where the plants are left untouched but the bits between (weeds) may be seriously modified.

Cweed4 is an experiment in specifying a coding standard in the form of an executable program.

Table of Contents

1.1.	Files used.....	2
1.2.	Recompiling Cweed4.....	2
1.3.	What Cweed4 shows you.....	3
1.3.1.	Refresh	3
1.3.2.	Show file +	4
1.3.3.	Show file -	4
1.3.4.	Show indents.....	4
1.3.5.	Show #.....	4
1.4.	What Cweed4 changes.....	5
1.4.1.	Replace tabs : 4	5
1.4.2.	Barr Rules 3.1	5
1.4.3.	-Trailing	5
1.4.4.	-2 Blanks.....	5
1.4.5.	// to /* */	5
1.4.6.	Weed	6
1.4.7.	Fix indents	6
1.4.8.	PC->Unix.....	6
1.4.9.	Unix->PC.....	6
1.4.10.	Restore	6
1.5.	Cweed4 Help	6
1.6.	Cweed4 options	7
1.7.	How Cweed4 works.....	8
1.8.	Change Log.....	8
1.8.1.	New in V2.3	8
1.8.2.	New in V2.4	8
1.8.3.	New in V3.0	8
1.8.4.	New in V3.1	8
1.8.5.	New in V3.2	8
1.8.6.	New in V3.3	9
1.8.7.	New in V4.0	9
1.8.8.	New in V4.1	9
1.8.9.	New in V4.2	9
1.8.10.	New in V4.21	9
1.8.11.	New in V4.22	9
1.8.12.	New in V4.23	9
1.8.13.	New in V4.24	9
1.8.14.	New in V4.25	9
1.8.15.	New in Cweed4 V4.01	9
1.8.16.	New in Cweed4 V4.02	10
1.8.17.	New in Cweed4 V4.02 2021 Nov 11	10
1.8.18.	New in Cweed4 V4.03 2021 Dec 21	10
1.8.19.	New in Cweed4 V4.04 2022 Jan 14	10



Figure 1 Cweed4 Information screen (Refresh)

1.1. Files used

Cweed4 uses the following Forth source files :

BigStrings.f	BigReplace etc. for strings > 255 bytes
CRC32.f	Cyclic Redundancy Check
Cweed4_script.f	loaded at compile-time and run-time
Files.f	parses the C file
Graphics.f	some Windows graphics support
Help.f	the help text
MD5.f	Message Digest 5
Menu.f	the GUI user menus
NumName.f	converts an MD5 hash into a pair of English words
walk.f	through sub-directories
Cweed4.f	the main load file (loads all of the others)

and :

Cweed4.bmp	the icon bitmaps
------------	------------------

Several C files are also included to test the program.

1.2. Recompiling Cweed4

The easy way, from a command window (cmd.exe) type :
go <enter>

The batch file will recompile and run Cweed4.exe. You must close any running Cweed4 program first.

Or you can double click on Win32For.exe to start the Forth system, and type :
tt <enter>

This needs the files *Win32For.img*, *WinCon.dll* and *kernel.bin* in the same directory.

Type :

fload Cweed4 <cr>

The executable Cweed4.exe will be automatically created in the current directory.

Double click on Cweed4.exe to run it, or type

Cweed4 <cr>

at the Forth "ok" prompt to run the program. You can also type **"ttt"** to run Cweed4.

Configuration options are at the start of the file Cweed4.f .

1.3. What Cweed4 shows you

1.3.1. Refresh

Numbers of each the following characters:

Tabs	(0x09)
Linefeeds	(0x0A)
Pagebreaks	(0X0C)
Carriage returns	(0x0D)
Control characters	(0 to 0x1F)
Del characters	(0x7F)
Characters with bit 7 set	(0x80 to 0xFF)

Shows formatting features :

Double slashes // single line C++ comments

Lines with more than 80 characters

Lines with only spaces

Lines with trailing spaces

Two blank lines in a row

"Floating pointers" : "char* foo" instead of "char *foo"

Lint Saves/Restores shows numbers of "//lint -save" and "//lint -restore" strings in the file.

If there is an error, shows a line number and

a) if there is an "-e" after a "-restore"

b) if there are one or more spaces between the "/" and "lint".

c) shows if the order of save/restore is wrong (lint stack underflow)

Shows the file format type : PC format, Unix format or unspecified.

