

GDSview User's Manual

Version 4.5

Integrand Software, Inc.

Copyright © 2008–2014

1 Introduction

GDSview is a layout viewer for GDSII files. It is intended for use with the EMX* electromagnetic simulator. GDSview reads an EMX process file in addition to the GDSII layout, and it processes the layout in exactly the same way that EMX does. Thus, it can be used to visualize the effect of the various layout manipulation and simplification operators in the process file.

You invoke GDSview like this:

```
gdsview layout.gds mystructure foundry.proc
```

This command tells GDSview to display the structure `mystructure` within the GDSII file `layout.gds`. The layout is processed according to the process file `foundry.proc`. If you omit the process file, GDSview will display the raw layout.

There are two versions of GDSview. One (the `gdsview` executable) has a basic X Window System interface, and the other (the `gdsview-ogl` executable) has a more fancy OpenGL[†] interface. Two executables are provided in case your computer lacks the necessary shared libraries for the OpenGL version. Normally when you run `gdsview`, it will automatically try to pass control to the OpenGL executable; if this fails, then a warning is printed and the basic X Window System interface is used instead.

GDSview also accepts some command-line options. The following options have the same meaning as they do for EMX:

--label-depth=number By default, GDSview only displays labels in the top level GDSII structure. If you have labels further down in the hierarchy, you can specify how deep GDSview should look using this option. The default, 0, means only look at the top level. Specifying **--label-depth=1** would mean to look in sub-structures of the top level structure too. You can abbreviate this option as **-1**.

--unique-labels Adds a numeric suffix to all labels so as to ensure that all labels are distinct. For example, multiple occurrences of label **a** become

*EMX is a registered trademark of Integrand Software, Inc.

†OpenGL is a registered trademark of SGI.

`a_1`, `a_2`, etc. This is mostly useful when you have multiple instances of a single cell with some labels in it and you want all of the labels to remain distinct. Using this option with GDSview lets you tell where the unique labels are.

- `--cadence-pins=number` When a GDSII file is streamed out from Cadence, there is an option to write special property records in the file that encode pin information. This option tells GDSview to interpret the pin information and add labels to the layout at the pin locations. See the EMX manual for more information.
- `--port=port specifier` This is the same as the `--port` option to EMX, and it tells GDSview to display only the labels relevant to the given ports. You may abbreviate `--port` as `-p`.
- `--signal-ground-separator=character` This sets the separator for the signal and ground parts of a port to the specified character, as in EMX.
- `--internal=specifier` This option is like the same option in EMX; it specifies the size of the internal connection point for a label. You can abbreviate this as `-I`.
- `--via-separation=number` To be consistent with EMX, by default GDSview merges vias that are closer than a certain distance to each other. The default is distance 0.5 (microns). If you don't want to merge vias, specify 0. This option may be abbreviated by `-v`.
- `--local-reflex-scaling=number` This option affects the merging of diagonal vias. For more information, see the documentation on this option in the EMX manual.
- `--width=number` When the process file contains width- and spacing-dependent quantities, the calculation of their values can be somewhat time-consuming. This option makes GDSview assume that the local geometry width is equal to the specified number of microns (instead of calculating it). If you specify both `--width` and `--spacing`, then the time-consuming calculations are avoided.
- `--spacing=number` Assume that the local geometry spacing is equal to the specified number of microns.
- `--no-geometry-dependence` Turn off local width and spacing computation.
- `--scaling=number` Scale the GDSII layout by the specified amount, i.e., do a lateral shrink or expansion of the layout. Scaling is applied when the GDSII file is read. Operations in the process file (such as `grow`, `merge`, etc.) are done *after* the scaling is applied. You can use `-s` as a short form for this option.
- `--define=name,number` This is equivalent to adding a definition of the form:

define name=number

in the process file.

--definitions-file=name This specifies an additional process file which will be read before GDSview reads the main process file.

--key=string When using an encrypted process file, this option provides the decryption key.

--case-sensitive By default identifiers in the process file are internally converted to lower case. This option makes GDSview preserve the case of identifiers instead.

--excluded-cells-file=name Omit any cells in the GDSII file whose name matches one of the regular expressions in the specified file. The file should have one regular expression per line.

