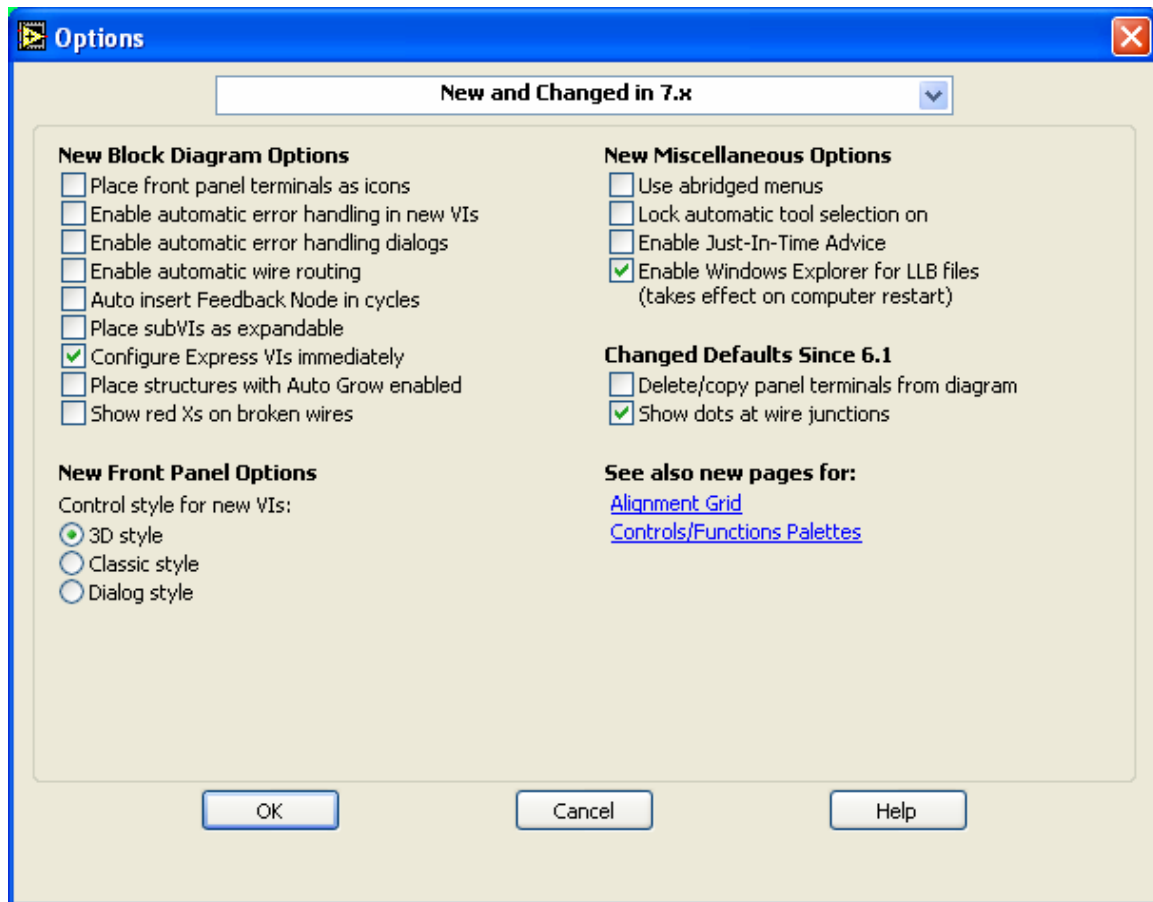


Most new users of LabVIEW just sort of accept the way that it operates as it comes out of the box but you can change some of its behaviors if you find that they are less helpful than you would otherwise desire.

