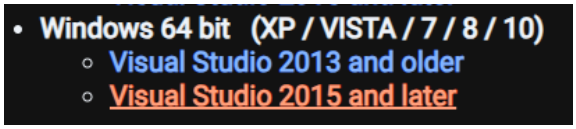
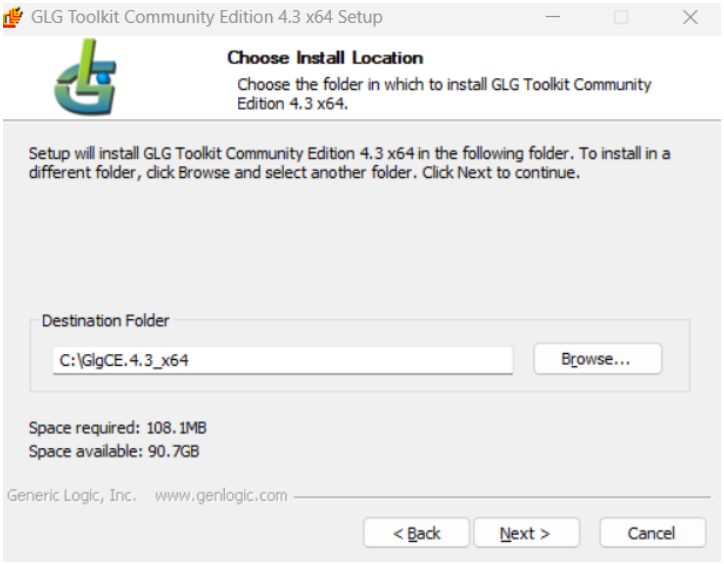
Documentation for Using Glg Toolkiit

➢ Instruction for Glg Toolkit download:

1. Open https://www.genlogic.com/select_platform.html to download the latest community edition version for Glg toolkit and download as per your platform requirements. Such as:



2. Go through the installation process after downloading **SetupGLG-CE-4-3-x64-vs2015.exe**
   - In the Destination Folder location change the setup path to: **C:\GlgCE.4.3_x64**



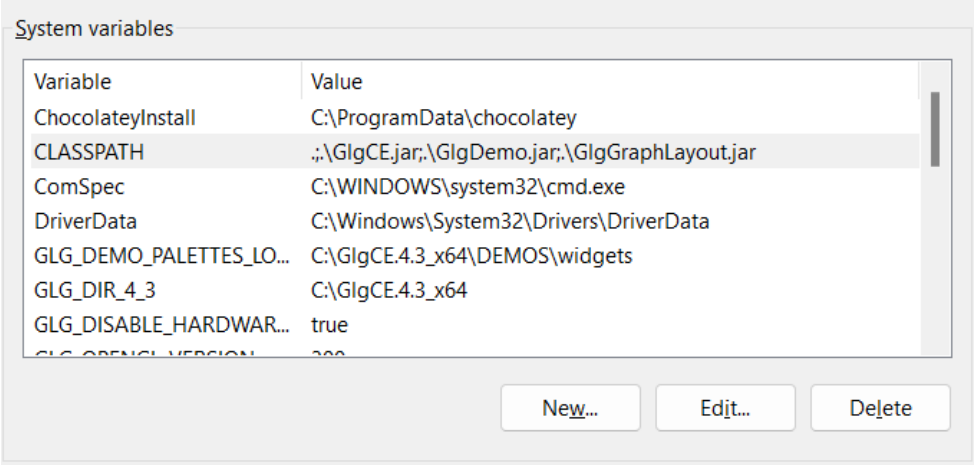
Note: *Donot install the GlG toolkit inside Program Files you can set the path to any other file.*

   - Setup the next setting to **OpenGL** and **SoftwareOpenGL** and finish the installation.