--included-cells-file=name This is like the **--excluded-cells-file** option, but GDSview will only include cells that match (or subcells of those cells), instead of excluding them.

--device-cells-file=name Any cells in the GDSII file whose name matches one of the regular expressions in the specified file will be treated as “black boxes.” GDSview will show internal labels of the form `~x0000`, `~x0001`, ..., for the terminals of black-box devices.

--corner=name Selects a particular process corner. Any random variable in the process file with a corresponding **corner** value will be fixed at that value. Multiple **--corner** options may be given, but only one may apply to any given random variable.

--colinear-fuzz=number In EMX, input polygons usually are simplified by eliminating “nearly colinear” vertices. This option has the same effect as the corresponding EMX option in controlling this behavior. See the EMX manual for details.

--internal-units-per-micron=number The geometry in GDSview is represented by polygons whose vertices lie on a grid. The internal grid spacing defaults to one nanometer (the same as in most GDSII files). This option changes the number of grid units per micron. Increase it to make the grid finer; for example, setting this option to 10000 makes the internal grid spacing 0.1 nm.

--label-snap-distance=number GDSview by default draws a line between a label and the edge to which the label will “snap” in EMX. This option controls the snap distance, as in EMX.

--device-label-snap-distance=number This option controls the snap distance for labels that come from device terminals.

- `--device-pins-internal` Treats all Cadence shape pins within black-boxed device as if they have access direction TBLR, so the pins will become internal ports.
- `--version` Print GDSviews's version number and exit.
- `--x11` Force GDSview to use only the basic X Window System interface. Use this if the OpenGL version will not run on your system due to, e.g., missing shared libraries.

These options are unique to GDSview:

- `--layers=layer list` Tell GDSview to show only the certain layers, or to omit certain layers. Example layer lists:
 - `me1,me5,me6` Show only `me1`, `me5`, and `me6`.
 - `-me1,me2` Do not show `me1` or `me2`.
- `--no-id` This turns off the display of whatever the mouse is pointing to. This can be significantly faster for complicated layouts. The abbreviation for this option is `-n`.
- `--no-labels` Tells GDSview to turn off the label display. You can abbreviate this as `-N`.
- `--no-nets` Turn off the display of nets (OpenGL version only).
- `--font=name` Use the specified font for display of labels and rules. The `--font` option can be abbreviated as `-f`. Available font names for the X Window System version can be found using the standard `xlsfonts` executable. Fonts for the OpenGL version have the form *name-size* or *name*, where *name* is a font name and *size* is the height of the font in pixels (defaulting to 14 if the size is omitted). The valid names can be found using the `--list-fonts` option.
- `--list-fonts` Show all the valid font names and exit (OpenGL version only).
- `--raw-polygons` When used without a process file, this option keeps GDSview from merging polygons. This is mostly useful for debugging, to see exactly what is in the GDSII file.

2 Controlling GDSview (X version)

The basic X Window System version of GDSview displays a simple window with the layout, as in figure 1. The line styles in the layout have the following meaning:

Thin solid line a regular edge

Thin dotted line connects an edge label and the edge that it will snap to.
If a label is too far from any edges to snap to them, it will be displayed between question marks.

Thin short dashed line the boundary of named resistor and capacitor regions in the process file