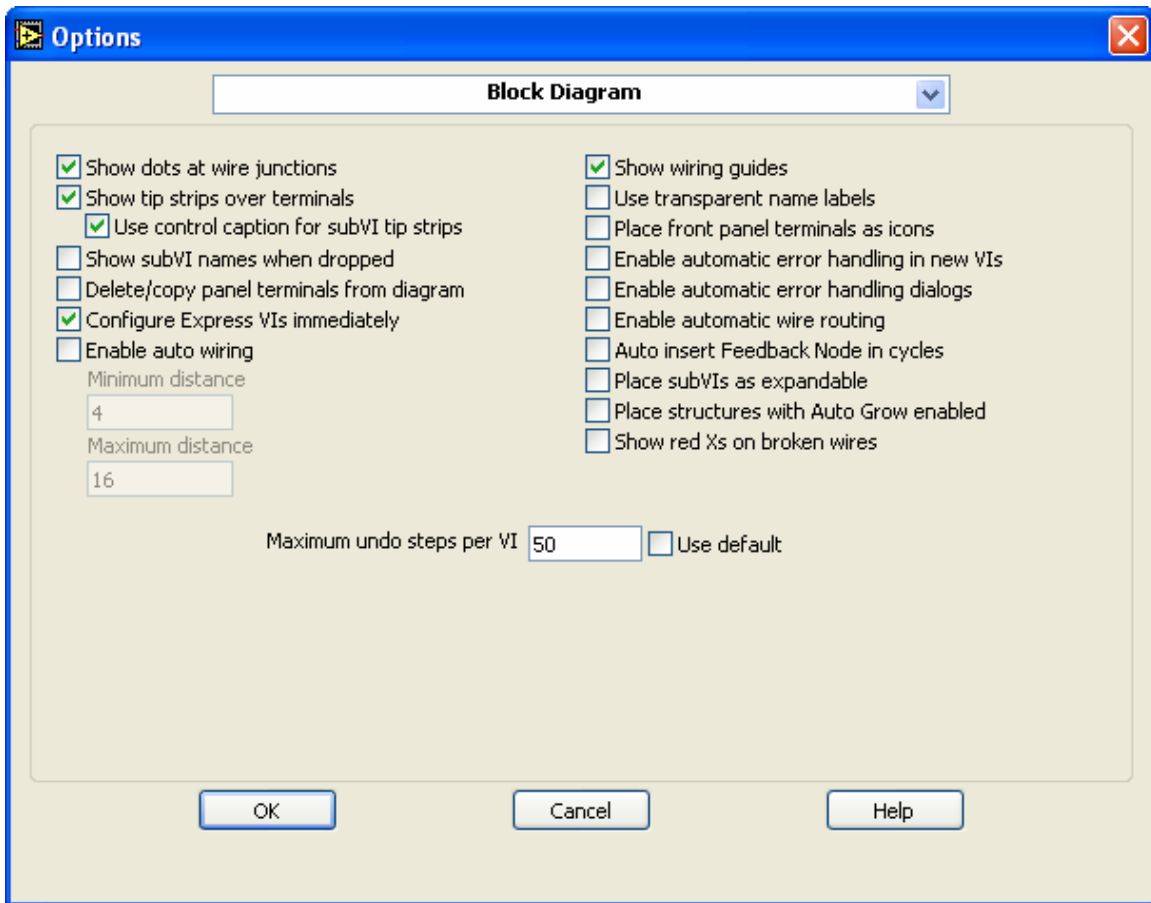
I must admit that I have been using LabVIEW for a very long time and back when I started none of these “intelligent” helpers were available for use. As a result, I am used to the old ways and find a lot of the new helpers more bothersome than helpful.

Here is how I have my LabVIEW setup to “just say no” to these helpers.

(menu) Tools → Options...

