

Working with



in Notepad++

Version 1.2

by Mikael Norrgård (witchmaster)

Introduction

I really like Blitz Basic 2 but the editor (SuperTed) is far from ideal to work in. Therefore I created a user defined language for Notepad++ (+other stuff) to ease the process of working with Blitz Basic 2. This is what you can expect:

- Syntax highlighting
- Auto-completion of most keywords (also asm keywords)
- Foldable region support
- Function / Statement list
- Automation (F5 in Notepad to run and compile code in WinUAE)
- Keyword help lookup

A screenshot of the Notepad++ application window. The main editor area shows Blitz Basic 2 code with syntax highlighting. The code includes a region for constants, bit definitions for CD status, a PrintAt function, and initialization commands. The function list on the right shows the PrintAt function. The code is as follows:

```
7 ; region @Constants
8 ; CDStatus bit definitions
9 #CDSTSB_CLOSED = 0 ; Drive door is closed
10 #CDSTSB_DISK = 1 ; A disk has been detected
11 #CDSTSB_SPIN = 2 ; Disk is spinning (motor is on)
12 #CDSTSB_TOC = 3 ; Table of contents read. Disk is valid.
13 #CDSTSB_CDROM = 4 ; Track 1 contains CD-ROM data
14 #CDSTSB_PLAYING = 5 ; Audio is playing
15 #CDSTSB_PAUSED = 6 ; Pause mode (pauses on play command)
16 #CDSTSB_SEARCH = 7 ; Search mode (Fast Forward/Fast Reverse)
17 #CDSTSB_DIRECTION = 8 ; Search direction (0 = Forward, 1 = Reverse)
18 ; endRegion
19
20 ; Print at location with colour
21 Statement PrintAt{x.w, y.w, col.b, text$}
22   Locate x,y : Colour col : Print text$
23 End Statement
24
25 ; Initialize
26 BLITZ ; Go into blitz mode
27 Slice 0,44,320,200,$fff8,4,8,16,320,320 ; Create a slice
```

There are a few limitations to this system, these are:

- Functions and Statements must be declared at the start of rows to show up in function list (no whitespace before Function/Statement keywords).
- Autocompletion is case sensitive, so you need to start writing the keywords in the correct case for them to show up.

Getting the necessary files

First you need to download and install WinUAE, Blitz Basic 2 and Notepad++ if you don't already have them installed. Here are the links for the user defined language files and tools:

<http://www.gamephase.net/files/linked/BB2-NPP-v1.0.zip>

<http://www.gamephase.net/files/linked/npp-bb2-tools-v10.zip>

Setting up syntax highlighting, regions, keywords and function list

1. Open Notepad++ and go to Language / Define your language...
2. Click Import... and select the file "UDL-BlitzBasic2.xml".
3. Close the User defined languages window and close Notepad++.
4. Copy the file "blitzbasic2.xml" to [Notepad++ installation directory]/plugins/APIs
5. Go to %AppData%\Notepad++ and open functionList.xml. Add the following association (under the other ones):

```
<association userDefinedLangName="BlitzBasic2" id="blitz_basic_2"/>
```

and add the following parser under the other parsers. Note that the entire <function ... > tag must be on the same line:

```
<parser id="blitz_basic_2" displayName="Blitz Basic 2">  
  <function mainExpr="^[ \t ]*(Function|Statement|\.)[\s\t]*([a-zA-Z0-9]+\$*)[\s\t]*({|:|\s|$)" displayMode="$functionName">  
    <functionName>  
      <nameExpr expr="(\.|[\t ]+)[a-zA-Z0-9]+\$*" />  
    </functionName>  
  </function>  
</parser>
```

6. All done, save Blitz Basic 2 source files with the extension .bb2 for the syntax highlighting and auto-completion to work. The function list window can be toggled from View / Function List in Notepad++.

