Created by Nick Nielson
ECE 5580

Tutorial: Using GEL files in Code Composer Studio V4

References

1. (Course Textbook) Real-time DSP from Matlab to C with the TMS320C6x DSK by Welch, Wright and Morrow. See section 11.4.2, pages 179-180.
2. http://processors.wiki.ti.com/index.php/GEL

GEL File Syntax

GEL files follow simple syntax patterns, as shown below.

```
1  menuitem "Gain control";
2
3  slider Right(0, 100, 1, 1, right_slider)
4  {
5    right_gain = right_slider * 0.01;
6  }
```

Here is what each line of the code above does:

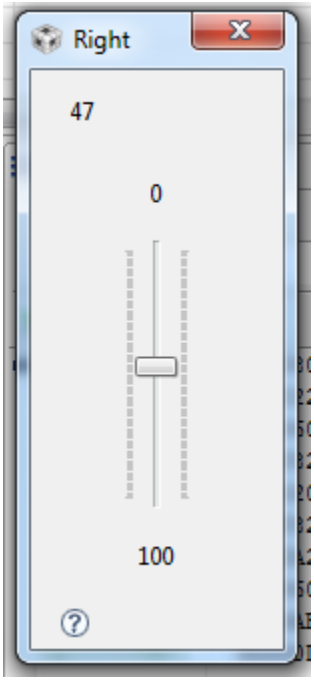
1	Create a menu item with name of *Gain Control.*
2	
3	Create a slider object with name *Right*, with range of 0 to 100. The scroll increment is 1, and the page scroll is set to 1 as well. Typically these values can be left as 1. The GEL variable that holds the current value of the slider is named *right_slider*.
4	Take the value of *right_slider* and multiply it by 0.01 (equivalent to division by 100
5	but avoids time-consuming division operation). Save the result to the global variable named *right_gain*. In your project, the declaration of this variable would be a float, such as `volatile float right_gain = 0.5;`
6	

Slider values do not have to set a floating point variable. Integers can be set just as well. This is demonstrated via the following code example.

```
slider NumBitsToTruncate(0, 16, 1, 1, num_bits)
{
  num_bits_to_truncate = num_bits;
}
```

Here, the global variable *num_bits_to_truncate* is set via the slider NumBitsToTruncate. The range is 0 to 16.

In CCS v4, the *Right* slider created above will look like this, where the current value is 47:

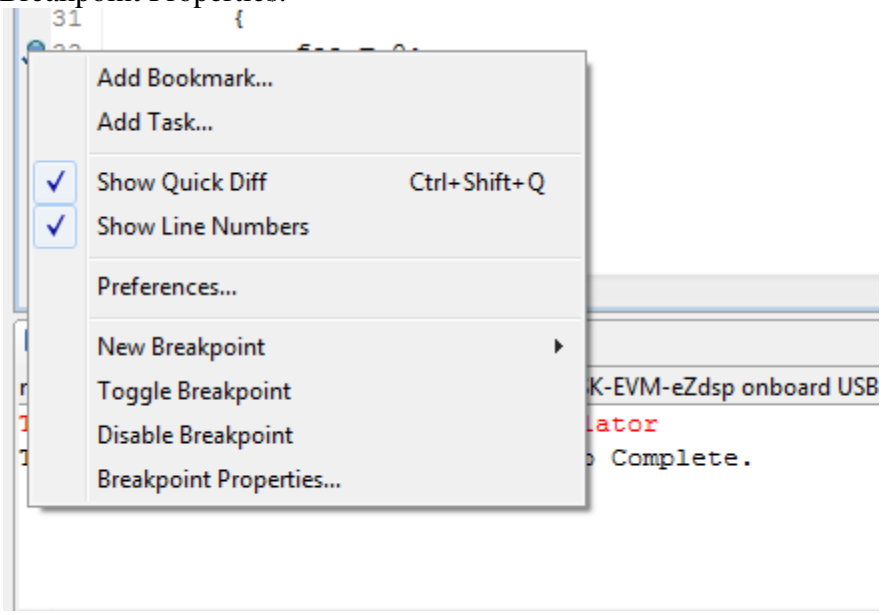


Using a GEL file with CCS V4

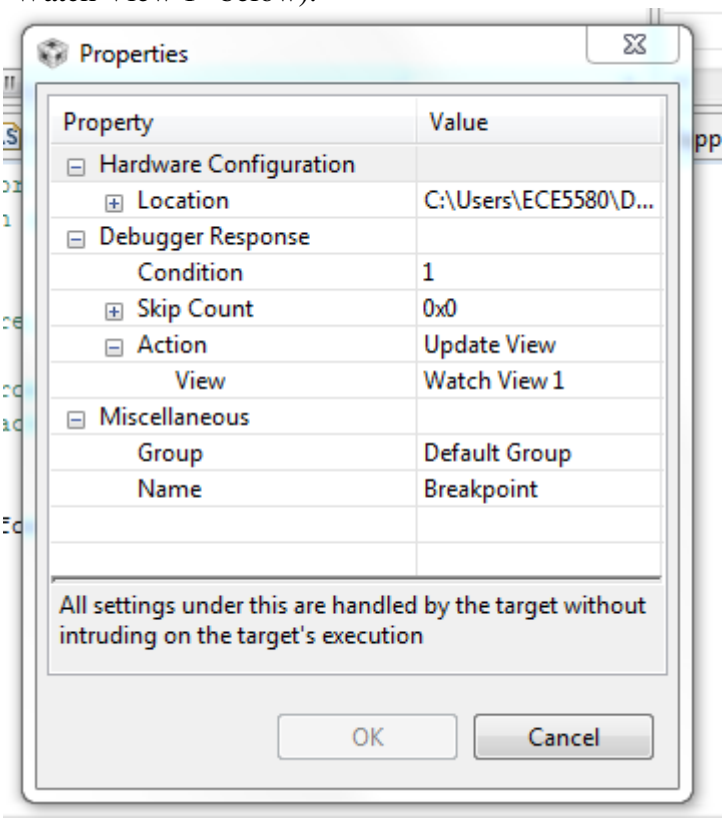
After you have added a GEL file to your project and written the code to describe the control you want, the following process should be followed to enable the GEL functionality in CCS V4.

1. Start the debugger perspective.
2. In the watch window, add the global variables whose values are set by the GEL sliders to the watch list.
3. Create a breakpoint on a line in your code.

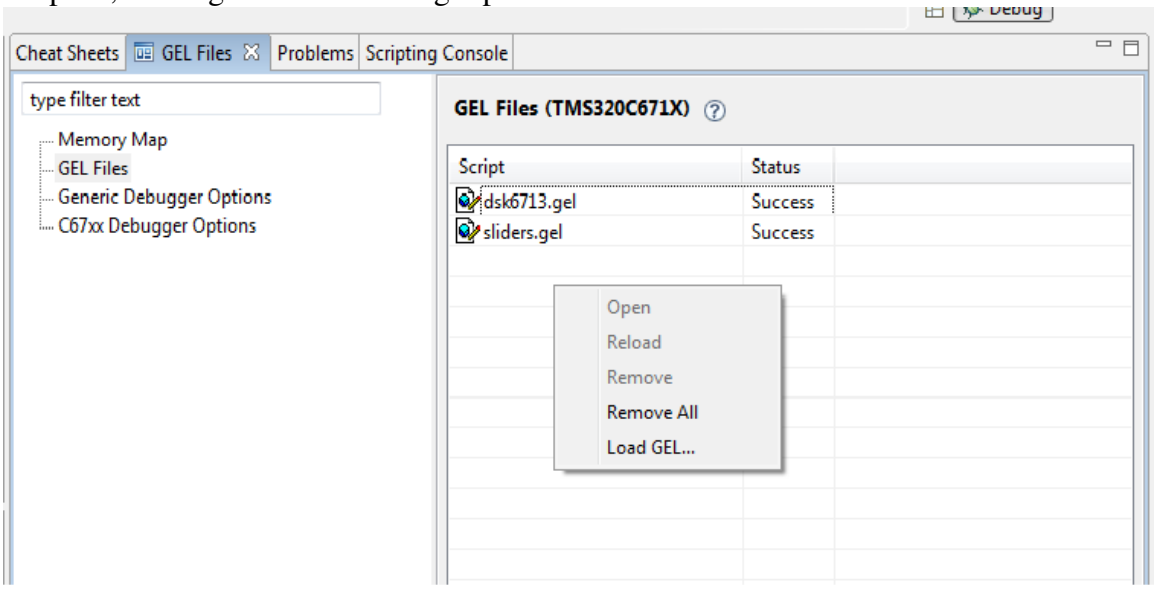
4. Modify the breakpoint settings by right-clicking on the breakpoint and choosing Breakpoint Properties.



5. Set the breakpoint properties as shown in the screenshot below. Under the Debugger Response section, set Condition to 1, Skip count to 0, and Action to "Update View". The View that you want to update is your watch window (called "Watch View 1" below).