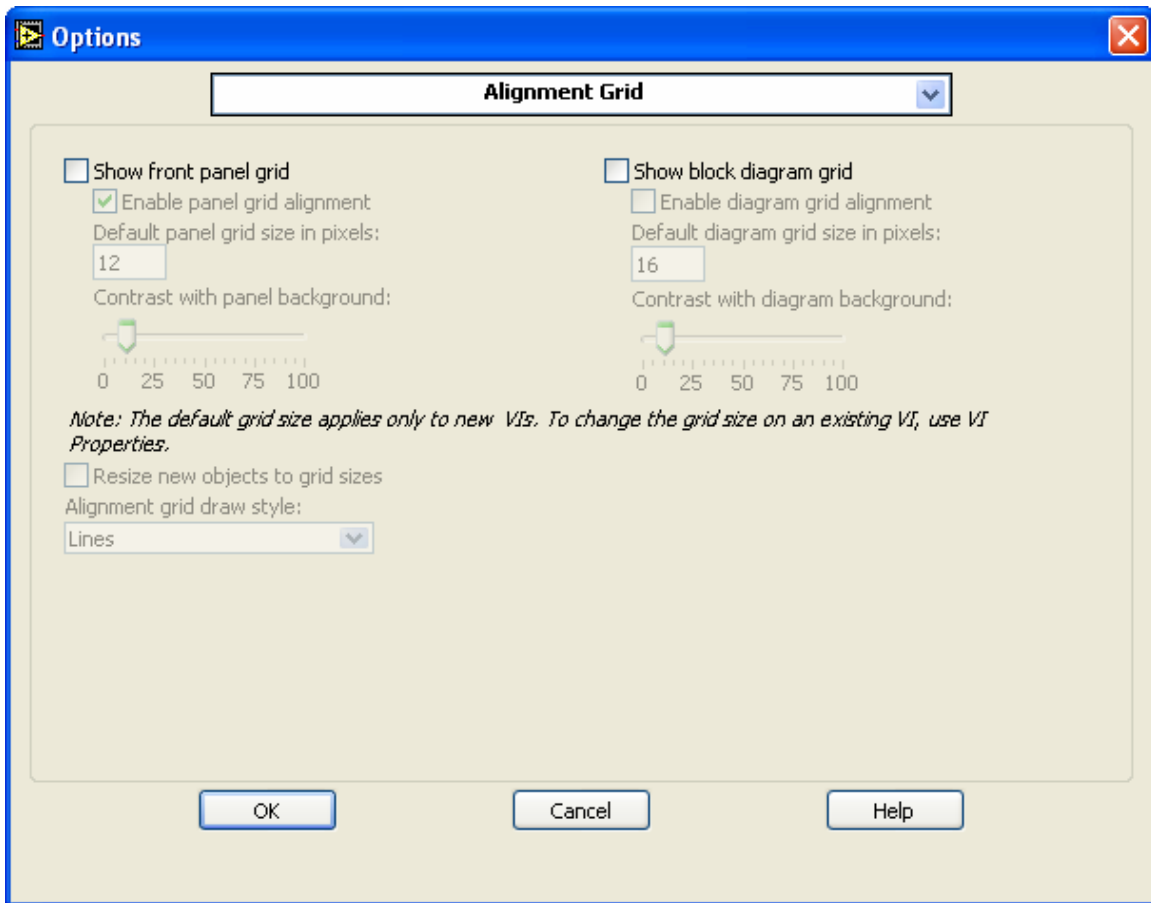


Front panel terminals as icons – I find I like the smaller size of the terminals better than the icons.  
Automatic error handling – causes a runtime performance hit. I’d rather do my own error handling.  
Automatic wire routing – I’d rather manage the block diagram layout.  
Autoinsert feedback nodes – when I want them, then I can insert them  
Expandable subVIs – take up too-much space on the block diagram  
AutoGrow structures – again, I’d rather manage the block diagram layout. It’s not hard to grow a structure if you hold down the CTRL key and then drag the arrow/positioning tool.  
Show red X’s – I don’t need a fat red X to see that the wire is broken.  
Use abridged menus – The full menu fits on my screen. Why waste mouse clicks on an abridged menu?  
Automatic tool selection – The auto tool selector & I can never agree on the tool that I want. The TAB key or the SPACE BAR work better for me (I’ve developed a rhythm...)  
Just in time advice – it tells me things I already know and gets in my way.  
Enable windows explorer for LLBs – it can be handy when it works. It doesn’t always work.  
Delete/copy terminals from diagram – handy but dangerous. I’ll do without.  
Show dot at wire junctions – useful to distinguish between wires that cross and wires that join.

