Advanced BAT to EXE Converter is the best choice to compile your Batch Files to .EXE for distributing commercially. **Advanced BAT to EXE Converter not only encrypts your code,** but also creates an actual native Windows .EXE program. These .EXE do not require any pre-installed software to function and can be digitally signed.

Even the most complex batch files can be converted to executables with Intelli-Parse technology. This reduces development time by making sure your EXE works exactly as the original batch file does. You can embed your Company name and Copyright information to build your online reputation. Use the "Include Version Info" feature when compiling your project.

Learn the simple batch file language then Compile your script to **.EXE** It is the easiest and fastest way to develop Windows applications! There are many batch file examples included in the Advanced BAT to EXE Converter menu.

.BAT to .EXE Benefits:

- Users of your batch file scripts cannot view/change your code after it is encrypted by the compiler.
- Any actions performed by the script can be kept secret.
- Feature to run your scripts "Invisible" in the background without the batch file console window showing.
- One of the powerful new commands can Launch any Windows program completely Invisible.
- **Executables compiled with Advanced BAT to EXE Converter will run on all Windows versions from Windows98 to Win10 64bit.**

Check out the Frequently Asked Questions section.

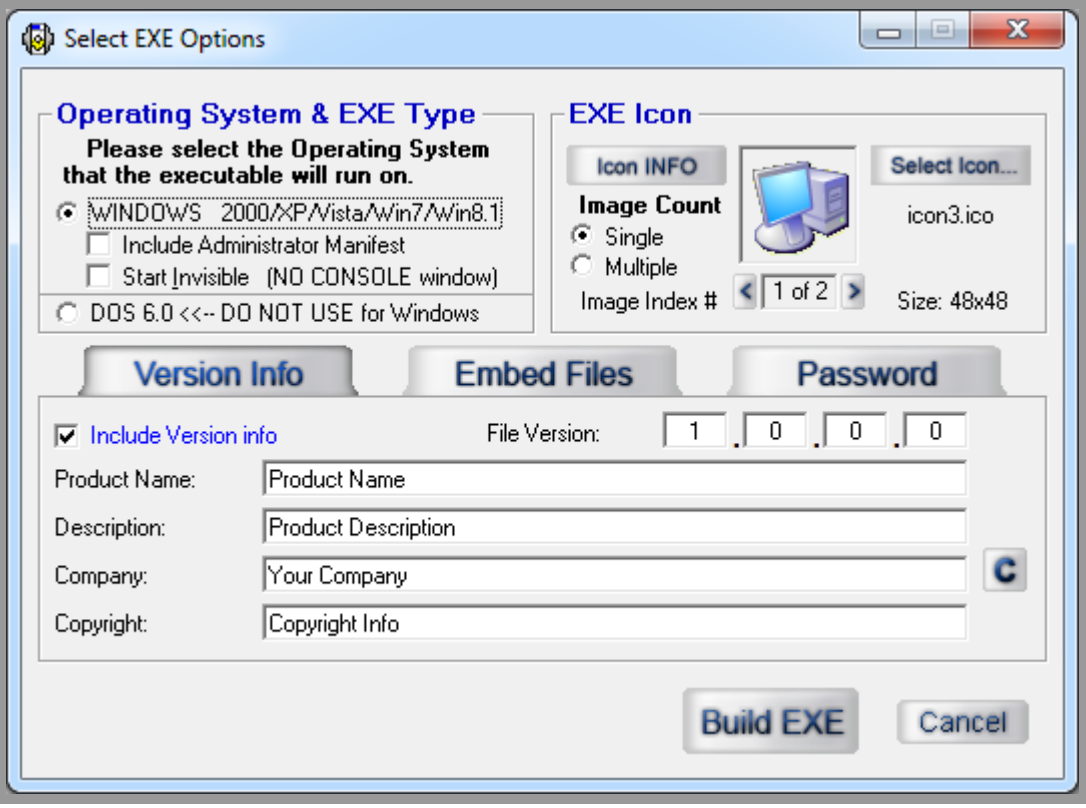
Advanced BAT to EXE Converter unique features:

- Allows Mouse Input for your batch files
- Built in editor.
- Select executable icon for a professional appearance.
- Encrypts batch file source to keep your code secret.
- You can set EXE product version and company info properties.

