Build Version Generator for mikroPascal, mikroBasic, mikroC

Author: VCC
Version: 1.0 (March 2014)
Version: 1.2 (January 2015)

Build Version Generator is a tool, which generates code, containing current build date of a mikroPascal/Basic/C project.

Features:
- It is started from a mikro compiler as a tool
- Saves project version in a separate file, as a library
- Generates version formatted as one, two, three or four numbers separated by user-specified separators
- In addition to numbers, a version string can contain user defined strings
- Build number can be autoincremented as an option
- In adition to saving version as a string, individual numbers are saved as integers (8/16/32-bit, un/signed)
- Build date and time are saved both as string and individual numbers
- Date string can have one of three separators "/", "." and " " and four formats of YYYY MM DD
- The application closes automatically after being started, waiting a specified time
- The generated library, containing version constants, is automatically saved when closing the application
- Except the autoclose timer constant, all options are saved in separate settings files

Features in version 1.2
- “Build” field can be disabled
- Leading and trailing spaces can be saved for text fields (“PreText”, “PostText” and separators).

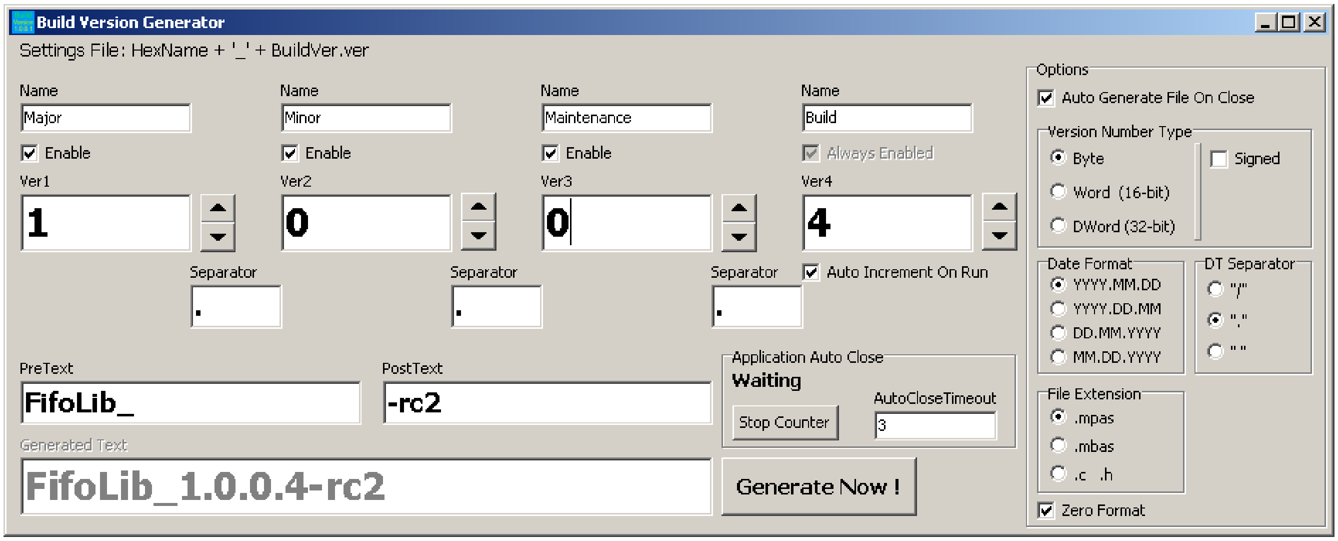
Requirements / Recommendations:
- The application must be run with **"%HEX_FILE_NAME"** parameter from compiler's IDE, including quotes.
- An IDE bug deletes the quotes from HEX_FILE_NAME parameter, so there must be two parameters, which are correctly saved: **"%HEX_FILE_NAME" %HEX_FILE_NAME** The second one has no quotes and is ignored by the application.

Limitations:
- Keeps track of only project version, not of individual sources.
- Since application options are saved in separate files (per project), settings revert to default on new projects. Thus, users have to configure settings for every new project.