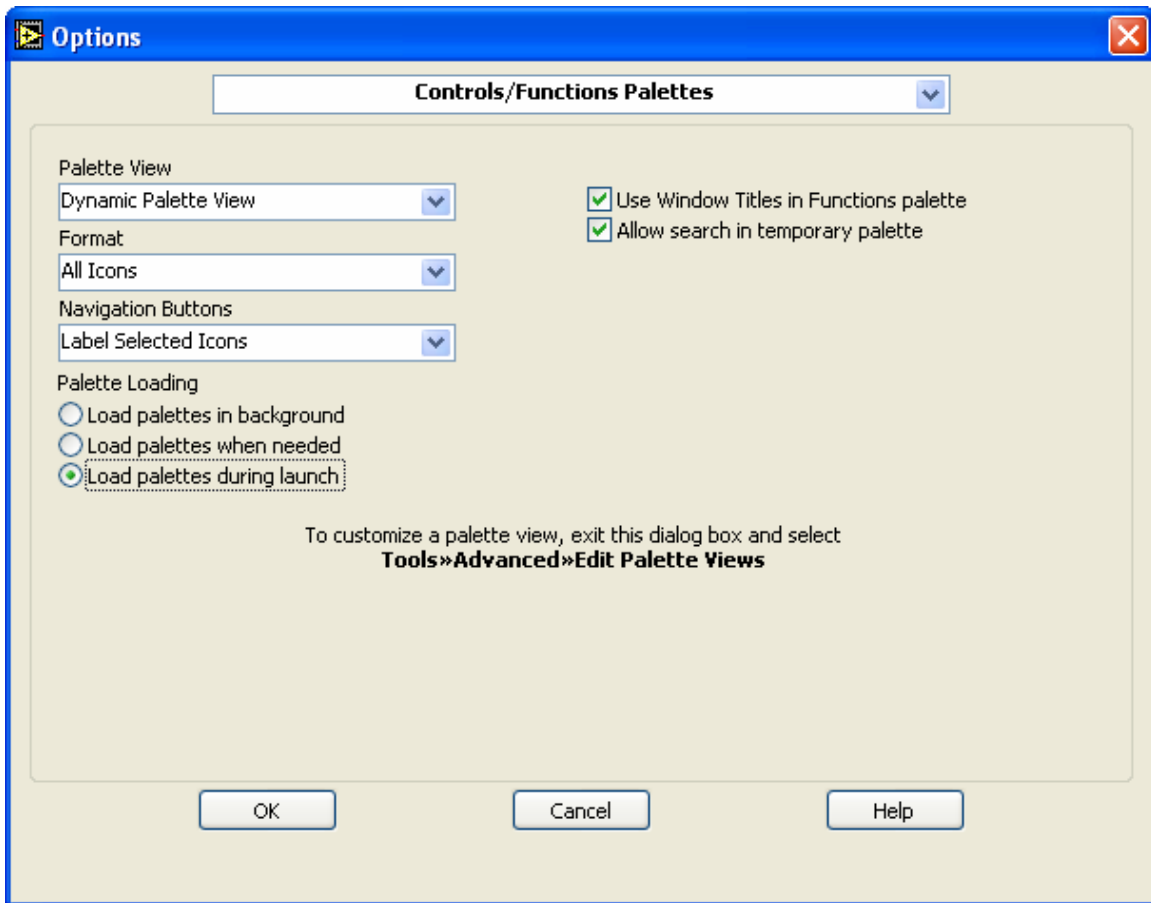


Show tip strips over terminals – can be handy to make sure the wiring tool is connecting to correct point  
Show subVI names – takes up more space on the block diagram. If I want the name to show I know how to show it.

Maximum undo steps – USE A REASONABLY LARGE NUMBER HERE! The default value gives you little room to change your mind.