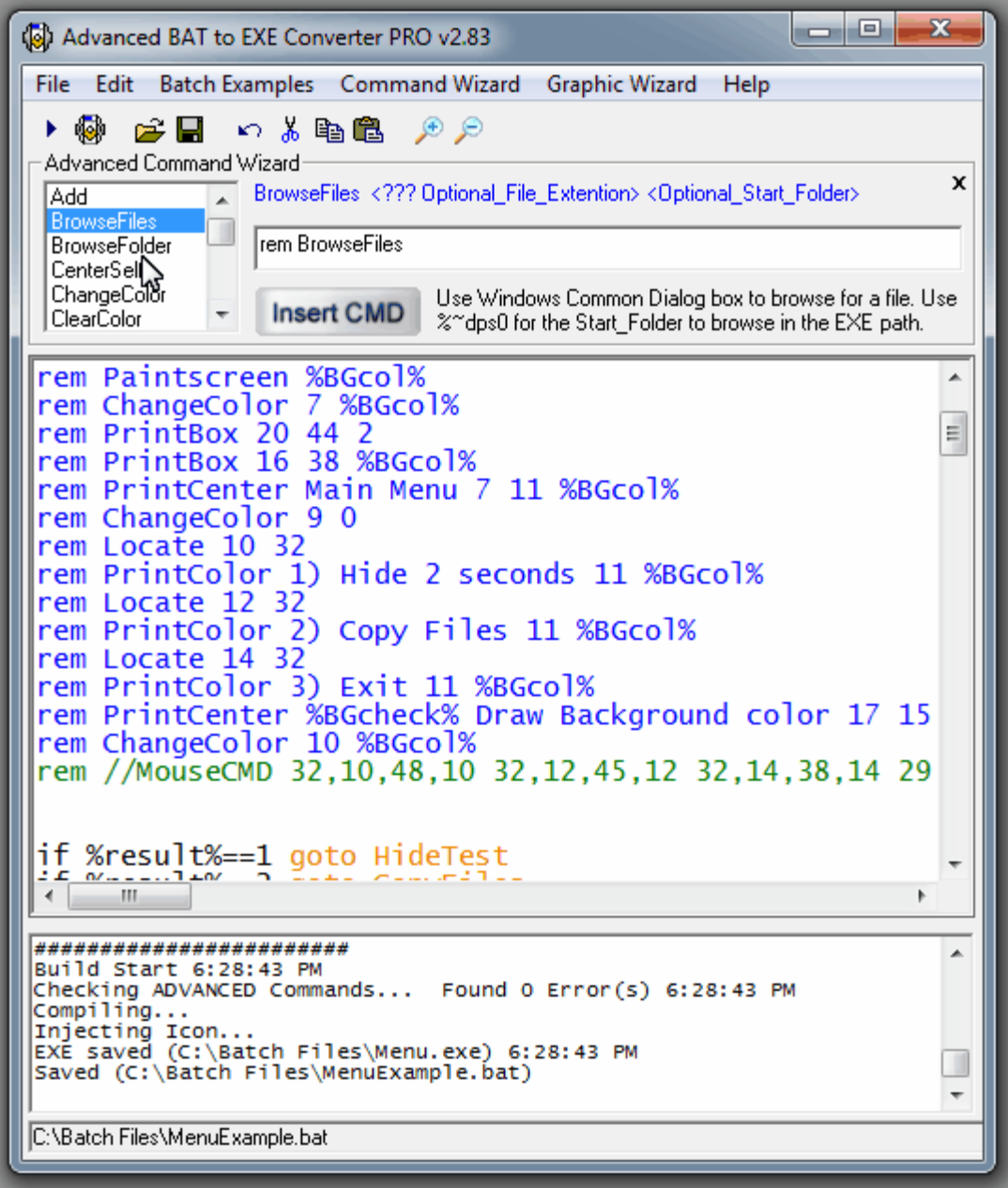
- Includes Advanced Extended Batch File Commands unavailable in normal batch files. There are many example batch files that utilize these extended commands.
- Invisible Mode will launch the batch file completely silent. Users will not know any program was launched.
- Embed and Encrypt additional dependency files (images, sounds, .exe)
- Useful for creating installation packages.
- Helps automate repetitive tasks.
- Generated .EXE are stand-alone files that do not require any .DLL or .NET framework to run.
- Works on all Windows 98/ME/2000/XP/2008/Vista/Win7/Win8.1/Win10 32+64bit compatible.
- MSDOS compile option.