Warns if the last line of the file is not correctly terminated with a carriage return and linefeed (PC format) or linefeed (Unix format).

Warns if the file has 1000 or more lines.

Hexadecimal Sum of the characters in the file.

Hexadecimal CRC32 of the file as per [ISO-3309] International Organization for Standardization, "Information Processing Systems--Data Communication High-Level Data Link Control Procedure--Frame Structure", IS 3309, October 1984, 3rd Edition

MD5 hash of the file.

Gives a "Pass" or "Fail" summary for characters in the file, file size and line length.

Gives a summary of indentation errors, or 'pass' if none.

1.3.2. Show file +

Shows the file with comments. Use the up and down cursor keys, PageUp, PageDown, Home and End to view the whole file.

Spaces, carriage returns and tabs are displayed as special characters – this is what is actually in the file, not a formatted interpretation of it.

Linefeeds are not shown, as they are implied by a new line.

Files must be less than 16384 lines long, and each line must be less than 252 bytes long – this is to limit the size of the fixed line array that it needs.

1.3.3. Show file -

Shows the file without comments and without unnecessary whitespace. Use the up and down cursor keys, PageUp, PageDown, Home and End to view the whole file.

All whitespace is reduced to one space (except linefeeds).

All `/* ... */` and `//` comments are ignored.

This is what the C compiler sees.

1.3.4. Show indents

Shows any lines with indentation errors.

Lines are either within a function's parameter definition or not – different rules apply to each type.

"Parameter" indentation looks at all lines within the function to determine the correct tab position for the identifiers. Leading `*`'s are dropped back before the tab position. The entire line is compared to the defined format.

"Normal" indentation uses `{`, `}`, `(` and `)` to define the correct indent level, provided these characters are not commented out. The indentation only applies to the first non-space character on the line.

`/* ... */` comments can have a less restrictive rule that the text must be indented at or beyond the calculated position. This allows "free-form" comments. See `Files.f : IndentOK? AllowGreaterThan` ;

1.3.5. Show #

Show the number of each character in the file. If there are characters with bit 7 set a prompt is given to press any key to view them.

1.4. What Cweed4 changes

Clicking on buttons which are not marked “Select file “ or “Show...” will immediately change the selected file.

The cycle is : open file, change it, save it .

There is no query to confirm the overwrite unless the file has been changed by another program. Click on “Restore” to put back the file originally opened by “Select file”.

If the file is too big, has too many lines or one or more lines that are too big for the internal buffers a warning will be displayed, and no changes will be made to the file. This is to protect zip and exe files etc. from being mangled.

Please take a backup before using Cweed4 to change your file – it is believed to work, but it is a computer program.

1.4.1. Replace tabs : 4

Replace all tab characters (0x09) in the file by the correct number of spaces to align the following text to the next tab position. Each tab position is interpreted as 4 characters - the default value. This will preserve the formatting for text editors with their tab value set to 4, and will remove all tabs.

Pressing the number keys 2, 3, 4, 6 and 8 will convert tabs to that number of spaces (if the action is confirmed). This will preserve the formatting for text editors with their tab value set to something other than 4, and will remove all tabs. This is useful if the file has been edited using such an editor.

1.4.2. Barr Rules 3.1

Press F2 to apply the Barr Coding Standard (2018) Rules 3.1 Spaces a, g, h, i and j. This corrects the number of spaces around certain C keywords, following :
Barr Coding Standard (2018) White Space Rules 3.1

https://barrgroup.com/sites/default/files/barr_c_coding_standard_2018.pdf

<https://barrgroup.com/embedded-systems/books/embedded-c-coding-standard/white-space-rules/spaces>

1.4.3. -Trailing

Removes all trailing spaces (space characters that are between the last non-space character and the end of the line). These characters have no meaning to C compilers and just take up space in the file.

1.4.4. -2 Blanks

Converts sequences of 2 or more blank lines into a single blank line. These extra lines have no meaning to C compilers and just take up space in the file.

1.4.5. // to /* */

Converts C++ // single line comments to C style . * ... *. Comments.
It does not convert C++ comments that are embedded inside a /* ... */ pair.