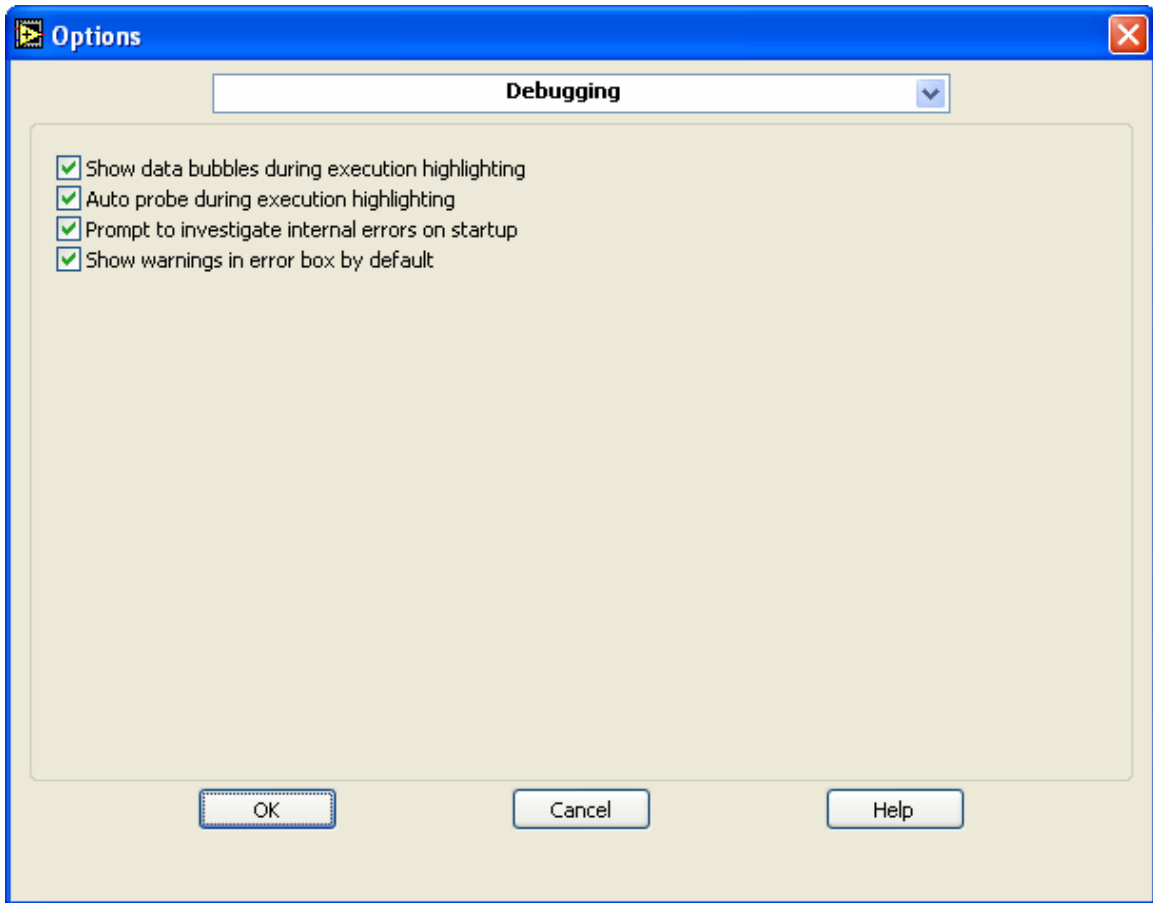


The grids get in my way, especially if the grid alignment option is turned on.