Table of contents:

1. General view of the application
2. Settings
3. Application as a mikroe compiler IDE tool

1. General view of the application



Build Version Generator

Settings File: HexName + '_' + BuildVer.ver

Name: Major — ☑ Enable — Ver1: 1

Name: Minor — ☑ Enable — Ver2: 0

Name: Maintenance — ☑ Enable — Ver3: 0

Name: Build — ☑ Always Enabled — Ver4: 4

Separator: . Separator: . Separator: . ☑ Auto Increment On Run

PreText: FifoLib_ PostText: -rc2

Generated Text: FifoLib_1.0.0.4-rc2

Application Auto Close: Waiting — AutoCloseTimeout: 3

Options

☑ Auto Generate File On Close

Version Number Type: ● Byte ○ Word (16-bit) ○ DWord (32-bit) □ Signed

Date Format: ● YYYY.MM.DD ○ YYYY.DD.MM ○ DD.MM.YYYY ○ MM.DD.YYYY

DT Separator: ○ "/" ● "." ○ " "

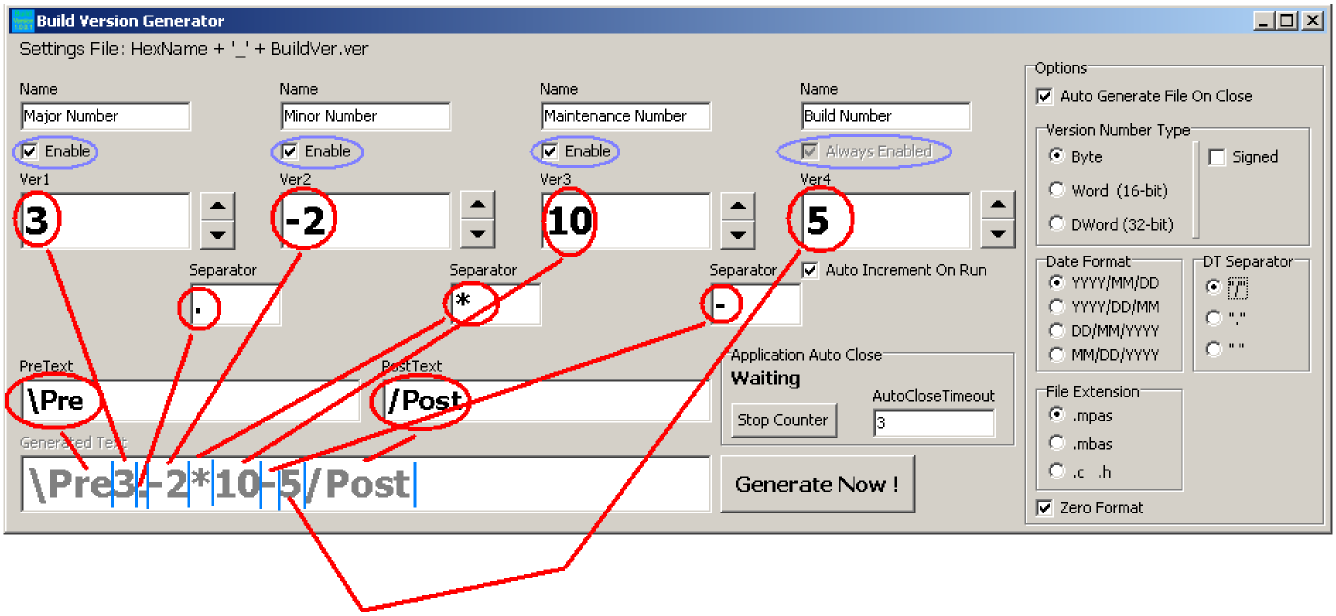
File Extension: ● .mpas ○ .mbas ○ .c .h

☑ Zero Format

Fig.1 Build Version Generator screenshot

The application window contains four version numbers, three of them being able to be switched off, three version number separators, one pre-text, one post-text, four version number names and various options. Each version number can be individually incremented / decremented, while the last of them has the option to be automatically incremented by application on startup. The four number names are used only as a version format track. They do not affect the version generation process.

After running the application, users can choose to modify either version numbers, various settings, or wait for the application to automatically close.