Safe Site Reputation





All of the advanced commands are available
in both the **FREE** and **PRO** version.



Advanced BAT to EXE Converter has been updated often since 2003. We have spent thousands of hours developing the only batch file compiler that will work for all complex batch files.

No adware or bloatware in this free software download!

Advanced BAT to EXE Converter v3.00

DOWNLOAD

NEW v3.00 (May 02, 2016)
(Automatic access to 64bit commands from 32bit EXE for maximum compatibility across all PC)

OS: Windows

Size: 805 KB
License: Freeware
Updated Documentation below (scroll down)

Show us some love with 5 star ratings on these download sites.



[Advanced BAT to EXE Converter award on Informer]





[Download Advanced BAT to EXE Converter v2.92 from Softpedia]



[Advanced BAT to EXE Converter download on SnapFiles]



[Advanced BAT to EXE Converter download on FileHippo]





Advanced Extended Batch File Commands Index

All of these advanced commands are available in both the **FREE** and **PRO** version.

Please compile and run the many included example batch files that utilize these extended commands. Click a command for a better description and usage.

Add - Add two numbers.
BrowseFiles - Use Windows Common Dialog box to browse for a file.
BrowseFolder - Use Windows Common Dialog box to browse for a folder.
CenterSelf - Center the batch file window on the screen.
ChangeColor - Set the color will print to the screen.
ClearColor - Set the text color to the default colors.
CloseWindow - Close any application Window.
CursorHide - Hide the blinking text cursor.
CursorShow - Show the blinking text cursor.
Divide - Divide a number by another number.
FastCMD - Load an ASCII animation created by the Graphic Wizard.
GenRandom - Generate a random number.
GetInput - Input a string of text to a variable.
GetLength - Count the length of a string variable.
GetMasked - Input a string of text to a variable while hiding the input on the screen. Useful for passwords.
GetPercent - Find the percent of a number compared to the MAX value.
GreaterThan - Check to see if one number is greater than another number.
Hideself - Hide the batch file window during run time.
HideWindow - Hide any application using the Window Title.
LaunchSilent - Launch any application completely invisible.
LessThan - Check to see if one number is less than another number.
LimitDecimal - Limit a decimal number to a certain number of decimal places.
Locate - Position the text cursor at the specified coordinates on the screen. Useful for printing text anywhere.
LocateAt - Position the text cursor at the specified coordinates on the screen. Accepts relative (-2 +3) coordinates.
MakeInteger - Change a decimal number to an integer.
MakeLower - Change a string of text to lowercase.
MakeUpper - Change a string of text to UPPERCASE.
MouseCmd - Allows the use of the mouse within your batch file. Create clickable buttons.
Multiply - Multiply two numbers.
PaintBoxAt - Print a graphical box centered on the console screen. Two borders available.
PaintScreen - Change the background color of the batch file window.
PrintBox - Print a graphical box on the screen at the supplied coordinates.
PrintBoxAt - Print a graphical box on the screen at the supplied coordinates.
PrintCenter - Prints color text centered on the console screen.
PrintColor - Prints color text where the text cursor is. Does not print a RETURN afterwards as ECHO does.
PrintColorAt - Prints color text at the specified coordinates. Does not print a RETURN afterwards.
PrintReturn - Prints a RETURN on the screen. This will send the text cursor to the first column on the next row.
PrintSpace - Prints a SPACE on the screen where the text cursor is positioned.
ShadeBoxAt - Paints 1 of 4 different text patterns in the form of a box at the specified coordinates.
ShowSelf - Makes the batch file console window reappear during run time. Useful if compiled as Invisible.
ShowWindow - Show any hidden running application window by specifying the window tile.
Subtract - Subtract a number from another number.
Wait - The batch file script will wait the specified number of milliseconds before running the next command.

NOTE - The Advanced Functions do NOT work for DOS 6.0 compiled executables.

Extended Functions Usage

Tip #1 - You MUST use the prefix "rem " for the compiler to recognize the **Advanced Commands**. There must be only one space between REM and the Advanced Command or it will remain a **COMMENT**.

```
rem HideSelf
rem  HideSelf
```

Tip #2 - Advanced Commands with the **"At"** suffix (graphical commands) will hide the console cursor.
The cursor is hidden so you can string graphical commands together without the cursor blinking on the screen.
You may want to use the CursorShow command after you use these commands.