➢ Instruction for setting the Classpath for working on Glg toolkit on Visual Studio Code:

1. In your Pc go to **Settings>System>About>Advanced System Settings(under device specifications)> Environment Variables> System Variables.**

2. In System Variables select new and create a new variable with: Variable Name- **CLASSPATH** and Variable Value as- " **.;.\GlgCE.jar;.\GlgDemo.jar;.\GlgGraphLayout.jar**" omit("").



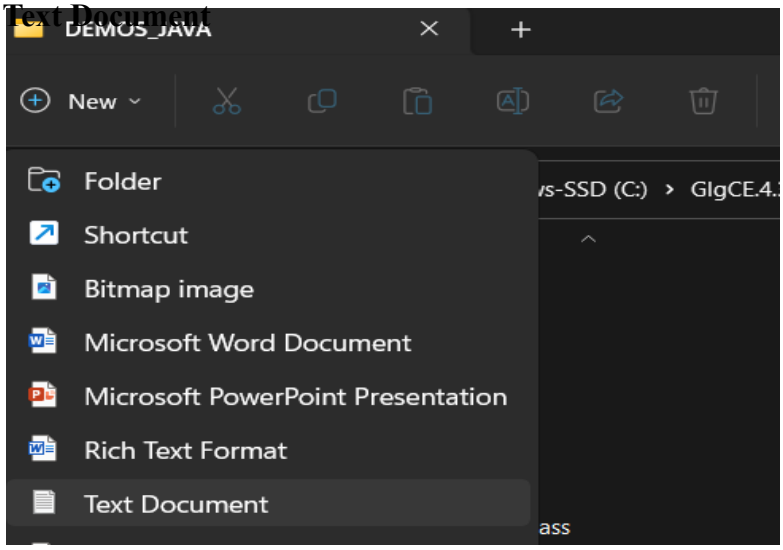
➢ Instructions for Opening Glg Demo Java programs in VS Code:

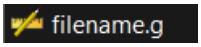
1. If you have installed the Glg toolkit in **C:** with name such as **GlgCE.xxx** then open your VS Code IDE and Select **File>Open Folder** and setup the path to **C:\GlgCE.4.3_x64\DEMOS_JAVA** . This adds all the Glg java demo programs to the IDE.

2. Select any program such as **GlgAircombatDemo.java** and select Run Code to view the graphics.

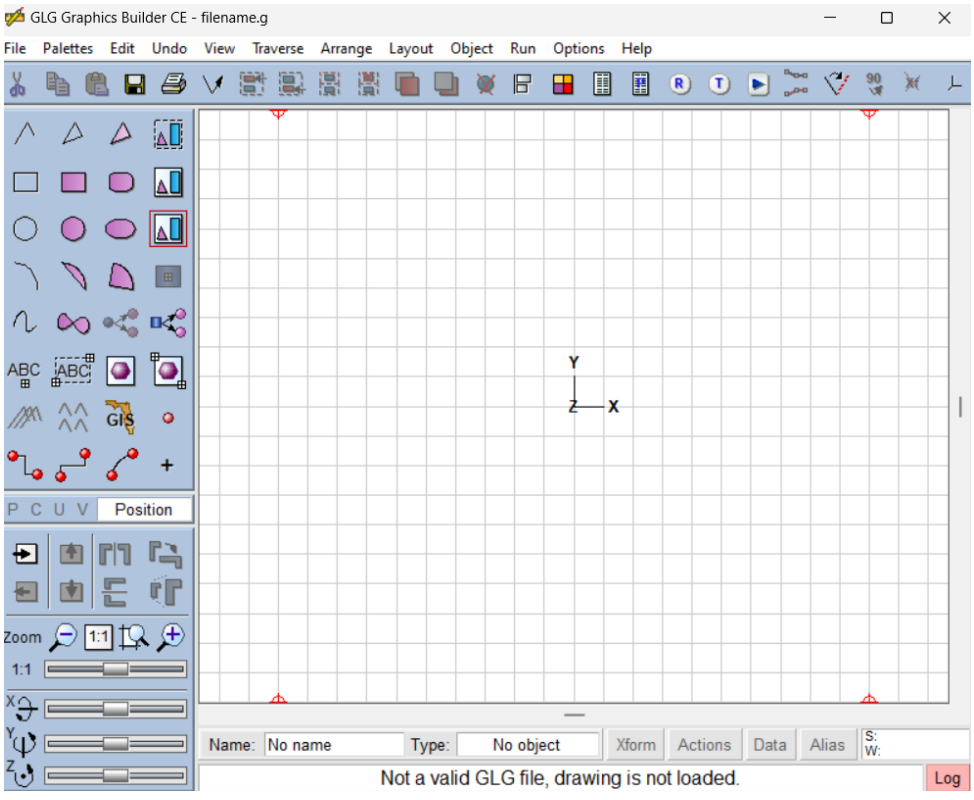
➢ Instructions for Creating and Editing on a Graphics(.g) file:

1. Open the DEMOS_JAVA file stored at **C:\GlgCE.4.3_x64\DEMOS_JAVA** .
2. Select **New> Text Document**

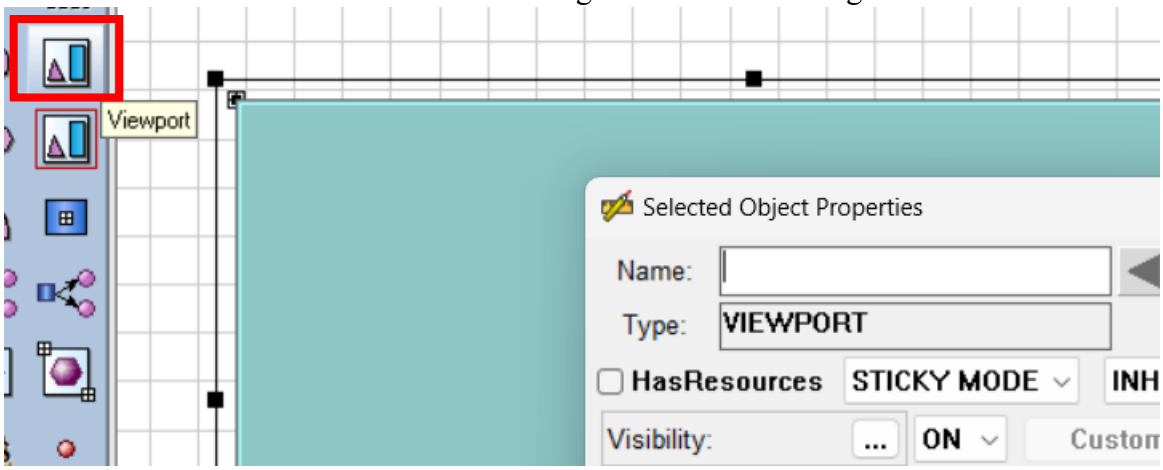


3. Edit file name and extension to **filename.g** and an icon appears as. 

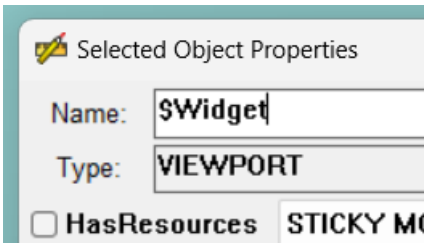
4. Open the file.
5. A workspace for editing and adding widgets appears.