Thick dashed line the region for an internal label

Thin long dashed line the boundary for a black-boxed device

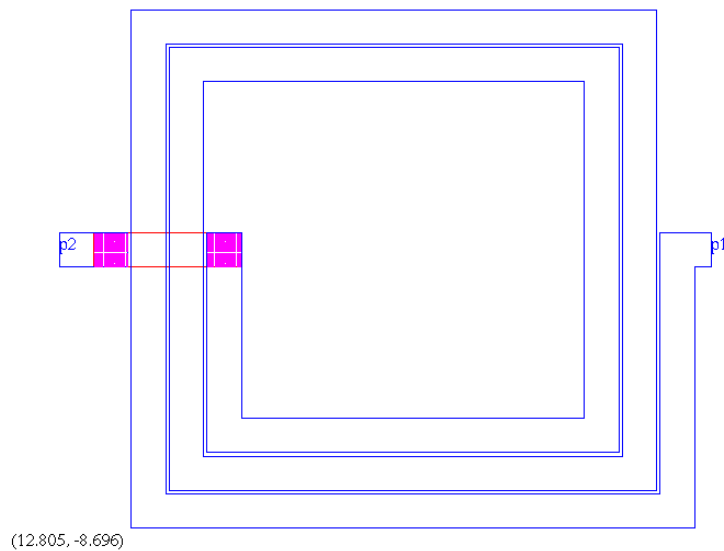


Figure 1: GDSview (X Window System version) layout window

You use the keyboard and mouse to control what is displayed. If you hold down the first mouse button (typically the left one), sweep out a rectangular area, and then release the button, GDSview will zoom in on the selected region (figure 2). If you hold down the second mouse button (usually the right one) and sweep, GDSview displays a rectangle with length and width information, but does not zoom in when you release the button.

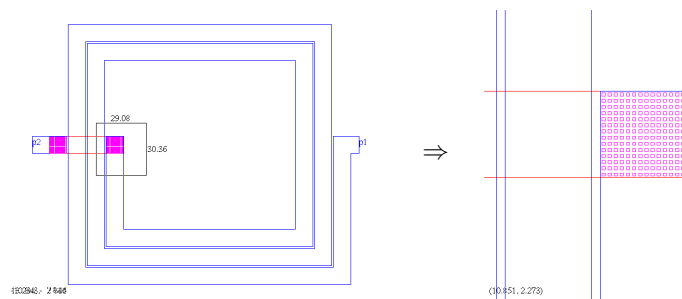


Figure 2: Zooming in with the left mouse button

Typing **z** makes GDSview zoom in by a factor of 2 (figure 3).

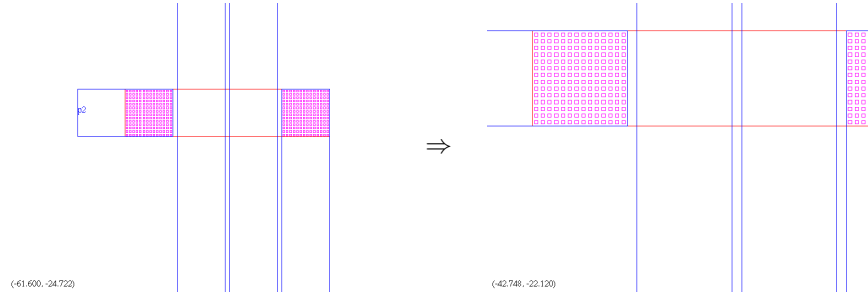


Figure 3: Zooming in with the **z** key

Typing **x** makes GDSview zoom out by a factor of 2 (figure 4).

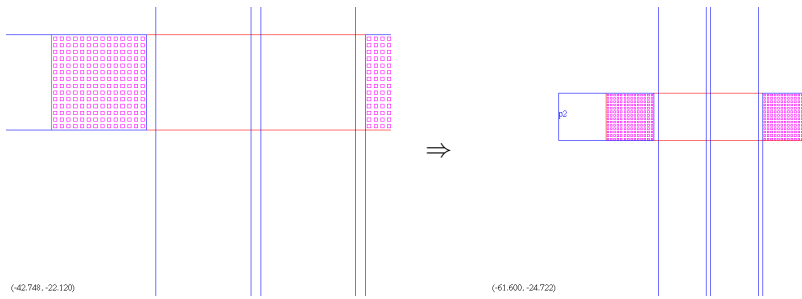


Figure 4: Zooming out with the **x** key

The `c` key makes GDSview recenter and rescale the layout so that everything is visible (figure 5).

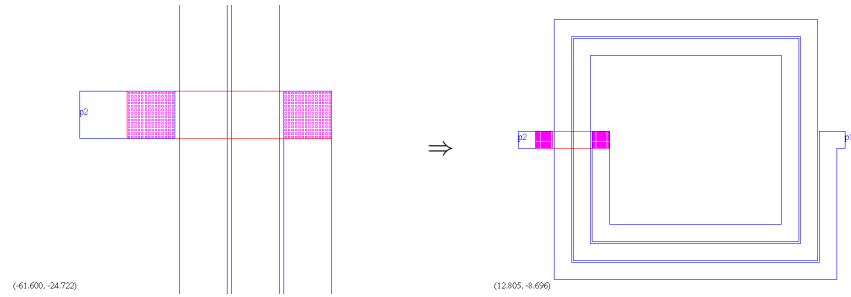


Figure 5: Recentering with the `c` key

The arrow keys (up, down, left, and right) can be used to scroll around the layout. The viewport moves in the direction of the arrow key (figure 6).

