

Basicbuilder 3.0

By Martin Warmer, mmartin@xs4all.nl

Project page at <http://basicbuilder.sourceforge.net>

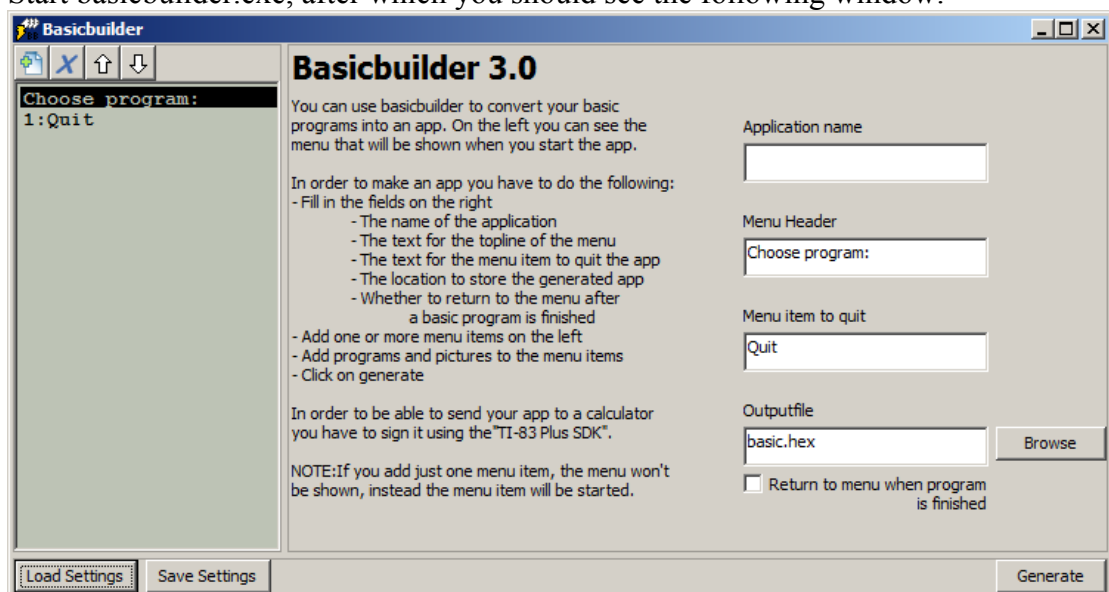
Basicbuilder is a program to package ti(83+/84+) basic programs into an app that runs on a ti83+/84+. When you start the programs from the app they will be copied to ram, executed and deleted afterwards. This is like unarchiving, running and archiving a program, only when they are included in an app this is all happens automatically. Another advantage of using basicbuilder is that everything is compressed, thus allowing more programs to be stored on your calculator than without using basicbuilder.

Requirements

You need to have the ti83+ sdk installed to be able to generate applications. You can find it at <http://education.ti.com/us/resources/developer/83/hilight/hilight.html> Just follow the instructions on screen to install it.