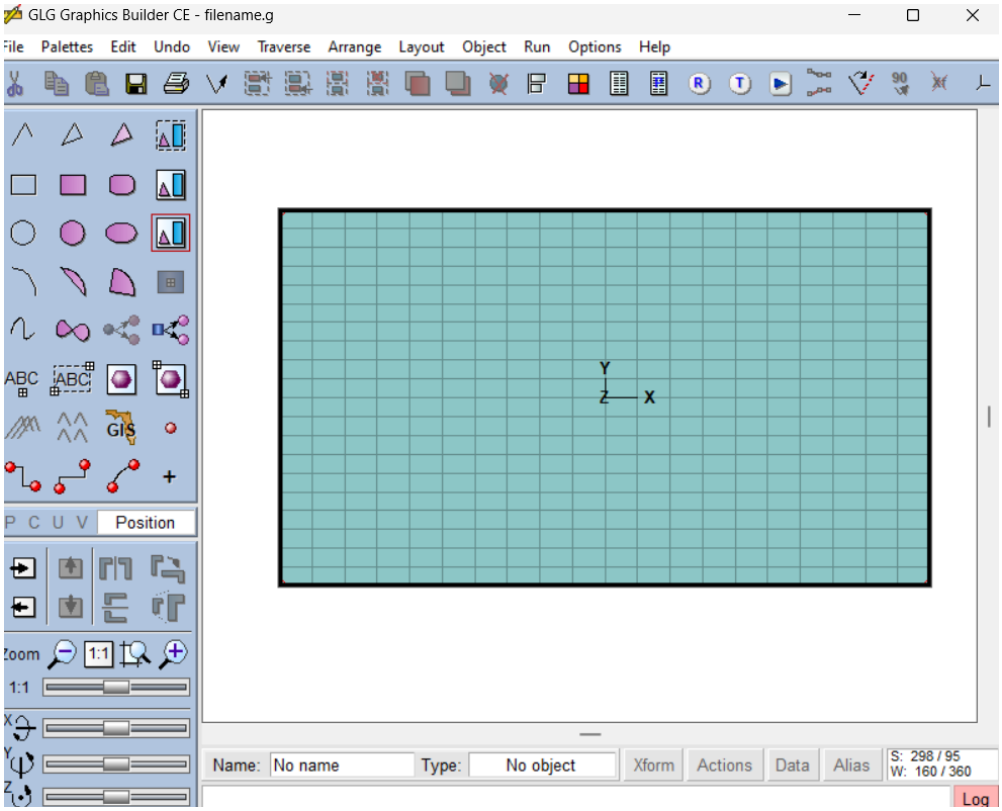
6. To add multiple Widgets such as dials, meters, gauges and graphs create a **Viewport** by selecting this icon on the left hand side and draw a rectangular area to add widgets on.



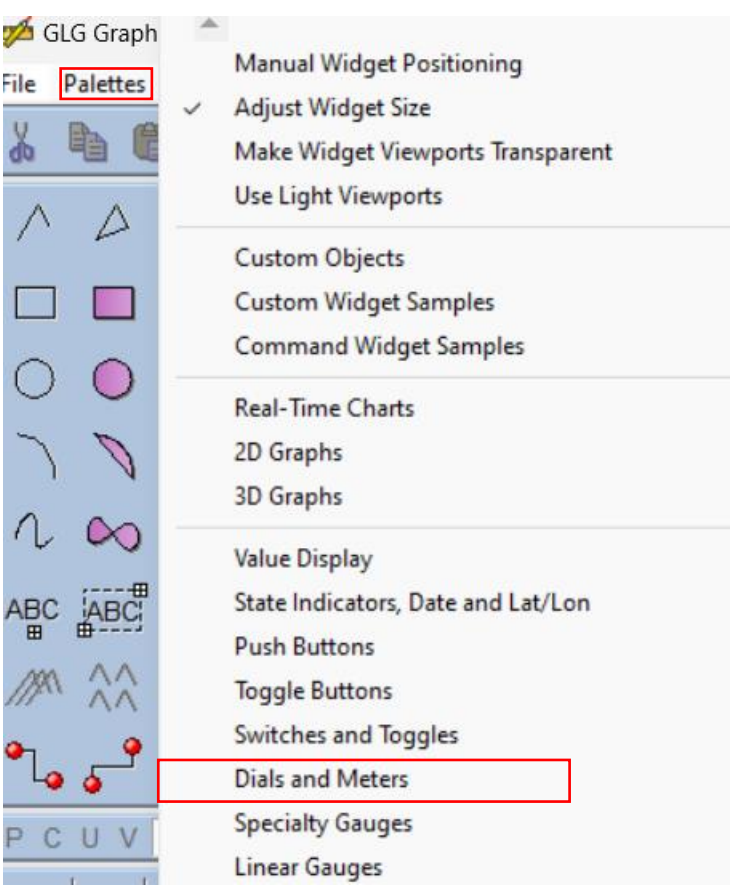
7. In Object properties set the Viewport name to **$Widget** as this is the default viewport name used in demo programs.