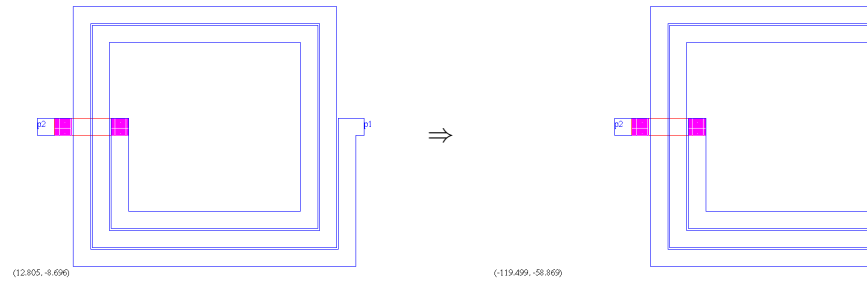


Figure 6: Scrolling with the left arrow key

If you move the mouse pointer near an edge, GDSview displays the name of the layer. When a layer name is displayed, typing **h** hides that layer (figure 7).

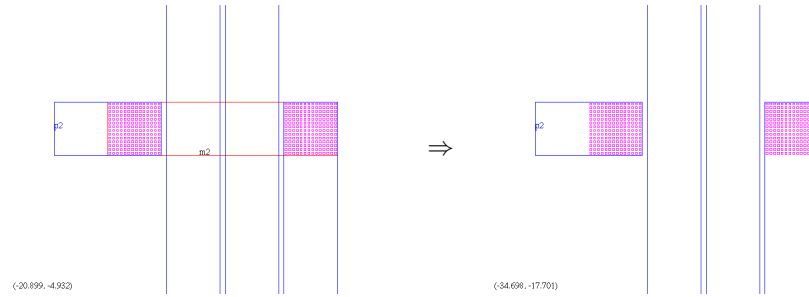


Figure 7: Hiding a layer with the **h** key

Type **v** to make all layers visible again (figure 8).

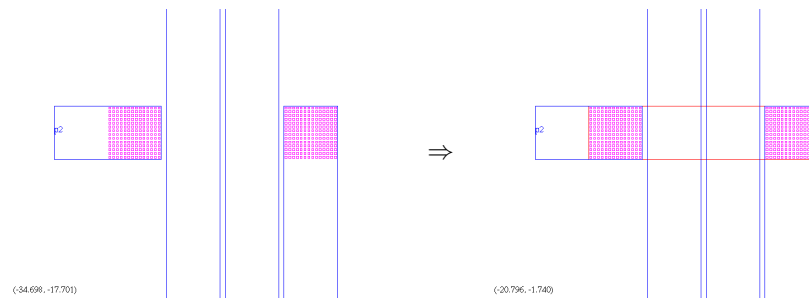


Figure 8: Making layers visible with the **v** key

When a layer name is displayed, click the third mouse button (usually the middle button on a three-button mouse, or a simultaneous click on a two-button mouse) to print an `--add-label` option to standard output (figure 9). If you copy the option and use it on the EMX command line, EMX will operate as if the GDSII file has a label on the specified layer. The label defaults to `p1` for the first click, `p2` for the second click, etc., but you can edit the label name. This method is useful for adding edge ports to a layout without having to modify the GDSII file. If you type the `i` key when a layer name is displayed, and then move the mouse and click the third mouse button, GDSview will print an EMX option to add an internal port at that position.