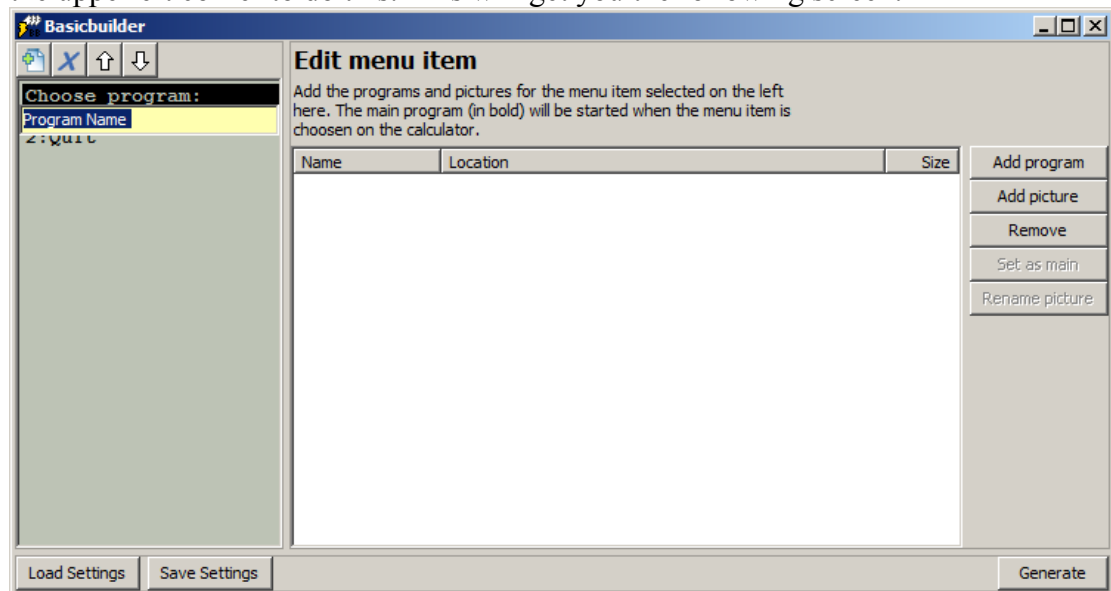
How to use basicbuilder

1. Find all basic programs, subprograms and pictures you want to include in the app and make sure they are available on your computer.
2. Start basicbuilder.exe, after which you should see the following window:



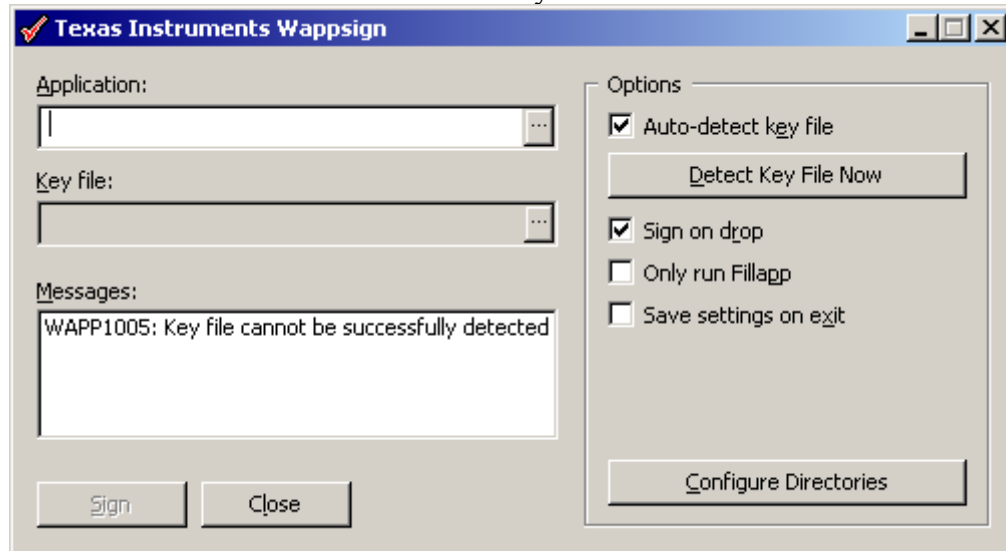
3. The above screen consists of two parts. The left part is the menu which will appear when you start the app on the calculator. The right part contains the settings. At this screen you can load in previously saved settings using the button in the lowerleft corner. Or you can do the following:
 - a. Fill in the name of the application as it will appear in the apps menu. The name should start with an uppercase character to make it appear in the right place in the appsmenu.
 - b. Fill in the menu header, this is the text it will appear above the menu in which you can choose which program to execute. (For example, the menu header of the appsmenu is "APPLICATIONS")
 - c. Fill in the text for the menu item which exits the app without executing any program.