- **rem Add [Number1] [Number2]**
  Adds two numbers and returns the result in the variable %result%
- **rem BrowseFiles [??? Optional_File_Extention] [%~dps0 Optional_Start_Path]**
  Use Windows Common Dialog box to browse for a file. Use %~dps0 for the Start_Path to browse in the EXE path. Selected file is stored in variable %result%
- **rem BrowseFolder**
  Use a Windows Dialog to browse for a folder. Selected folder is stored in the variable %result%
- **rem CenterSelf**
  Center the console window on the screen
- **rem ChangeColor [Integer_FGcolor(0-15)] [Int_BGcolor(0-15)]**
  Changes the color text will be printed on the screen
- **rem ClearColor**
  Changes the color text will be printed on the screen back to default colors (FG=7 BG=0)
- **rem CloseWindow "Window_Title"**
  Close an application window. You must know the window title. Surround the Window_Title with quotes. Watch for double spaces.
- **rem CursorHide**
  Hides the blinking cursor on the console window
- **rem CursorShow**
  Shows the blinking cursor on the console window
- **rem Divide [Number1] [Number2]**
  Divides Number1 by Number2 and returns the result in the variable %result%
- **rem FastCMD [Slot(1-9)]**
  Embed Graphic Wizard File (Saved Slot 1-9)
- **rem GenRandom [MaxInteger]**
  Generates a random integer no larger than MaxInteger and returns the number in the variable %result%
- **rem GetInput**
  Lets the user type input and returns the result in the variable %result%
- **rem GetLength [Text]**
  Returns the length of text in the integer variable %result%
- **rem GetMasked**
  Lets the user type input (MASKED for passwords) and returns the result in the variable %result%
- **rem GetPercent [Number1] [Number2]**
  Finds the percent [Number1] out of [Number2] and returns the result in the variable %result%
- **rem GreaterThan [Number1] [Number2]**
  If Number1 >= Number2 (Greater than or equal to) returns 1 if true and a 0 if false in the variable %result%
- **rem Hideself**
  Allows the compiled batch file window to go invisible and run in the background
- **rem HideWindow "Window_Title"**
  Hide any window on the screen. You must know the window title. Surround the Window_Title with quotes. Watch for double spaces.
- **rem LaunchSilent [Text_Filename] [Options(or "")] [Int_WaitTillFinish(1or0)]**

Launches any executable completely silent. Use double quotes "" if no options. Wait(1=YES 0=NO)
Filename and path cannot contain spaces. Recommend Embedding the target EXE.
Use **%MYFILES%\filename.exe** to launch it.
If you must run an App with spaces in the path, use **CD** command to change the current directory then just enter the EXE filename without path.