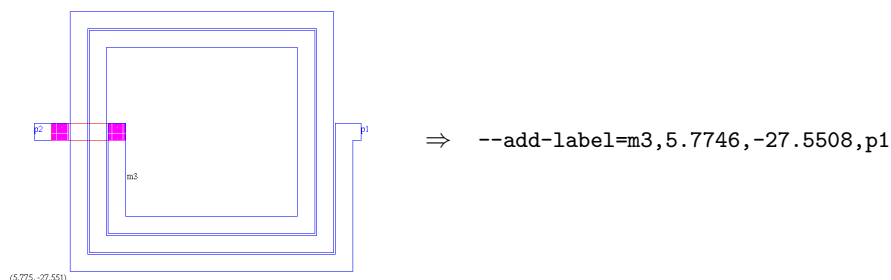


Figure 9: Adding labels with the mouse

Typing `p` produces an encapsulated PostScript picture of the current window (this is not available in the OpenGL version). The output file is called `layout.eps`.

Type `q` or the escape key to quit GDSview.

3 Summary of GDSview controls (X version)

First (left) mouse button Sweep an area to zoom in

Second (right) mouse button Measure an area

z key Zoom in by a factor of 2

x key Zoom out by a factor of 2

c key Recenter the layout

Arrow keys Scroll around the layout

h key Hide a layer

v key Make all layers visible

Third (middle) mouse button Print an EMX option to add a label

p key Write encapsulated PostScript file

q key Quit GDSview

4 Controlling GDSview (OpenGL version)

The window for the OpenGL version of GDSview looks like figure 10, with a main layout display and a sidebar. The sidebar has control buttons and shows layer names. If you're using GDSview with a process file, then the sidebar also shows net names.

The basic key and mouse controls within the layout display are the same as those mentioned earlier (and summarized in the previous section), plus there are a few other available keys:

w key Go back to the previous view (useful if you want to zoom in closely for a moment to check something)

r key Raise the pointed-to layer to the top in the display order

l key Lower the pointed-to layer to the bottom in the display order

0, 1, ..., 9 keys Set the opacity of the pointed-to layer to 0%, 10%, etc.

f key Set the opacity of the pointed-to layer to 100%.

s key Makes the opacity of the pointed-to layer be controlled by the fill slider.

d key Reset the display order to the default (layers are drawn in the order that they appear in the process file, typically from substrate upward)