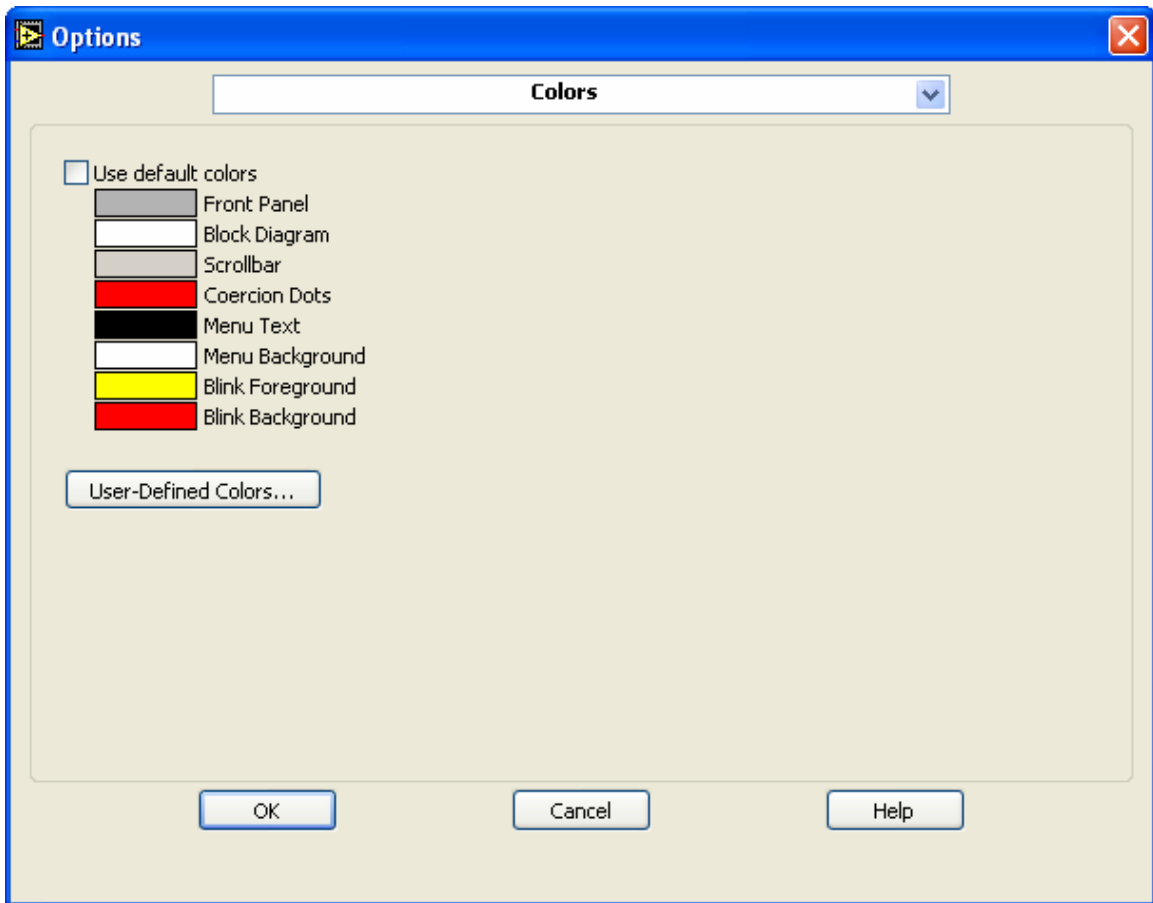


The Dynamic Palette View is something that the [OpenG](#) library makes use of and I love having the [OpenG](#) library available.