- **rem LessThan [Number1] [Number2]**
  If Number1 <= Number2 (Less than or equal to) returns 1 if true and a 0 if false in the variable %result%
- **rem LimitDecimal [DecimalNumber] [Integer_X]**
  Limits the DecimalNumber to X number decimal places and returns the result to the variable %result%
- **rem Locate [Integer_Y(1-25)] [Int_X(1-80)]**
  Moves the cursor Y number of spaces down and X number of spaces right from the top left corner
- **rem LocateAt [Integer_Y(1-25)] [Int_X(1-80)]**
  Hides Cursor. Moves the cursor Y number of spaces down and X number of spaces right from the top left corner.
- **rem MakeInteger [DecimalNumber]**
  Makes the DecimalNumber an integer and returns the result in the variable %result%
- **rem MakeLower [Text]**
  Makes the text all lowercase and returns the text in the variable %result%
- **rem MakeUpper [Text]**
  Makes the text all UPPERCASE and returns the text in the variable %result%
- **rem MouseCmd [x1,y1,x2,y2]**
  Create a button/hotspot that can be clicked with the mouse. Separate multiple buttons with a space.
  Compile "rem MouseCMD GET" to get the coordinates of your buttons at run-time. They are stored in the clipboard.
  The variable %result% will return which button number was pressed. See example.
- **rem Multiply [Number1] [Number2]**
  Multiplies two numbers and returns the result in the variable %result%
- **rem PaintBoxAt [Integer_Y] [Int_X] [Int_Height] [Int_Width] [Int_BGcolor(0-15)]**
  Hides Cursor. Paints a box on the screen at the Y X coordinates with the specified dimensions and BGcolor
- **rem PaintScreen [Integer_BGcolor(0-15)]**
  Paints the screen the selected integer background color
- **rem PrintBox [Integer_Height] [Int_Width] [Int_BorderType(1or2)]**
  Prints a box centered on the screen with the specified border type
- **rem PrintBoxAt [Integer_Y] [Int_X] [Int_Height] [Int_Width] [Int_Border(1or2)]**
  Hides Cursor. Prints a box on the screen at the Y X coordinates with the specified dimensions and border type
- **rem PrintCenter [Text] [Integer_Y(1-25)] [Int_FGcolor(0-15)] [Int_BGcolor(0-15)]**
  Prints one centered line of text on screen with the specified colors in the row Y
- **rem PrintColor [Text] [Integer_FGcolor(0-15)] [Int_BGcolor(0-15)]**
  Prints text on the screen with the specified colors without printing a Return after. Use PrintReturn command if you must
- **rem PrintColorAt [Text] [Int_Y] [Int_X] [Int_FGcolor(0-15)] [Int_BGcolor(0-15)]**
  Hides Cursor. Prints colored text on the screen at the Y X coordinates without printing a Return. Use PrintReturn
- **rem PrintReturn**
  Moves cursor down a line
- **rem PrintSpace**
  Moves the cursor a space to the right
- **rem ShadeBoxAt [Int_Y] [Int_X] [Int_Height] [Int_Width] [Int_ShadeType(1-4)]**
  Hides Cursor. Shades a box on the screen at the Y X coordinates with the specified dimensions
- **rem ShowSelf**
  Allows the compiled batch file window to reappear after going invisible with HideSelf
- **rem ShowWindow "Window_Title"**
  Reveal a hidden window. You must know the window title. Surround the Window_Title with quotes. Watch for double spaces.
- **rem Subtract [Number1] [Number2]**

Subtracts Number2 from Number1 and returns the result in the variable %result%

- **rem Wait [Integer_Milliseconds]**
  Waits the specified number of milliseconds before continuing (1000 = 1 second)

Tips for using Advanced BAT to EXE Converter

- Open and look through the example batch files included in the compiler menu
- While editing your batch files you will notice the code changing colors when using the **Advanced Commands**:
  GREEN for REM comments (REM still works for non Advanced Command statements)
  PURPLE (warning) for recognized Advanced Commands missing options
  BLUE (ok) for rem Advanced Commands with the correct number of options
  RED (error) Advanced Command with possible typo
  ORANGE for GOTO and :Labels
- Please note that the compiler color codes the script statements based on the number of options entered after an extended command. The compiler does NOT check for the correct data type.
- Save your batch file after compiling it to save the Project Options including EXE path, icon, version information.
  This data will be loaded the next time you open the batch file with the compiler.
- To add a comment in the batch file, use a REM statement
- Be sure not to use any commands that may require user input while the compiled batch file is hiding with the HideSelf command
- PrintColor does not print a "return" after it prints to the screen. This is so you can print more than one color on a line without having to use Locate. Simply use PrintReturn to start at the next line.
- Use ChangeColor when using PaintScreen to change the text backround color to match the current background color.
- To Embed files to the compiled EXE, click the plus sign "+" on the Select EXE Options window (Embed Files tab)
- Use the **%MYFILES%** variable within your script to access these files as if it was a folder.
- Use **%0** to get the EXE filename of the compiled batch file EXE at runtime.
- Use **%~dps0** to get the folder path of the compiled batch file EXE at runtime.
- Batch file windows are 80 spaces in width and 25 spaces in height.
- To close the window when a batch file finishes executing, simply put a CLS (clear screen) statement at the end of the batch file.
- If the console window closes before you can see any text on the screen, put a PAUSE (press any key to continue) statement at the end of the batch file.
- Make sure English is selected on your Language bar.

Command Line Compile Usage:

Copy your **"ab2econv283pro"** installation folder out of %program files%
to a location where you have write permissions first.

```
aB2Econv.exe "c:\Batch Files\input.BAT" "c:\Batch Files\output.EXE"
```

It is recommended that you use quotes and full path filenames.

You can right click a batch file in Windows Explorer and use the context menu
to automatically compile a batch file as well.