- d. Fill in the location of the outputfile, you can use browse to find the location. NOTE: The output file is not ready to be sent to the calculator, it needs to be signed first.
 - e. The last thing you do is choose whether the application should show the menu again when it has finished executing a program or that it should exit to the home screen.
4. After having filled in this screen you can start to add a menu item. These are the separate “programs” that can be started inside the application. Press the button in the upperleft corner to do this. This will get you the following screen:



5. On the left you can type in the name of the menu item. Or you can select the menu header (Choose program:) or the menu item to quit (Quit) to return to the previous screen (shown in step 2)
6. Now you can start adding the basic programs and pictures required for this “program” using the add program and add picture buttons on the right. The bold item is the main basic program. This basic program will be started when you select the menu item on the calculator.
7. After having added all the required basic programs and picture you can return to step 4 and add another menu item. Or if you’re finished you can press the generate button in the lowerright corner.
8. You should see a message telling you how much space was saved by compression and how much space is still left in the app. If you don’t see that message, you are probably looking at an error message. You should correct the error, which probably means adding or removing some programs to get the correct size.
9. Start the wappsign program included in the ti83+ sdk. If you have installed the sdk it should be available from the start menu. If it isn’t you probably have a version of

the sdk that is too old. When it is started you should see this window:



10. Make sure the auto-detect key file, sign on drop and only run fill app options are checked like in the above screen.
 11. Find your file in the windows explorer and drop it on the window. It should say that it signed the application in the messages section and you should see an application file in the explorer.
- NOTE: There are some problems when loading the .hex file in the ti flash debugger, it is preferred that you sign the file before loading.

Known problems

Basicbuilder works has been tested to work under windows 2000, xp and vista. But the tests under windows 95, 98 and me have failed. The suggested solution for this problem is to find a pc running a NT-based version of windows and use that computer to generate the basicbuilder applications.

Support for asm programs

You can run asm program using the normal syntax of `asm(prgmPRGMNAME)` And you can start them using `asm(nr)` as well when they are included in the app. Asm programs you use shouldn't use the last 12 bytes of `appbackupscreen` because basicbuilder uses them.

Thanks to:

- Benny Baumann and all other beta testers for beta testing the program
- Thomas Dickerson for the original wxpython port (although the current gui is another complete rewrite gui)
- Dan Englander for the menu routine (which I modified a lot and fixed one or two bugs in it)
- Jeff Frohwein for the gameboy Z80 port of pucrunch unpacker
- Pasi Ojala for pucrunch (which I use to compress everything)

Programs used to create this:

- NSIS (nsis.sourceforge.net)
- Python (www.python.org)
- Py2exe (www.py2exe.org)
- Vi Improved (www.vim.org)

- WLA-DX (users.tkk.fi/~vhelin/wla.html)
- WxPython (www.wxpython.org)

Version history:

3.0:

- Basicbuilder is now opensource!
- The gui has been completely rewritten in wxpython (original gui was in delphi)
- The assembly code has been ported to WLA-DX (the previous assembler was Z80Eclipse)

2.1:

- Now runs at full speed on Ti83+ silver edition and ti84+ (silver edition) instead of running at ti83+ speed. (2.5x speedup)

2.0:

- Basicbuilder will leave out support for features (like recalling pics) if you don't use them, thereby saving which can be used for including programs
- Added compatibility with omnicalc, symbolic, and other programs which introduce new basic functions
- Doesn't display the menu when you only add one menu item
- Added support for adding multiple files at once
- Now you can make your basic program return to the menu after executing
- Solved some problems with the gui not displaying correctly
- Solved a problem when recalling pics from the app without an initialised graph
- Did some size optimisation in the code
- Improved compression of pictures in some special cases
- Improved program compression in cases where programs are almost the same
- Fixed a rare bug that happened when decompressing programs or pics
- Fixed a bug in the autorename function
- The decompiler now also generates an ini file and autorenames programs back

1.2:

- Added support for up to 99 subprograms
- Added support for asm programs
- Added auto renaming of programs (prgmABC to asm(3)

1.14:

- Fixed a bug in deleting the basic programs afterwards

1.13:

- Fixed a bug in the user interface when moving programs up and down

1.12:

- Now you can see how many bytes you have to add to reach the 13kb limit
- The maximum size was reduced a little bit because appsign can't handle apps that take up the full 16kb

1.11:

- Fixed a bug that caused ti83 programs to not be included correctly in some cases

1.1:

- Redesigned parts of basicbuilder to make it always work when python is installed

1.02:

- Fixed a bug that caused it to only work under the nt series of windows. It should also work under the 9x/me series now.

1.01:

- Fixed a bug with 14 character menu items

1.0:

- First public release