EasyConsole - create a console application in VB just

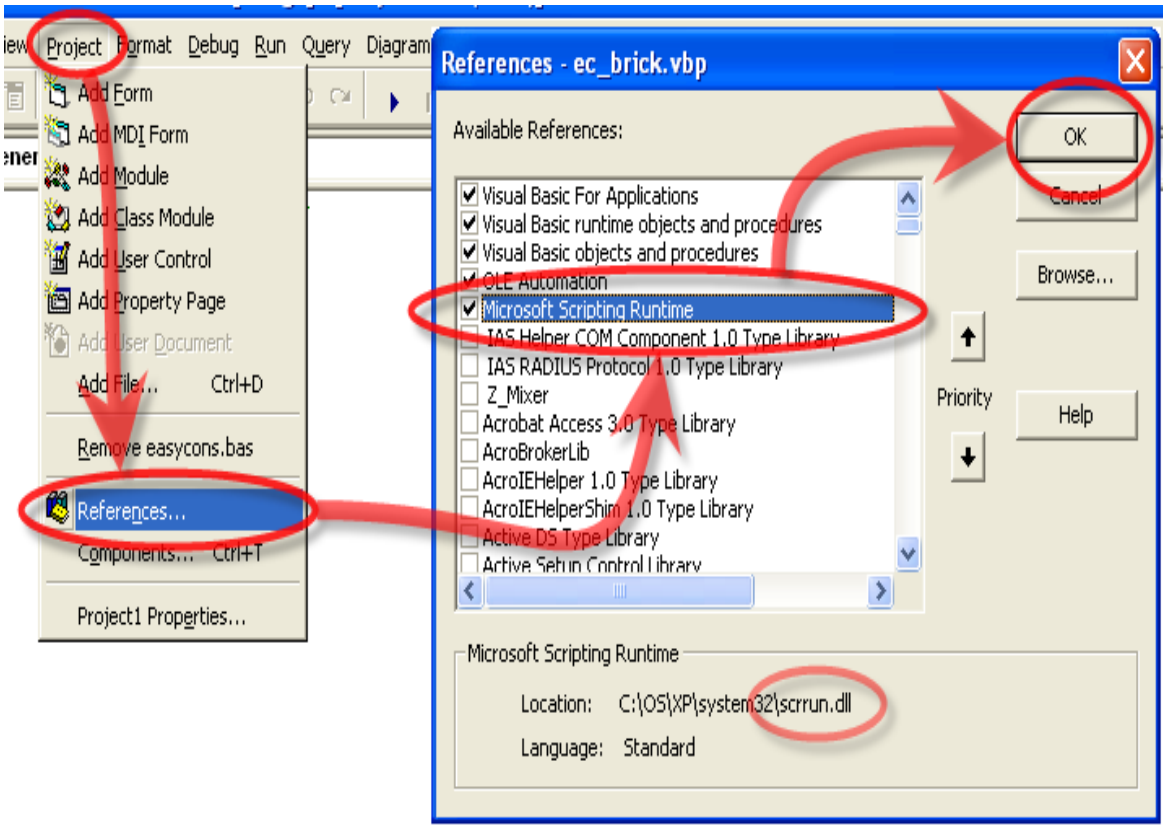
In this issue: New version, bug reports and questions on the use of brick.
In the "Projects" is the theme of the satellite - in her discourse on technology, praise and criticism.

This brick allows you to create a VB (VB6) **full console application,** attaching to this record a minimum of effort: adding in the project about 10 lines of code

Himself brick - the usual unit EasyConsole (easycons.bas):
Latest version:

To use a brick, you need to perform only three steps:

1. Add **EasyConsole** module into your project.
2. Connect to Project → References a type library «Microsoft Scripting Runtime», known to some as the townsfolk FSO.



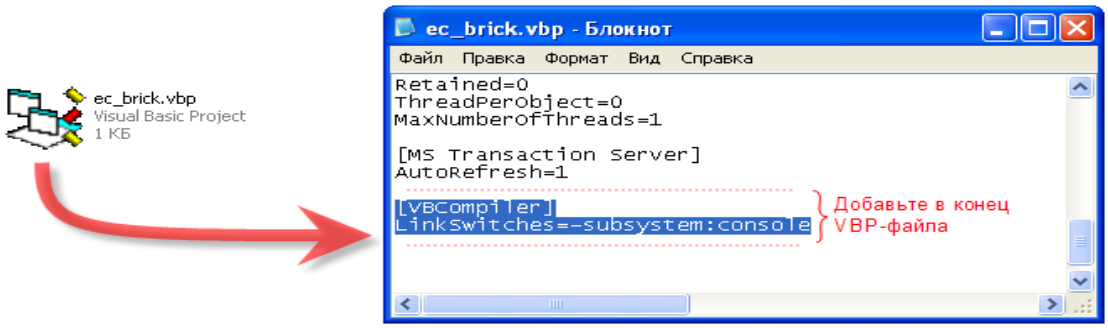
   Note that before you add it to look for the item should be **somewhere** in the **middle of the list.**