Note that the following line, which is usually allowed by compilers :

```
//  /* some doubly commented text */
```

will be converted to :

```
/*  /* some doubly commented text */ */
```

which is usually not allowed. You must hand edit this yourself...

1.4.6. Weed

Replaces all tabs by the equivalent 4 character tabs.

Removes all trailing spaces.

Removes all double blank lines.

Adds a carriage return and linefeed or linefeed (depending on file format) to the end of the file if the last line does not have them. Most text editors add these for you, and most C compilers need them (they ignore the last line in the file if it is not correctly terminated).

One click and the file is weeded!

1.4.7. Fix indents

Adjusts the indentation of each “normal” line and the format of the entire line inside a function parameter list.

This is not included in the “Weed” function because it is so complicated that it cannot be guaranteed to be 100% accurate.

Type `ctrlI` to ignore the “parameter” list formatting errors and thus not try to fix them.

1.4.8. PC->Unix

Removes all carriage returns from the file.

1.4.9. Unix->PC

Adds a carriage return to the end of each line (having first removed all carriage returns from the file).

1.4.10. Restore

Puts back the file originally opened by “Select file”.

There is no query to confirm the overwrite unless the file has been changed by another program.

1.5. Cweed4 Help

There is currently one help screen. Click on “Help” or press F1 to see the help screen

1.6. Cweed4 options

There are some control key indentation options listed in the indent help screen.

ctrlE allow an extra space inside a /* ... */ comment block.

Press again to return to the normal state. Click on "Show indents" to see if the extra space mode is active.

Default :

```
/*  
*  
*  
*/
```

With extra space :

```
/*  
 *  
 *  
 */
```

ctrlF show Full indent details for each line

```
123 {1} (0) /*- c- d- s- I-+ GC 0 NS 4 1 some text here
```

indicates line 123, {}indent level of 1, () indent level of 0, "normal" Indent state with correct formatting.

The indent level is incremented by one for each '{' or '(' and decremented by one for each '}' or ')' (provided these characters are not commented out).

Negative levels are indicated by {-} or (-).

For the following a '+' means active, a '-' means not active or not found

/*- means this line is not within a /* ... */ comment block

c- means this line has 'case' as its first word

d- means this line has '#define' as its first word

s- means this line has a class specifier (public:, private: or protected:) as its first word.

I- means an indentation or formatting error. (I+ means no error)

GC x indicates a GetCurrentIndent value of x (in characters)

NS x indicates a NameSpace Indent value of x (in characters)

The '1' indicates within a /* ... */ comment block.

'2' indicates a single line comment (//) on this line.

'|' neither of the above.

1.7. How Cweed4 works

The selected file is copied into the InputBuffer.

A copy is made when the file is selected so that it can be restored.

The file is then split up into lines and put into a LineArray buffer which has a fixed length for each line. Each line has several attributes such as indentation, comment status and length.

This allows lines to be accessed by calculation, rather than searching.

A line is a sequence of characters terminated by a linefeed character.

If the last character in the line is a carriage return the line attribute is set accordingly. Carriage returns are not stored in the line data.

Carriage returns which are not followed by a linefeed are ignored.

A global variable LineNumber allows LineStartPtr and LineLength to refer to the current line.

DoLine takes an execution token and performs that action on every character in the current line.

DoLines takes an execution token and performs that action on every line in the current file.

An output file buffer accumulates re-formatted lines in sequence, so that they can be written back to the file.

1.8. Change Log

1.8.1. New in V2.3

Added BlocksToFile conversion function : if you open an old-style Forth block file (with 64 character lines) Cweed4 will not be able to copy the file into its line buffer. Press 'ctrlR' and click on OK to add a CRLF pair after every 64 characters and save this as your-original-filename.f .

1.8.2. New in V2.4

Changed the file filter from "*.c" to "*.c*" which then includes C++ source files.

Modified the "Fix Indents" function to not indent a '{' when the keyword "namespace" appears at the start of a line. This is to save whitespace at the start of every program line if a C++ namespace is used.

Also tidied up the "Show Indents" function – it now checks for brackets on the last line of the file.

1.8.3. New in V3.0

Bug fixes to namespace indentation.

Adding 2-space indentation for public: , private: and protected: .

Adding CRC32 and MD5 values for the file.

Block comments indented by one space.

