

Basic source code formatter

30-12-2012

1	Introduction.....	1
2	Usage	2
2.1	How to make it work from within the IDE.....	3
3	Details.....	3
3.1	Source code Preconditions.....	3
3.2	Indentation	3
3.3	Line breaks.....	4
3.4	Additional remarks	4

1 Introduction

This is a *simple* Basic source formatter, only meant to give correct and readable **indentation** to Basic code. This is also partly achieved by inserting line breaks before and/or after “keywords” where necessary.

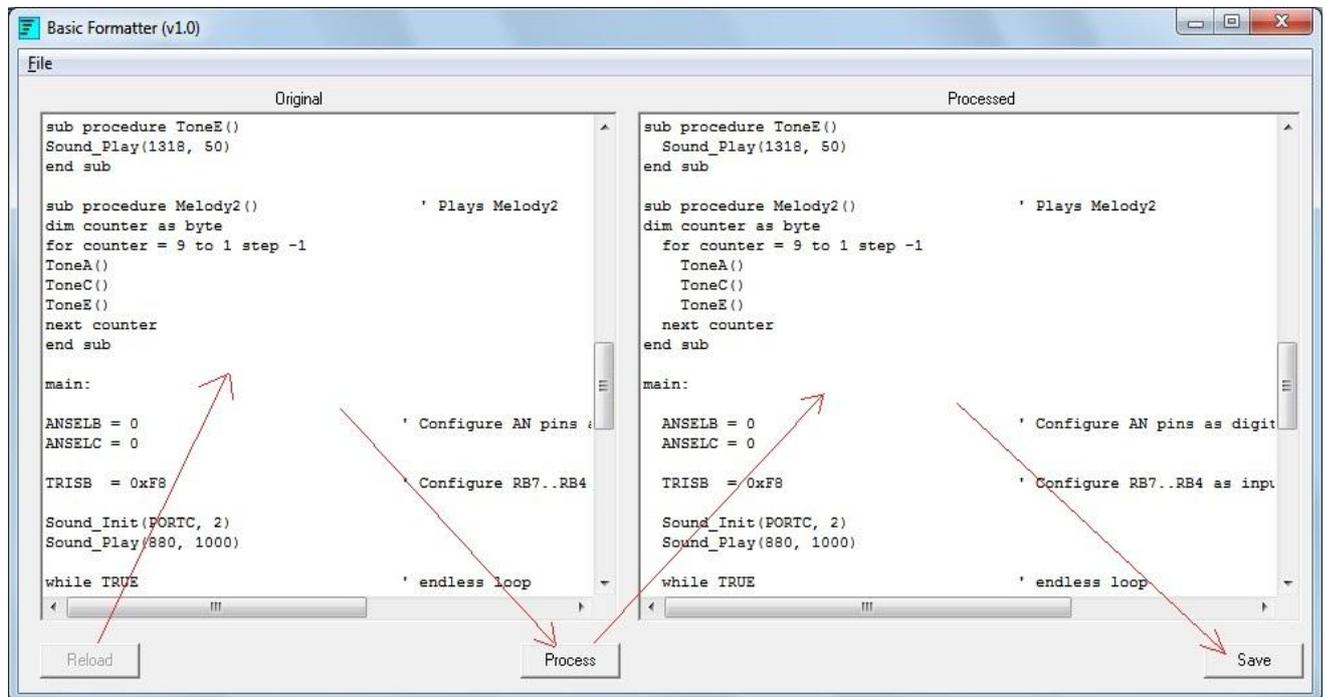
The tool can work stand alone or be started up from the IDE. In the first case the “File” menu in the tool has to be used to load a Basic source file, in the latter case the currently selected source in the IDE will be loaded in the tool.

In case the tool is to be started up from within the mP IDE, one of the IDE "tools" “must be programmed to start up the formatter and pass the currently selected file to it (*which has to be "saved" beforehand*).

The software: [Format_Basic.zip](#)

2 Usage

On startup, the basic source is loaded automatically (using the IDE startup) or manually (using the "File" menu) and the tool shows this window:



The left side pane shows the original (unprocessed) source code, and hitting the "Process" button produces the formatted (processed) code in the right hand pane.

This processed source code can be saved (overwriting the original code) by pressing "Save".

Additionally one can, before hitting "Process" trim (delete spaces) the code in the left pane. This is done with a popup menu (right mouse click inside the left pane). Two options are offered: trimming all lines, and trimming only the lines that do not start with a comment.

If wanted, one can reload a saved file in the left page by hitting "Reload".

Remarks:

- before hitting "Process" one can make changes in the original source code in the left pane, they will be taken into account.
- before hitting "Save" one can make changes in the "processed" source file in the right pane, they will be taken into account.

2.1 How to make it work from within the IDE

To make the tool work from within the mP IDE the following has to be done:

1. The tool must be added in the "Tools" (Tools menu --> Options --> Tools --> Toolx) menu:

Tool Name: Format Source

File Name: C:\Delphi Projects\PIC Related\Format Source Basic\Format_Basic.exe

Parameters: %UNIT_FILE_NAME

Macro: %HEX_FILE_NAME Full path and name of the out... Insert

Shortcut:

As you can see, the path to the tool must be given and one parameter: "%UNIT_FILE_NAME".

2. Additionally mP has to be told to accept changes in source files made by an external tool, this is done in the IDE menu "Tools -> Options -> Editor -> Editor settings":

If Opened File Is Externally Modified

Prompt for action Reload file, but do not prompt Ignore externally made changes

3 Details

3.1 Source code Preconditions

The tool expects the following to be true in the original source code (precondition):

- the keywords "include", "const", "typedef" and "symbol" should be at the beginning of a line (only spaces before it are allowed).
- keywords in general must always be surrounded by spaces, or start at the beginning of a line or end at the end of a line. The exception to this rule is "end" which can be followed by '!'.

3.2 Indentation

The **indentation** in the processed file is done according the detection of the following keywords:

- "Program", "module", "implements", "end sub", "end structure": the indentation of the keyword is zero, and so is the indentation of the lines following it.
- "include", "const", "typedef", "symbol", "main:", "sub", "dim" (if used outside a structure or used inside a subroutine definition), "structure" : the indentation of the keyword itself is zero, the indentation following it is 2.

- “dim”: if used inside a structure definition then the keyword itself has an indent of 2 and the following lines have an indent of 4.
- “while”, “for”, “if”, “asm”, “select case”, “case”, “do”: the keyword itself has the current indent, the indentation of the following lines is the current one plus 2.
- “end if”, “end asm”, “loop until”, “wend”, “next”: the indentation of itself and its following lines is decremented by 2.
- the indentation of lines that start with comments is not changed.

3.3 Line breaks

- the keywords “dim”, “end sub”, “while”, “for”, “end if”, “if”, “end asm”, “asm”, “end structure”, “structure”, “select case”, “case”, “do”, “loop until”, “wend”, “end select”, “next” and “else” *are always placed at a new line*, the line is split up for that if necessary
- text after the keywords “end sub”, “end if”, “end asm”, “asm”, “end structure”, “do”, “wend”, and “end select” on the same line is moved to the next line (the line is split up for that if necessary), unless the text is pure comments

3.4 Additional remarks

- Double spaces are removed in the following keywords:
 - Select case
 - end select
 - end if
 - loop until
 - end structure
 - end asm
 - end sub