3. Making VB generate "Console» EXE-Schnick. When my plugin ACM-Tools will be made, this would require just-just install one checkbox in the properties of VB-project. But while I did this plugin, you will need to edit the VBP-file (project) by hand in Notepad and add the following

two lines:

```
[VBCompiler]
LinkSwitches=-subsystem:console
```

It will look something like this:



Then you can use the module to compile and complete console application on VB6!

The use is as follows:

- Once you call a function module from EasyconInitialize EasyConsole.
- Are you from anywhere in your project accesses objects **stdin, stdout or** stderr to control their console.
  For example the following code:

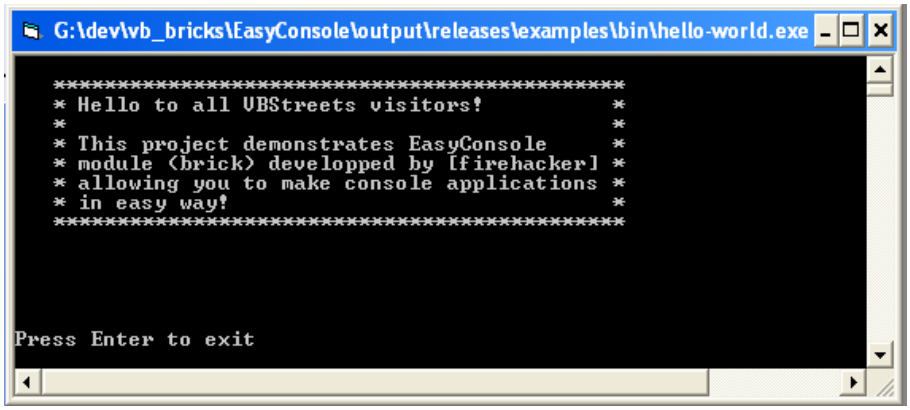
  ```
  stdout.WriteLine "Hello Wolrd! Long Live VBStreets!"
  ```

  Outputs to the console line «Hello Wolrd! Long Live VBStreets! ».

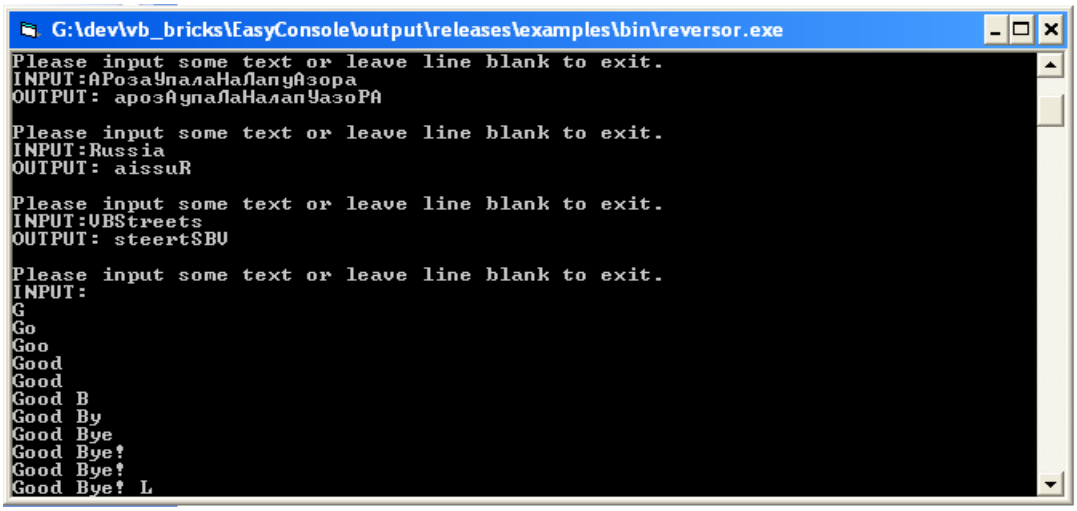
Deal with the bricks will help you examples:

The archive 4 examples:

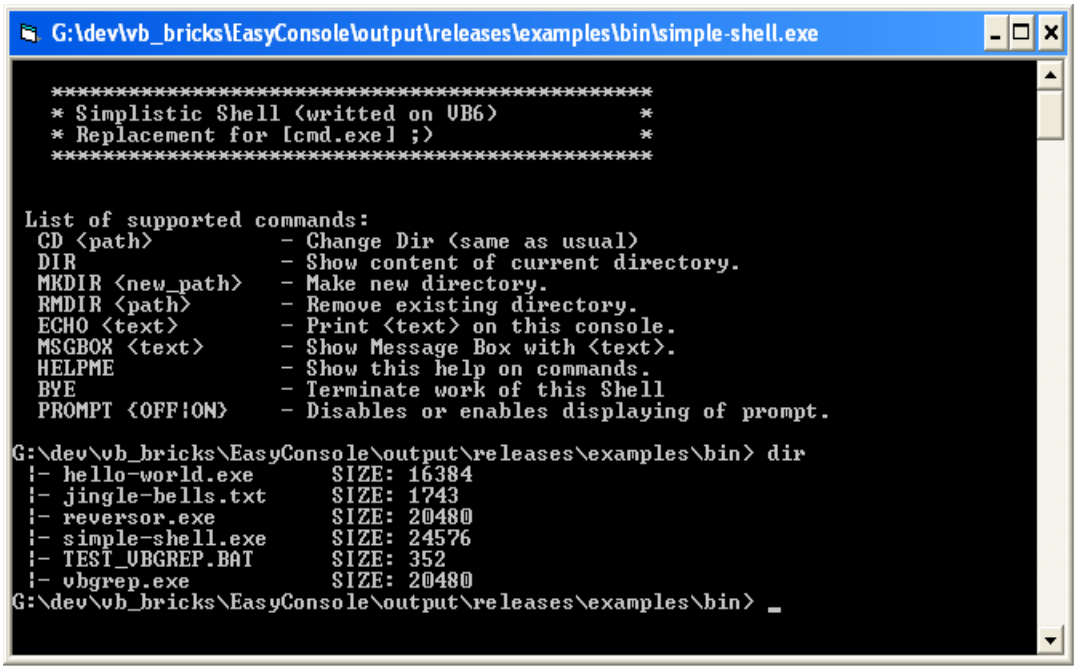
- **HelloWorld** - simply displays some welcome text in the console.



- **Reversor** - you enter the console a word or sentence, and the program displays it in the console otzerkalennom (strreverse) form.