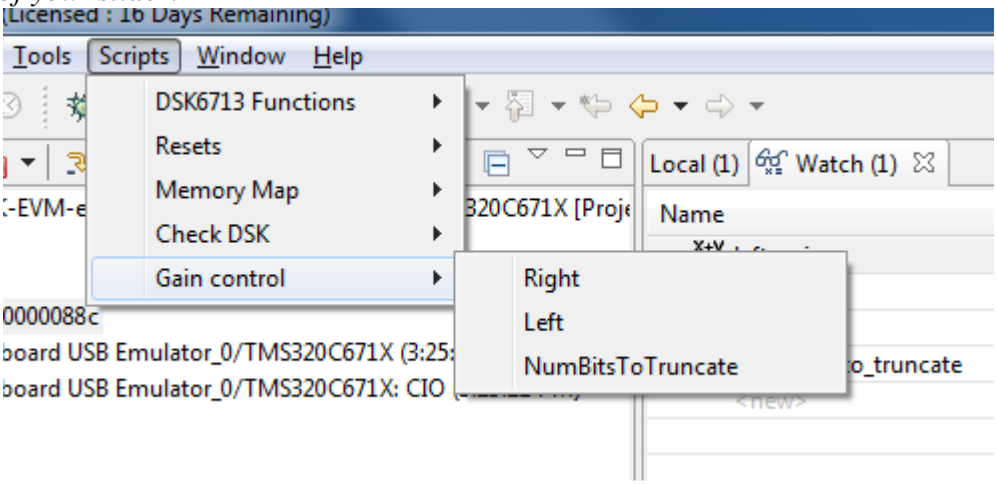


6. In the “GEL Files” tab of the debug perspective, right click on GEL Files in the left pane, then right click on the right pane and select “Load GEL”.

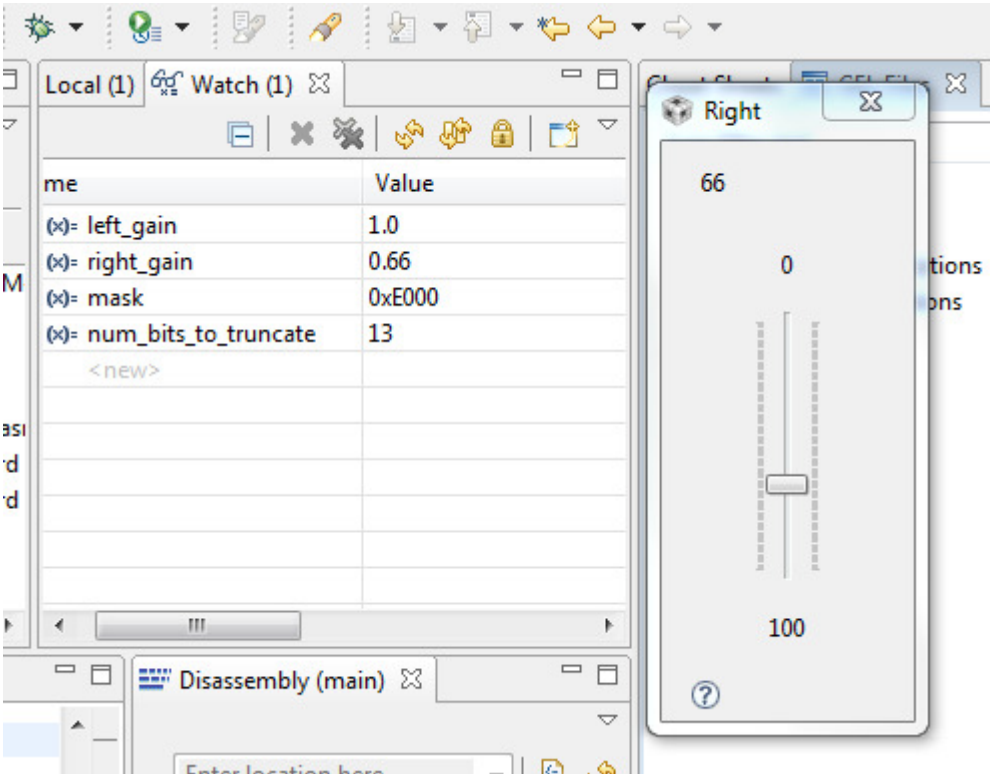


7. Browse to the GEL file you wish to load and select it. In the screenshot above, the GEL names sliders.gel has already been loaded.

8. Bring the GEL sliders up by selecting *Scripts* -> *name of your menuitem* -> *name of your slider*.



9. Run your application in the debugger. Observe the watch window as you can see the values changing.



Helpful Tip

Where the breakpoint is placed is important. For example, if the breakpoint is in a location that will be triggered very often (in main.c's while loop or in an ISR) the breakpoint will trigger, causing the DSP to halt momentarily while the global variables are updated (from the sliders). This would happen even if the values hadn't changed, and hitting the breakpoint this often will result in poor performance (e.g. lack of any spectrum output when the audio is analyzed on the PC). It is therefore a good idea some code around the breakpoint that will cause it to not be triggered very often, as one does not typically need the values to be updated continuously. An example is given below, where the breakpoint would be placed on the "foo=0" line. There are some downsides in that the GEL sliders will not always update the global variable.

```
// main stalls here, interrupts drive operation
while(1) {
    i = i + 1;//from original code
    //begin hack for GEL
    foo += 1;
    if(1000000 < foo)
    {
        foo = 0;
    }
}
```