Build Version Generator

Settings File: HexName + '_' + BuildVer.ver

Name: Major Number — ☑ Enable — Ver1: 3

Name: Minor Number — ☑ Enable — Ver2: -2

Name: Maintenance Number — ☑ Enable — Ver3: 10

Name: Build Number — ☑ Always Enabled — Ver4: 5

Separator: . Separator: * Separator: - ☑ Auto Increment On Run

PreText: \Pre PostText: /Post

Generated Text: \Pre3.-2*10-5/Post

Application Auto Close: Waiting — AutoCloseTimeout: 3

Options

☑ Auto Generate File On Close

Version Number Type: ● Byte ○ Word (16-bit) ○ DWord (32-bit) □ Signed

Date Format: ● YYYY/MM/DD ○ YYYY/DD/MM ○ DD/MM/YYYY ○ MM/DD/YYYY

DT Separator: ● "/" ○ "." ○ " "

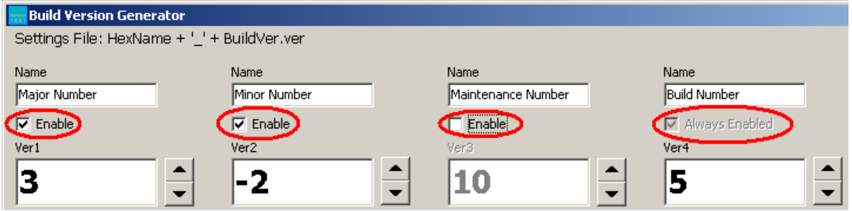
File Extension: ● .mpas ○ .mbas ○ .c .h

☑ Zero Format

Fig.2 Version string format

2. Settings

The Build Version Generator application has various options regarding the use and generation of version string, build date-time and application behavior. The first settings a user may interract with are the version numbers enable / disable checkboxes, see Fig.3. Users should notice that the fourth checkbox, for the fourth version number is always checked and grayed, because at least one version number must exist. Enabling or disabling these version numbers adds / removes them to / from version string, see Fig.2.



Build Version Generator

Settings File: HexName + '_' + BuildVer.ver

Name: Major Number
☑ Enable
Ver1: 3

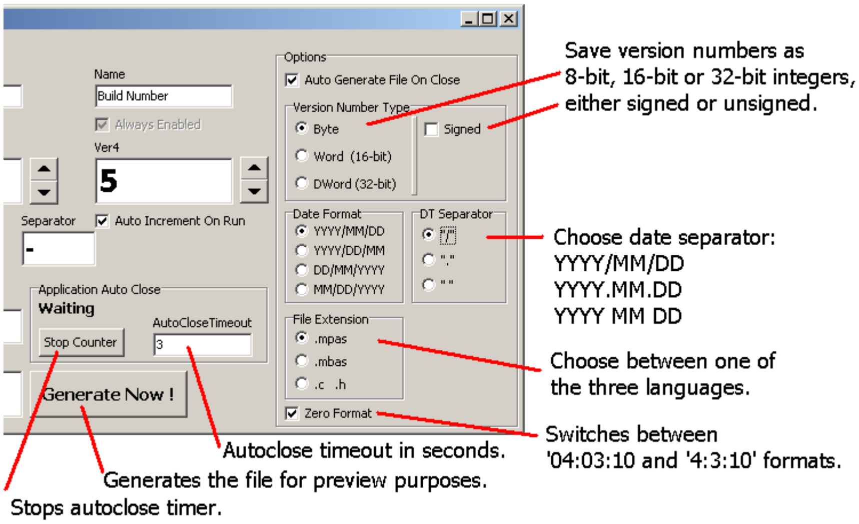
Name: Minor Number
☑ Enable
Ver2: -2

Name: Maintenance Number
☐ Enable
Ver3: 10

Name: Build Number
☑ Always Enabled
Ver4: 5

Fig.3 Version numbers enable / disable checkboxes

As mentioned before, the application closes automatically after a predefined timeout. This timeout allows users to interract with the application as needed. If users move the mouse cursor over the application window, the autoclose timer is reset to prevent closing. As long as the mouse cursor is kept over the window, the timer is constantly reset. Moving the mouse outside of the window perimeter, the autoclose timer counts again, and finally, it closes the application. Further, the autoclose timer can be disabled by pressing the *Stop Counter* button, see Fig.4. This way, the application will not close automatically, allowing the user to freely interract. However, the timer will be active again on next application run. It must be mentioned that moving the mouse over or outside the application window is not always detected, thus either allowing the timer to count, or keeping it reset. The timeout constant is saved in application's settings file, therefore being the same for all user projects.