FAQs (Frequently Asked Questions)

- **Why should I use Advanced BAT to EXE Converter?**

  Advanced BAT to EXE Converter is the first and only compiler that will work for all complex batch files without the need to make specific changes to your script based on which compiler you are using with **Intelli-Parse technology**. Less hassle=less development time.

  You can encrypt your script to hide the code from end users. This allows you to share scripts with passwords for FTP or networking functions. You can also embed Icons and other binary files that your script needs to function and turn them all into one stand-alone EXE file to share. You also have the option of password protecting your EXE so it will not run without the correct password.

- **What is Intelli-Parse?**

  Intelli-Parse technology is unique to Advanced BAT to EXE Converter. This feature scans your batch file script word for word and will make any necessary changes to your code in memory so it runs exactly how your unmodified batch file does before compiling. Other solutions would require modification of some commands within your batch file so they operate correctly when converted to EXE.

- **What else makes Advanced BAT to EXE Converter better than other BAT to EXE compilers?**

  There are over 40 Extended commands unique to Advanced BAT to EXE Converter that makes your scripts much more powerful including hiding any windows application dialog and using mouse input within your batch file script.

- **Will my batch file need any modifications before compiling?**

  Advanced BAT to EXE Converter is the only compiler that will compile your script and not need any modifications for all complex tasks.

- **How can I modify my batch script so it does not require an Admin Manifest?**

  Most batch files only require Admin privileges because it uses folders that are off limits to non-admin users. One excellent feature of Advanced BAT to EXE Converter is the %MYFILES% variable. Use can use the %MYFILES% variable as the working folder for your batch file since you always have write access to this directory and files even without Admin rights.

- **If I buy the PRO version, do I have to pay for future updates?**

  No. Free updates for one year. Discount after. Free technical support with a fast response time for the free and PRO version.

- **How do I compile as Invisible or include an Icon and Version properties when compiling a batch file with the Windows Context menu or command line?**

  First, compile the batch file using Advanced BAT to EXE Converter with all of the project options you would like. Next, save the batch file. All of your project settings including Icon, Embedded files, Company Name, Version and Invisible properties are saved in the header of the batch file. These settings will be loaded and used when you compile from the command line or use the Windows Explorer context menu compile option.

  When compiling from the Windows context menu, the output EXE file will always be named after the batch file ignoring the (BFCPE)EXE filename variable in the header.

- **How do I use Embedded files in my batch file script?**

  First click the "Build EXE" option in the File menu or on the toolbar. Next, select your EXE filename. Embed any of the files that your script needs to use upon execution in to your EXE using the "Embed Files" tab on the "Select EXE Options" dialog.
  Use the %MYFILES% variable within your script to access these files as if it was a folder. This will create a single stand-alone EXE file instead of sending your project in multiple files.

- **The LaunchSilent extended command does not seem to work for the program that I am launching?**

  Some windows applications will launch a second dialog that will not remain invisible. In this case, use the HideWindow extended command. Make sure you use the "rem Wait 250" extended command between launching the program and hiding it. This will allow the program time to load before attempting to hide it. You may want to launch the application minimized in this situation. Here is a quick example:

  ```
  start /MIN notepad.exe
  rem Wait 250
  rem HideWindow "Untitled - Notepad"
  ```

- **My Icon is not showing after compiling the batch file script?**

  Simply copy your compiled EXE to another folder and you will see the icon appear. There is a known issue with the Windows7 fast-cache feature that in rare situations could prevent the new icon from showing at first.

- **Advancd BAT to EXE Converter started as Batch File Compiler Professional Edition in 2003**

  Batch File Compiler Professional Edition was created in 2003 as a trial based software. Since then, many new features have been added and it was changed to a limitation free on one computer model and renamed.

Graphic Wizard

Steps to get started:

- Choose the FG (foreground) and BG (background) colors you wish to use with the color strip (bottom right)
- Choose type of tool in preview box (top right)
- Click where you would like the top-left corner of the tool to start. Then click the bottom-right corner in the work surface. This should draw a box based on which tool was selected in the previous step.
- Click the "Wait30ms" button when you wish to start a new "frame" if you are creating an animation. This will also refresh the GE Wiz window.
- Click the "Copy Code to Clipboard" command button. Paste in Advanced BAT to EXE Converter, then BUILD EXE to test.
- Examples can be found in the LOAD button