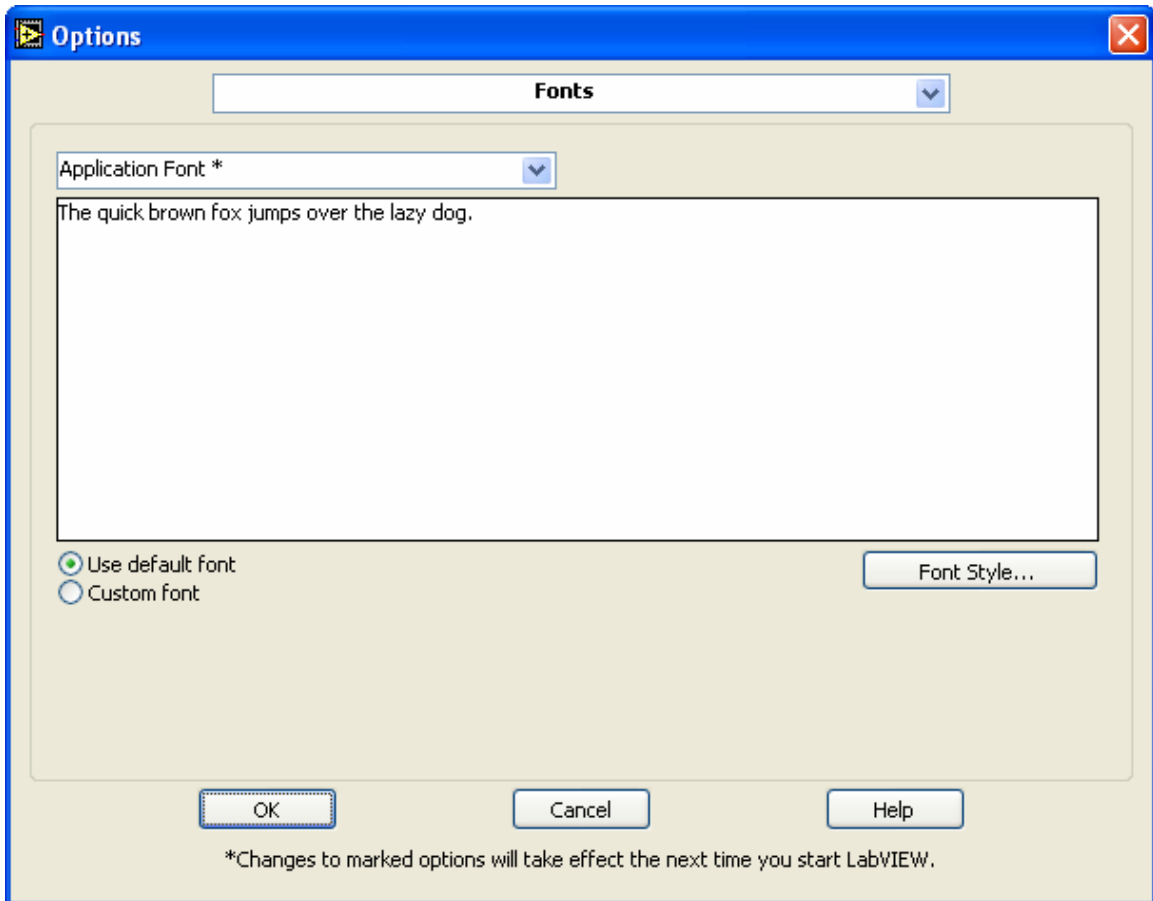
The installation process uses something called [OpenG Commander](#) and, while it can be a bit confusing, it is more than worth the trouble in the end!