Name: Build Number
☑ Always Enabled
Ver4: 5

Separator: -
☑ Auto Increment On Run

Application Auto Close
Waiting
Stop Counter
AutoCloseTimeout: 3

Generate Now !

Options
☑ Auto Generate File On Close

Version Number Type
● Byte
○ Word (16-bit)
○ DWord (32-bit)
☐ Signed

Date Format
● YYYY/MM/DD
○ YYYY/DD/MM
○ DD/MM/YYYY
○ MM/DD/YYYY

DT Separator
● "/"
○ "."
○ " "

File Extension
● .mpas
○ .mbas
○ .c .h

☑ Zero Format

Save version numbers as 8-bit, 16-bit or 32-bit integers, either signed or unsigned.

Choose date separator:
YYYY/MM/DD
YYYY.MM.DD
YYYY MM DD

Choose between one of the three languages.

Switches between '04:03:10 and '4:3:10' formats.

Autoclose timeout in seconds.

Generates the file for preview purposes.

Stops autoclose timer.

Fig.4 Application settings

The generated libraries, cotaining version information, can be automatically saved before closing the application, or can be saved on demand, using the *Generate Now* button. To make the application automatically save the library, the *Auto Generate File On Close* checkbox must be checked, see Fig.4.

Settings, directly affecting version string, are grouped as *Options* on the application window. With Version Number Type, users can choose between one of the three integer representations, 8-bit, 16-bit and 32-bit. The generated types can be either signed or unsigned as set by *Signed* checbox.

Build date-time is also saved as a string by *"date time"* format. Date can be further formatted as *YYYY.MM.DD*, *YYYY.DD.MM*, *DD.MM.YYYY* or *MM.DD.YYYY*. Also, the date separator can be "/", "." or " ".

The programming language can be choosen from *File Extension* groupbox, as *Pascal*, *Basic* or *C*. For mikroC, the application generates two files, a *.c* file and a *.h* file.

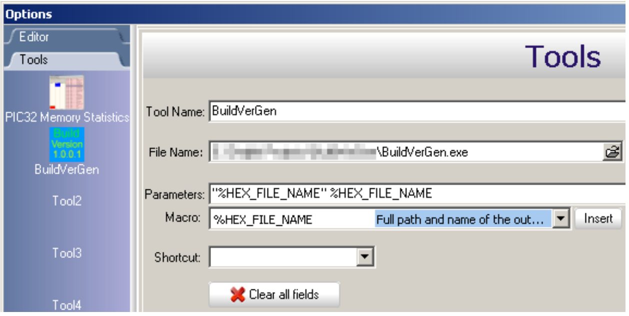
To keep the string length constant, date-time numbers can contain zeros, if *Zero Format* checkbox is checked. This setting does not apply to generated version string, because version numbers can contain many digits.

It must be mentioned that the generated library is saved in the same folder as the .hex file, passed as command line parameter to the application. It is saved as *BuildVer.mpas*, *BuildVer.mbas*, *BuildVer.c* and *BuildVer.h*. Users should write their own routines to read the version constants saved in these files.

3. Application as a mikroe compiler IDE tool

This application is usually run as an IDE tool, see Fig.5. It requires as a command line parameter to have the project hex file. From the hex filename, it generates a settings file formatted as *HexName + '_' + BuildVer.ver* . This *.ver* file is a *ini* file, containing all the application settings, except autoclose timeout and windows position/size. Running the application without command line parameters, generates no *.ver* file and no library is generated.

If a user renames a project, then the *.ver* file must be renamed accordingly. Otherwise, the application will not find the file and will assume a first version.



Tool Name: BuildVerGen

File Name: [redacted]\BuildVerGen.exe

Parameters: "%HEX_FILE_NAME" %HEX_FILE_NAME

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

Fig.5 mikro compiler IDE tool settings for Build Version Generator

As command line parameters, users should fill in: **"%HEX_FILE_NAME" %HEX_FILE_NAME**
The first parameter must contain quotes, for path that contain spaces, while the second parameter is used to avoid a bug in the IDE, which makes the quotes disappear. The second parameter is ignored by the application.