Setting up automation

We will set up the automation so that we will work with source files using Windows encoded linebreaks, naming these files with a `_win` suffix. The source files will be stored in a virtual harddrive set up in WinUAE so you they can be shared with Windows. The automated process will make copies of the source files (automatically converting the linebreaks to Unix style and converting the include files to the tokenized format used by BB2). The main source file will then be loaded and compiled in BB2 running in WinUAE.

Blitz basic 2 has an AREXX port so it's possible to automate it using AREXX. You can take the following steps to set things up.

1. Ensure that AREXX is started with WorkBench. The line:
`SYS:System/RexxMast >NIL:`
should exist in either `S/startup-sequence` or `S/user-startup`. If not, add it to the end of `user-startup`.
2. In WinUae set serial port to `TCP://0.0.0.0:1234` and select "Direct" below the drop down box (Settings / Host / IO ports). Save the WinUAE configuration and quit WinUAE, add the following to the configuration file manually (under the other lines concerning serial): `serial_translate=crlf_cr`
3. Create the file `DEVS:MountList` on the virtual harddrive with the following content (if it does not already exist)

```
AUX:  
Handler = L:Aux-Handler  
Stacksize = 1000  
Priority = 5
```

4. Add the following commands to the end of `S:user-startup`

```
mount aux:  
newshell aux:
```

5. Create a file named blitzbasic2.rexx in S: with the following content:

```
/* Blitz automation */

/* Get all command line parameters */
PARSE ARG commandline
count = 0
DO WHILE LENGTH( commandline ) > 0
    commandline = STRIP( commandline, 'B' )
    count = count + 1
    IF LEFT( commandline, 1 ) = '"' THEN DO
        PARSE VAR commandline '"'parameter.count'"' commandline
    END
    ELSE DO
        PARSE VAR commandline parameter.count commandline
    END
END
parameter.0 = count
DROP commandline

/* Address Ted and show it */
address TED_REXX1
SHOWSCREEN
WINDOWTOFRONT
ACTIVATE

/* Load and save all the include files to make them tokenized
(command line parameters 2+) */
DO counter = 2 TO parameter.0
    LOAD parameter.counter
    SAVE
END

/* Load the main project file (first command line parameter) */
LOAD parameter.1

/* Compile the project */
COMPILE
```

6. Download the command line tools I made specifically for this purpose (but can be used for other purposes) and copy them to the Windows directory. <http://gamephase.net/files/linked/npp-bb2-tools-v10.zip>. The tools are:

<code>ConvertEOL</code>	Converts EOL characters in text files between different formats (LF/CR/CRLF)
<code>WinUAEArexx</code>	Runs an Arexx script in WinUAE over Telnet
<code>BringToFront</code>	Brings an application with specific window title to front

7. Create a batch file (.bat) in the same folder as your source file(s) with the content below (change paths and filenames according to your environment, the example is for a project named "HelloWorld" with one include file).

```
:: Set project folder
cd "C:\Dropbox\Programming\BlitzBasic2\HelloWorld"
:: Copy all _win source files and at the same time convert end of
line coding to unix style
ConvertEOL unix HelloWorld_include_win.bb2 HelloWorld_include.bb2
ConvertEOL unix HelloWorld_win.bb2 HelloWorld.bb2
:: Show the WinUAE window
BringToFront "WinUAE"
:: Run the ARExx script
:: (First bb2 file is main source file, the following are included
:: files. BLITZ: in this example is a virtual harddrive set up in
:: WinUAE set to a folder where I store my BB2 projects.)
WinUAEArexx blitzbasic2 1000 "BLITZ:HelloWorld/HelloWorld.bb2"
"BLITZ:HelloWorld/HelloWorld_include.bb2"
```

- The WinUAEArexx command should be supplied with the path to the source file as seen by the emulated Amiga. 1000 after the script name is how long to wait before closing the telnet connection (try a number >1000 if you have problems), if closing instantly it seems that the command won't be transferred properly.