Configuration options are now at the top of file Cweed4.f .

1.8.4. New in V3.1

Lint report shows numbers of "//lint –save" and "//lint –restore" strings in the file. Also checks for "–e" after a "–restore" and a one or more spaces between the "//" and "lint".

1.8.5. New in V3.2

Fix to non comment block lines being indented if they start with a '*'.

Folder report added – select a file, then click on Show Folder to perform a scan of all C, C++ and header files in the containing folder.

1.8.6. New in V3.3

Added Windows7 support in the Win32Forth supplied in the Cweed4 package, also made the window re-paint after it is covered by another window.

1.8.7. New in V4.0

Fixed lack of update when the Cweed4 window is covered, added NumName text feature to give the file MD5 hash a human readable form – see NumName.f for details. The NumName.txt file list of words is now precompiled.

1.8.8. New in V4.1

Minor bug fixes in Folder Report and new NumName.txt file.

1.8.9. New in V4.2

“Select File” now defaults to All Files *.* ,made INDENT_SPACES changeable.

1.8.10. New in V4.21

Added 'F' to ShowFolder filtering only *.c and *.h files.
The Show Folder button or 'R' shows *.c??, *.h?? and *.ini files.
Also fixed bug in # tabs reporting (again).

1.8.11. New in V4.22

Added file format indication before the filename in ShowFolder file

1.8.12. New in V4.23

Indenting now ignores #ifdef, #ifndef, #else and #endif if the line contains double underscores :
__IGNORE_THISE_INDENT__
This is to prevent indentation in *.h files with the usual redefinition macro format, below.
Please make sure any #else and #endif has a double underscore in a comment on the same line :

```
#ifndef __HEADER_FILE_NAME__  
#define __HEADER_FILE_NAME__
```

... your code here...

```
#endif // __HEADER_FILE_NAME__
```

1.8.13. New in V4.24

Now packaged with Win32Forth6v05H 2019 Mar 02 which fixes a bug that forced its window to be too small.

1.8.14. New in V4.25

Added F2 key to add/remove "interior spaces" : [i] <--> [i] etc.
Press F2 to apply the Barr Coding Standard (2018) Barr Coding Standard (2018) Rules 3.1 Spaces.
Added source files for Win32Forth, so that source for system words can be viewed.

1.8.15. New in Cweed4 V4.01

New name – Cweed4 . Searching for “Cweed” online finds lots of sites about “weed” – “Cweed4” is easier to find. The version has also been reset to Cweed4 V4.01.
Added a custom file Cweed4.script for the Barr Coding Standard (2018) – this allows modifications to be made without recompiling Cweed4.

1.8.16. New in Cweed4 V4.02

The file Cweed4_script.f is now loaded when running the Barr 3.1 function.
This allows the rules to be changed and reloaded without re-compiling Cweed.
If the file Cweed4_script.f is not found in the directory where Cweed4 is running, the default file is used i.e. the one that existed when Cweed4 was compiled.

1.8.17. New in Cweed4 V4.02 2021 Nov 11

Changes to Cweed4_script.f to improve the Barr 3.1 coding rules.
Previous versions of Cweed4 V4.02 could be used if you simply update the Cweed4_script.f file, but this causes an “interesting” version control effect :
Cweed4.exe is compiled with the version of Cweed4_script.f that exists at the time that it is created. If this is different to the Cweed4_script that exists when Cweed4.exe is run you can get a different behaviour if you delete Cweed4_script.f (because it would use the pre-compiled words from an earlier Cweed4_script.f file).
Not sure how to handle this yet... Maybe compare a hash of the built in script to the script file, and issue a warning if they are different... Watch this space :-)

1.8.18. New in Cweed4 V4.03 2021 Dec 21

Changes to Cweed4_script.f to improve the Barr 3.1 coding rules.
Added ctrlQ to show and fix lowercase characters after a '//' comment
Increased Barr Coding Standard buffer size to \$400 to allow longer lines to be processed

1.8.19. New in Cweed4 V4.04 2022 Jan 14

Changes to Cweed4_script.f to improve the Barr 3.1 coding rules.
Used BigReplace to allow long strings to be processed.
Changed “Show Info” menu button to “Refresh”.

Howerd Oakford

howerd@inventio.co.uk www.inventio.co.uk