t key Toggle layer visibility for all layers

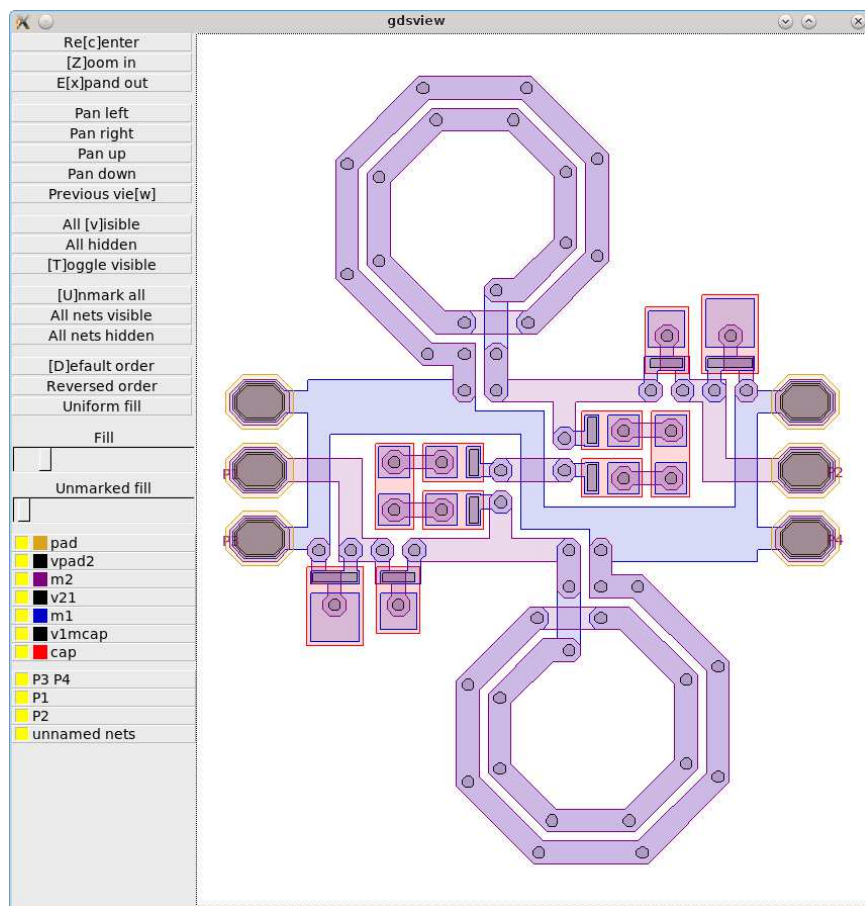


Figure 10: GDSview (OpenGL version) layout window

m key Mark the pointed-to net (unmarked nets are shown dimly). You can mark multiple nets.

u key Unmark the pointed-to net. If the mouse is not pointing within any layers, then all nets are unmarked.

n key Hide the pointed-to net.

In the sidebar there are buttons that can be used for the navigation functions discussed earlier. Buttons labeled with a letter in brackets have the same effect as pressing the corresponding key in the layout display. For example, the **Re[c]enter** button has the same effect as pressing the **c** key in the layout display. The following buttons have no key equivalents:

All hidden Make all layers invisible

Reversed order Make the display order the reverse of the order in the process file

Uniform fill Use the same opacity value for filling all the layers (the value is set by the fill slider)

All nets visible Make all nets visible. (This is independent of layer visibility, so if the layers for a net are all hidden, you won't see the net.)

All nets hidden Make all nets invisible. Things that aren't part of any net (like device boundaries and labels that are not close to any edge) will still be visible.

The **Fill** slider adjusts the opacity of the fill for layer interiors. Move the slider all the way to the left for full transparency and all the way to the right for full opacity. If you've marked some nets, then their opacity is what the fill slider controls. The opacity for the remaining unmarked nets is set by the **Unmarked fill** slider.

The buttons showing layer names can be left-clicked to toggle the display of individual layers on and off. Right-clicking on a layer name brings up a menu where the layer can be raised or lowered in the display order, and where the opacity of the layer can be set (independent of the fill slider value). Keyboard shortcuts for the menu items are available and are shown in brackets in the item labels. The shortcut is available when the button with the layer name has keyboard focus (typically because you're pointing at it with the mouse).

Similarly, the buttons for net names can be used to control net visibility, and right-clicking will show a menu with net operations. All the nets that don't have any ports (or device terminals) are grouped together and controlled by the button labeled **unnamed nets**.

Also note that as you move the mouse pointer around the layout window, the names for any layers that are under the pointer are highlighted in red. Actions like hiding (**h**), raising (**r**), setting layer opacity (0, 1, ...), etc., normally act on the layer whose name is displayed (because the mouse is pointing at an edge

of that layer). However, if no name is displayed (i.e., the pointer is not close to any layer edge), then the action is instead applied to the layer that the pointer is within. If the pointer is within multiple layers, then the one that is topmost in the display order is the one that is affected. (If no name is displayed and the pointer is not within any layer, then the action has no effect.) Nets that the pointer is within are also shown in red. Keystrokes normally operate on the net whose name is displayed. However, if no name is displayed but the pointer is within a *unique* net, then that net is the one affected.

5 Acknowledgements

Robust geometric predicates are by Jonathan R. Shewchuk.

Routines for Arbitrary Precision Floating-point Arithmetic and Fast
Robust Geometric Predicates
(predicates.c)

May 18, 1996

Placed in the public domain by
Jonathan Richard Shewchuk
School of Computer Science
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, Pennsylvania 15213-3891
jrs@cs.cmu.edu

Regular expression support is provided by the PCRE library package, which is open source software, written by Philip Hazel, and copyright by the University of Cambridge, England. Source code for the PCRE library may be obtained from:

<ftp://ftp.csx.cam.ac.uk/pub/software/programming/pcre/>

The PCRE library is available under a BSD license:

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the University of Cambridge nor the name of Google Inc. nor the names of their contributors may be used to endorse or promote products derived from this software without specific prior written permission.

This software is provided by the copyright holders and contributors “as is” and any express or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall the copyright owner or contributors be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage.

The user interface for the OpenGL version of GDSview is based in part on the work of the FLTK project (<http://www.fltk.org>).