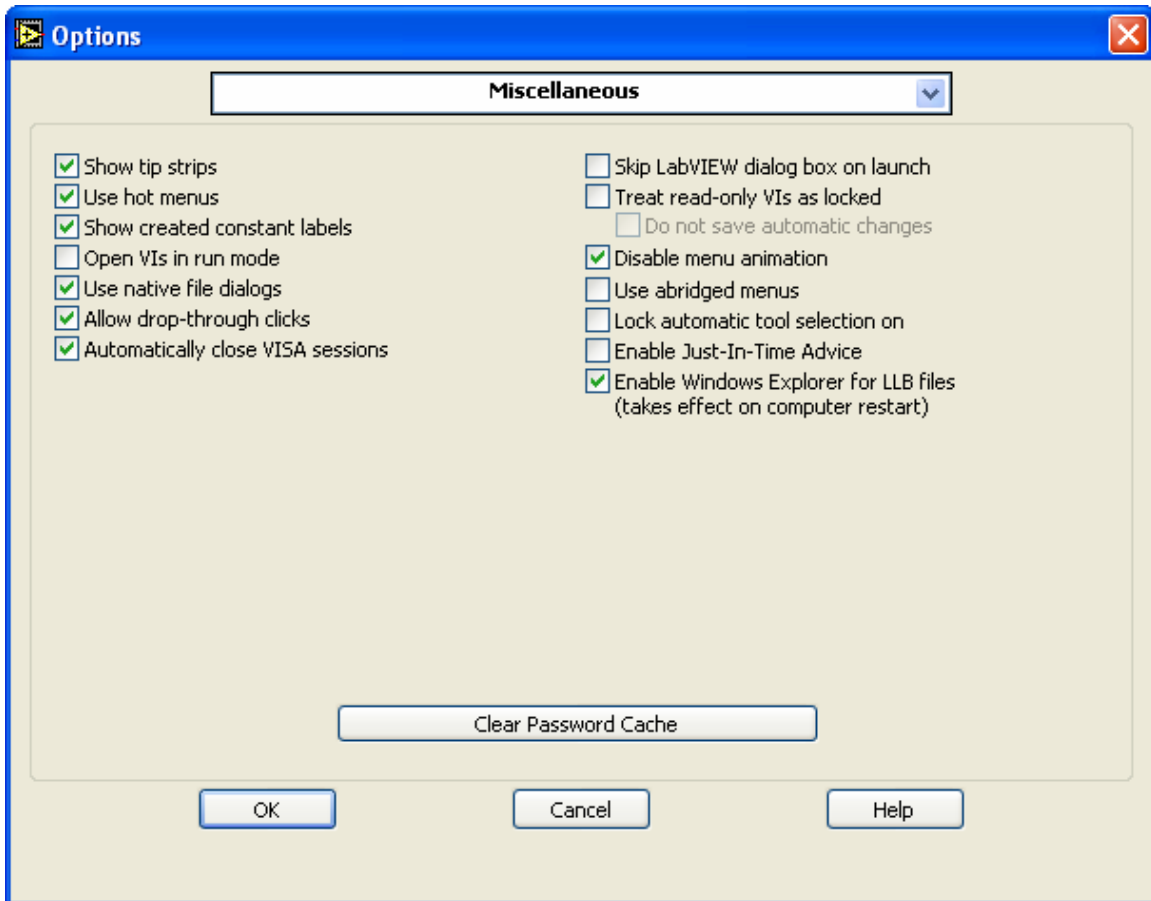
Show data bubbles & Auto Probe are both useful during debugging.



Pick a color for your coercion dots that will stand out (and then program to avoid the dots!)



Using fonts other than the default font may make your application less portable (or more ugly) as you move it to other machines.



Show tip strips – they sometimes get in the way but I find that their usefulness outweighs their interference.

Hot menus – they allow for more casual menu browsing

show constant labels – I like to label most constants

Open in run mode – I'm usually going to be editing it rather than running it on a development system

Native file dialog – I'd rather use the Windows file dialog than the NI file dialog

Allow drop-thru clicks – If I can see something to click, I want to be able to click it on the 1<sup>st</sup> try.

AutoClose VISA – I try to do this within the code but this provides a good backup.

Dialog box on open – I don't need it.



Some useful links for beginners:

The installation process uses something called [OpenG Commander](#) and, while it can be a bit confusing, it is more than worth the trouble in the end!

[Share your favorite tips and shortcuts](#) and more tips are [here](#)

[The LabVIEW FAQ Subforums](#) If you will be doing much LabVIEW programming then make sure you read the state machine information [here](#). The tunnel wiring wizard found [here](#) to can be very useful when working with state machines.

Tutorials:

[Three Hour LabVIEW Introduction Course](#)

[Six Hour LabVIEW Introduction Course](#)

[LabVIEW Graphical Programming Course](#)

Another useful utility for showing hidden controls: [Hidden Controls](#)

Somewhat dated but still useful Windows management VIs can be found in [these lvutil32 libraries](#), such as renaming, resizing, and running an application window

The NI examples library is [here](#).

The NI LabVIEW forum is [here](#).

Remember that if you are ever at a loss about how to treat a front-panel or block diagram object (or wire), try opening the Context Help Window (CTRL-H) and then poke the object with various tools to see what shows up in the help window. Not every tool will generate the same message.

It is also useful to right-click on objects to see what various menu options are presented for the objects.