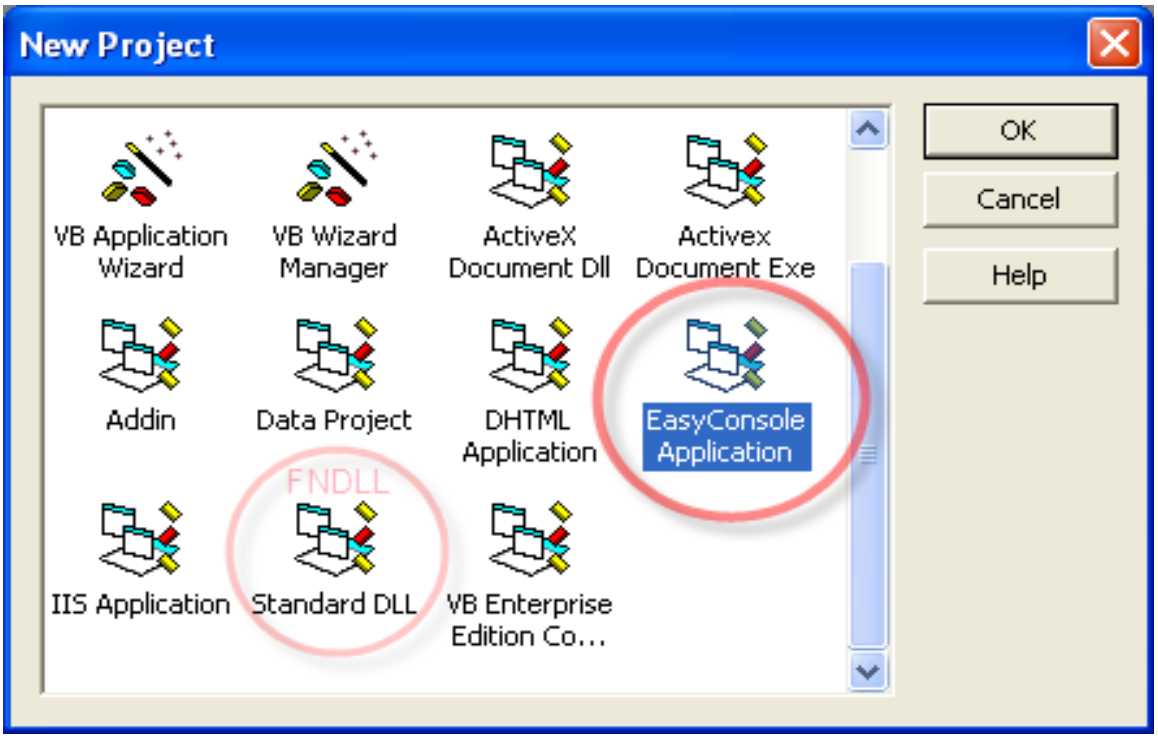


- **SimpleShell** - dull primitive command-line shell (like cmd.exe). It supports approximately 9 teams and launch external programs.



- **VbGrep** - a simplified version of the utility grep - the command line specified search words (separated by a space), the utility reads text from stdin-and to stdout and displays only those rows where the search text. At the same time, and displays the line number in which the search term is found. The example does not need to run the utility itself, and the batch file TEST_VBGREP.BAT, which feeds the utility text file with the text of the song *Jingle Bells*, as well as the search terms specified word «the» and «fun».

Template (Template)
Also, for your convenience, each time that you do not have to create a project, add a module to add the item in the Project → References and edit the project file in notepad, I **made** a template that will immediately create a project with the necessary screwing:



You may be familiar with the mechanism of templates when the template was added by FNDLL. To install the template, you need to download this file here:

And extract its contents into a folder with the project templates.

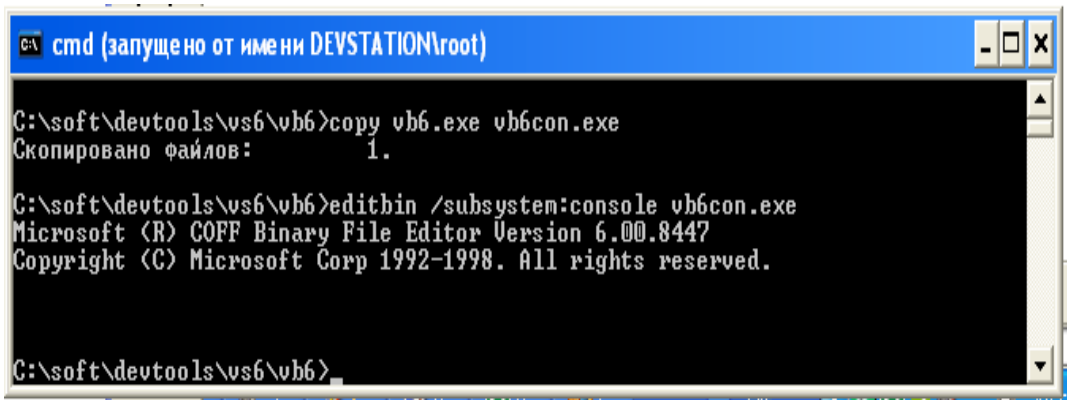
For example, if you have VB6 installed on a standard location:
c: \ program files \ Microsoft Visual Studio \ VB98 \
What you need to extract the files into this folder:
c: \ program files \ Microsoft Visual Studio \ VB98 \ Template \ Projects

Minuses
This solution has one drawback. Console applications can not be debugged. Rather, debug, of course you can, but attempt to write anything to stdout when working under debugging (or read anything from stdin-a) privedë error (which, however, can be treated).

If you are debugging is critical, I see two solutions to this problem:

1. Make a copy of the file and name it **VB6.EXE, say,** VB6CON.EXE. Then, using a utility **EDITBIN** edit this file, changing the value of the **subsystem** on the **console.** That is something

like this:



Then to work on your console project use it as an IDE VB6CON.EXE, and not just VB6.EXE. In this case, along with the environment will be created console window and your cantilever debugged program will operate fully, just as would behave in a compiled form.
The utility is part of the **editbin** and Visual Studio 6, and the Platform SDK, so it must be you, except that you're not one *of* those strange people who put only VB6, but not the whole VS6. But in any case, you can edit vb6con.exe any other PE-file editor.

2. It is another option. You can create a class or form **CDbgConsoleEmulator,** which write Implements ITextStream and implement this interface. The module also **EasyConsole** via I described elegant techniques determine under debugging if the project works, it is necessary to when working under debugging instead FSO-shnyh objects slip an instance of your class CDbgConsoleEmulator (which can be not just a class, but also a form included) . This instance will handle calls himself Read / ReadLine / Write / WriteLine and do something. For example, if this is the shape - a fully emulate the console interface.