- BringToFront needs the exact window title of the running WinUAE instance. If WinUAE is started from a configuration, the Window title will probably be simply "WinUAE".

8. If you want to use the above HelloWorld example, create two files, one named HelloWorld_win.bb2 and one named HelloWorld_include_win.bb2. The content for the files.

HelloWorld_win.bb2:

```
; Main source file
INCLUDE "HelloWorld_include.bb2"
NPrint "Hello World"
```

HelloWorld_include_win.bb2:

```
; Dummy include file for Hello World project
#ANSWER_TO_THE_ULTIMATE_QUESTION = 42
```

Using the automation

1. Start WinUAE, load your configuration and start the emulation, then run Blitz Basic 2 (SuperTed).
2. Edit and save the Blitz Basic 2 source file(s) in Notepad++. Name them with a suffix, like _win (example: HelloWorld_win.bb2). Use Windows style linebreaks in your source files (CRLF).
3. To compile and run, press F5 in Notepad++ and select the batch file you created in step 6 above. Blitz Basic 2 should reload and compile the source file and WinUAE should be brought to the front.

Creating regions in your code

Regions are simply created like the example below:

```
; region @YourRegionName
; your code goes here
; endRegion
```

Edit Compiler Options

The compiler options in Blitz Basic 2 are saved per project and are stored with the project in a .extra file. When using Notepad++ to edit your code, this .extra file will not be saved automatically (since you don't save the project in the BB2 SuperTed editor). This is how you can edit and save these options manually:

1. Open the project file (the one converted to unix eol) in blitz basic 2.
2. Change the compiler settings to what you want.
3. Go to Project / Save to save the project along with the compiler options. This will replace the ascii version of the project file with a tokenized file. It will be overwritten with an ascii version next time you run the .bat file, this doesn't matter, the compiler options will still be loaded.

Setting up easy help for BB2 keywords

Here we will set up a help system that will open the documentation for the keyword at the current cursor position by simply pressing F1 (or another key). The documentation will be shown in a web browser (using the Amiga Developer Site). Not all BB2 commands are documented on that site but it's the most suited one I have found so far.

1. First Install the NPPExec plugin from the menu Plugins / Plugin Manager / Show Plugin Manager.
2. Once installed, go to Plugins / NPPExec / Execute... and write the following lines in the Commands window (edit the path to the web browser if needed). Note that everything after the line NPP_CONSOLE 0 should be on the same line.

```
NPP_CONSOLE 0
"C:\Program Files (x86)\Google\Chrome\Application\chrome.exe"
"http://amiga.sourceforge.net/amigadevhelp/phpwebdev.php?keyword=$
(CURRENT_WORD)&funcgroup=BlitzBasic&action=Search"
```

Click Save... and save the script with the name "Blitz Basic 2 Keyword Search".

3. Go to Plugins / NPPExec / Advanced Options... and add the script you created as a menu option. Under Menu Item, write "Blitz Basic 2 Keyword Search" as item name and select the script with the same name under associated script. Click Add/Modify, click Ok and restart Notepad++.
4. Now create a keyboard shortcut to the script. Go to Settings / Shortcut Mapper... and select the Plugin commands tab. Here you should find the item "Blitz Basic 2 Keyword Search". Select it and click modify and create a shortcut to it. The F1 key in Notepad++ will normally just show the About window so it can safely be used. You can get a message saying that a conflict was found. The conflicted items will be highlighted red so they are easy to find. Remove the F1 shortcut for the About window to clear the conflict.
5. You should now be set, pressing F1 (or another shortcut you created) should now open a browser and show the documentation for the keyword at the cursor position, if the keyword can be found of course.

Conclusions

This system should make the process of making games etc. In Blitz Basic 2 easier. If you have any problems or suggestions, please post them to the thread about this system on English Amiga Board: <http://eab.abime.net/showthread.php?t=86316>