GC Studio 1.0 is now officially released

# Important:

The first stable release of GC Studio IDE 1.0 is released.

The IDE, based on VS code it is an open-source IDE with a specific extension/ for the Great Cow BASIC; PIC, AVR and LGT 8-bit range of microcontrollers are supported on the open-source compiler GCBASIC. Other extension include integrated Git and SVN.

Great Cow BASIC development started in 2013 and now GCBASIC supports over 1300 microcontrollers. GC Studio gives a modern and user-friendly user interface, improved help and compilation within many more features than the previous editor-based IDEs.

GC Studio was initially released as a BETA in March 2022 with milestone support, the improved user interface with features such as Intellisense, integrated Help, auto-completion of command, variables and methods. After 7 months of improvement with the help of members of the Great Cow BASIC community GC Studio IDE 1.0 is ready for prime time and it is now the default download for the toolchain. The download page is [**here**](https://www.gcbasic.com/reps/stagebuild/setup/GCstudioSetup.exe)**.**

The GC Studio 1.0 supports Windows 32-bit and 64-bit

The roadmap now includes a new integrated terminal and data acquisition, GC Studio for Linux X86-64, and macOS 10.14 “Mojave” or newer, Raspberry Pi integrated communication and programming on Basic and Python – the journey will keep going.

**Features**

The user interface has quick access to your program folders and files, auto-complete for new programs, snippets and direct access to the GC Code libraries will speed up your code writing..

You can even code programs for Windows and Linux on GC Studio, also you can access the dev environment with the source code of the compiler to modify and build it.

Compilation and programming are now controlled via the session manager – an enhancement that improved the user experience over the previous IDE.

A new installation takes a little while but with the automated update feature – the updates to the complete toolchain will ensure that GC Studio remains up to date.

**Previous BETA/Release Candidates of GC Studio**

This is a NEW release. It is important to Uninstall your previous GC Studio BETA installation and make a fresh install. The previous versions of GC Studio will not update to 1.0

**GC Studio**

GC Studio is the application that manages the Great Cow BASIC toolchain environment. It can maintain a consistent installation via channels. The channels are selectable by the user.

GC Studio is also the application launcher. The user can select difference code editors

1. GC Code – see below
2. Great Cow Graphical BASIC
3. SynWrite – the previous code editor
4. Geany

GC Studio was developed by Angel and is published on the Great Cow BASIC Organisations GitHub.

**GC Code**

GC Code is the code editor that supports the Great Cow BASIC specific extension developed by Angel and Evan. The Great Cow BASIC extension contains all the functionality/knowledge specific to coding in Great Cow BASIC.

GC Code.exe (the executable) is built on an open-source project called Code-Open Source Software (Code-OSS). Code-OSS is the core layer of VS Code. It is available on GitHub under the standard MIT License.

GC Code runs on macOS, Linux, and Windows.

GC Code is free for both private or commercial use.

GC Code telemetry reporting can be disabled to meet specific user privacy concerns.

/

**Questions and Answers**

Will Sourceforge will remain as the Open Source Repository for Great Cow BASIC compiler code? yes

Can users continue to use legacy installers? yes –they will be on SourceForge. The Windows distribution that was published in Jan 2022 is now the legacy installer.

Can users can use GC STUDIO to create local configurations/installations? yes

Can users can fork the project to create other distributions? yes.

Can users can compile a local installation using the Open Source Sources published on SourceForge Forum? yes

.

How will this be governed going forward? A Great Cow BASIC Open Source Charter will be published in early December. This Charter will be open for revision for 28 days. After 28 days the Charter will be the Design Authority responsibilities, working practices going forward. The Charter will address 'how to request a change to the Charter'.

Has GC STUDIO met the functional criteria described in the JANUARY 2022 forum post ? Yes.

Has GC STUDIO met the non-functional criteria described in the MARCH 2022 forum post ? Yes.

Will the Windows patch zip be published ? No. GC STUDIO functionally replaces the ZIP.

What happens to the Linux distribution? It will continue as-is.

What happens to the MAC OS distribution ? It is no longer supported and is now deprecated as a distribution.

What is the scope of the GC Studio in terms of the tool chain ? The software defined in the Great Cow BASIC Bill of Materials is the scope of Great Cow BASIC solution.

**Engineering to a new IDE (from the Forum post March 2022)**

The section answers the question ‘how to engineer a move from SynWrite to an alternative IDE would be a similar program of change when Trev moved the editorial department from one software tool to another but we do not have the executive sponsorship to drive the change.

To move across from the existing SynWrite to any new IDE would be relatively simple but time consuming. Here is a high level flight plan.

1. Prototype the IDE using the target IDE, publish to trial group of motivated testers and revise the prototype IDE based on learning.

2. Create release packages specifically for the target test platforms Windows, Linux etc– test and revise the release packages. Decide whether to proceed to the next step.

3. Create the Training materials and update Help, Helpers etc.

3. Then, release the package(s) alongside the existing SynWrite package for one release. Learn from release feedback, revising and updating the package(s) to address feedback. Decide whether to proceed to the next step.

4. Then, on the next release do not release SynWrite as the IDE but only release the new IDE. - Syn@Write would still be available as a separate download but is no longer recommended for future use

Timeline would look something like this. If these dates do not look aggressive.. They are aggressive from my experience of the community effort.

1. Prototype new IDE in 1Q2020

2. Create release package(s) – so an IDE editor that is ‘lite’ in 2Q2020 (not the mega downloads that I am aware of when I reviewed the IDEs)

3. Dual release of Great Cow BASIC for SynWrite and new IDE in 3Q2020

4. Release of Great Cow BASIC with new IDE only in 4Q2020

This is based on the next release of Great Cow BASIC - which is near completion. So, this would aim for the dual release in 3Q2020 with full replacement of SynWrite by 4Q2020.

Key to the success of this activity is someone to lead. I will call this role the Design Authority. A Design Authority has the final say on what actually happens. So, in reality step zero of this activity is someone to take the role of the Design Authority. The role is crucial. No Design Authority this will simply not happen.

Design Authority will own the solution.

The Design Authority that then leads the end to end testing. Do not underestimate the hours to maintain the IDE as the Design Authority. In the past. Frank was the initial Design Authority for SynWrite but as time progress I took over the role and Frank became a Technical Authority responsible for the code base in terms of functionality and quality. As Design Authority I am responsible to ensure changes are reflected in the end to end Great Cow BASIC tool chain.

A Technical Authority will also be needed. Someone who knows the IDE inside out. Someone needs to able to answer the very detailed questions. With SynWrite - Frank knew SynWrite, he integrated the Help, the Great Cow BASIC IDE Helpers and then others added capabilities under Franks guidance. I also ‘hacked’ the SynWrite exe to remove the horrid bugs (yep… in a hex editor).

I would be very happy to work with anyone to make this transition happen. The current build process can be replaced by any tooling but the key is the quality of the outcome. I care about the quality and the consistency of Great Cow BASIC.

But, I will not be the Design Authority for the replacement IDE. I will continue as the Design Authority for the Great Cow BASIC tool chain, and, Hugh would remain Design Authority for the Core Compiler, Program Editor and Great Cow Graphical BASIC.

So, to make this happen.... Someone needs to take the role of Design Authority to lead the program of change.

- The community will decide is .emacs works as the IDE. And, in my experience the community will provide honest and frank feedback.

- Any IDE can work with the right configuration and with the right approach to the change program then the goal of a cross-platform IDE will become a reality.