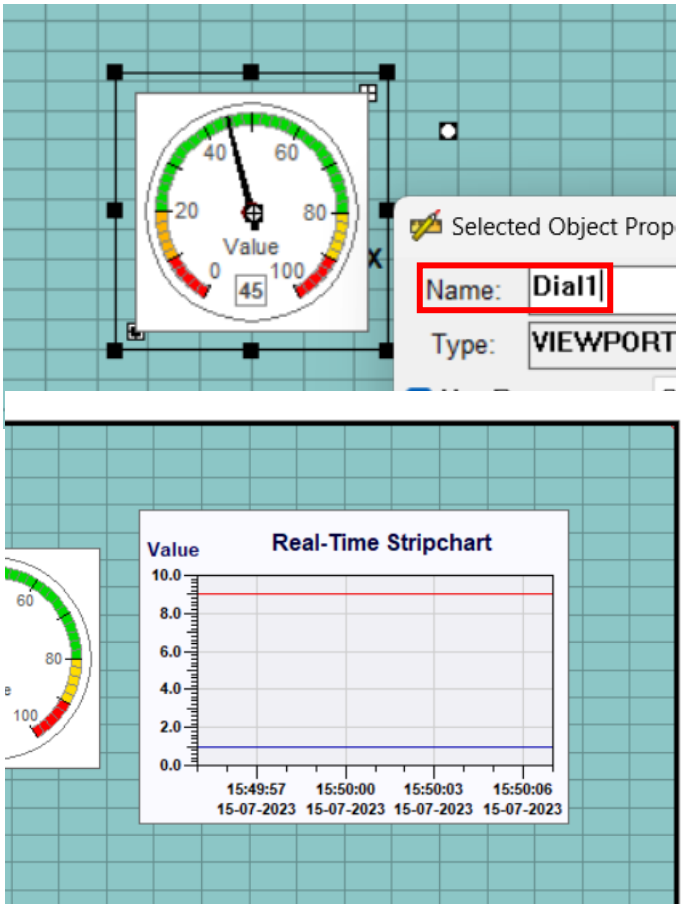


8. To add different widgets first set the **Focus** or **Hierarchy** to Viewport by *Ctr+Shift+click* on viewport or using  icon and clicking on the viewport so that focus shifts to viewport :



9. Then click on **Palettes> Dials and Meters** to add different meters.





10. You can set the meter name to **Dial1** in object properties. Similarly add **realtime charts.**

11. Then select the icon after adding all the widgets to the view port to reset the focus to the workspace.

12. To check for widgets running select icon from the taskbar at top and use the following animation commands: ***datagen d0 10*** *Graph1/**Chart/Plots/Plot#0/ValueEntryPoint***

datagen d0 100 *Dial1/**Value***

(The commands plot graphs for random values between 0 to 10 and meter values from 0 to 100.)

13. For smooth plot or meter testing you can also write *datagen -sin d xxxxxx(rest same).*

14. Now the **.g** file is ready to be integrated with our program in VS Code. For that you need to set the filename of your .g file in the **.SetDrawingName(“***filename.g***”);**

15. And pass the commands for setting meter/ graph values using ***SetDResource*** or ***animation array*** as:

SetDResource("Dial1/Value",velocity); *where velocity is the variable passing velocity value*

animation_array[0] = new GlgAnimationValue(this,GlgAnimationValue.SIN,0, 2 ,vel ,vel, "velgraph/Chart/Plots/Plot#0/ValueEntryPoint"); *where vel is the variable passed to graph.*

Documented by:
-Viswajeet Ray