

gEDA gschem Users Guide

Ales Hvezda

This document is released under GFDL
(<http://www.gnu.org/copyleft/fdl.html>)

September 21st, 2003

Contents

1 Overview	3
2 Introduction	3
3 Overview	3
4 Running gschem	3
5 User interface	4
6 Basic Operations	5
6.1 File related operations:	5
6.2 Addition of objects:	7
6.3 Manipulate objects:	10
6.4 View operations (Zoom/Pan/Redraw etc...)	15
6.5 Hierarchy operations (...)	17
6.6 Attribute operations (Attach/Detach/Toggle visibility etc...)	18
6.7 Buffer operations	19
6.8 Option changing (text size/gridding/snap etc...)	20
6.9 Grips	21
7 Files used/created by gschem	21
8 Electrical connectivity	22
9 Dialog boxes	23
10 Components/symbols	23
11 Attributes	23
12 Resource File	25

1 Overview

2 Introduction

This is a very basic and crude first attempt at getting some real user docs written for `gschem`. This document is hardly finished or polished but it is better than having no docs at all (which was the case for the longest time). Please forgive the tone of this document; I (Ales) am writing this just to get something done. I am by no means a technical writer so I am sure some of my sentences are truly awful. As everybody says, programmers should not write documentation for their programs and this document is a perfect illustration of a programmer breaking that rule. Please report any errors/inconsistencies to ahvezda@geda.seul.org.

This document assumes you understand basic schematic capture concepts. For example that a component represents something and that nets and buses interconnect these components to form a schematic etc... Eventually this document will describe all this in more detail, but for now this document is will describe how schematic capture is accomplished in the gEDA system.

3 Overview

`gschem` is the schematic capture program in the gEDA tool suite. It's sole purpose is facilitate the input of circuit schematics or block diagrams graphically. `gschem` is also the symbol editor for the gEDA system. This is possible because the schematic format and symbol format are exactly the same.

4 Running `gschem`

Running `gschem` is straightforward once you have installed it on your GNU/Linux or UNIX system.

The first step is to build and install the gEDA tools. This step is not covered in this document. You can tell if you have installed the tools correctly by running the following commands:

```
which gschem
ldd 'which gschem'
```

The first command should return the path to the `gschem` binary and the second command (only on unix-like operating systems which include the `ldd` utility for listing dynamic dependencies of executibles or shared objects) will return

which libraries are linked to `gschem`; all of the request libraries must be found for `gschem` to run. If these commands do not return the expected results, then most likely the gEDA tools are not installed properly. Please see the appropriate INSTALL docs (which came with the distribution) for more info on installing the gEDA tools.

To run `gschem` execute “`gschem`” at any shell prompt (without the quote marks). This will start the `gschem` program with a blank schematic. To look at an example schematic which came with `gschem` run “`gschem schematic_name.sch`” For a listing of the various command line flags run “`gschem -h`”. For a detailed explanation of the command line flags look at the `gschem` man page (“`man gschem`”).

5 User interface

There are several ways to interact with `gschem`. `gschem` requires a keyboard and mouse. There are three ways to initiate an operation or command:

- Using the mouse to select the operation off a menu
- Typing the keyboard shortcut(s).
- Draw the appropriate stroke (if stroke support is enabled)

To make usage matters more confusing, selecting an operation off of the menus behaves slightly differently than typing the keyboard shortcut. Most of the operations operate on the currently selected object(s), hence you need to select the object first before manipulating them. The menu selected operations usually require some more input (usually a mouse click) after they are picked off of the menu. The keyboard shortcut operations take that required input as the current mouse position. This saves an extra click since you can position the mouse at the right place, type in the shortcut(s), and the command then executes. Note, you can change this so that both menu and shortcut behavior is exactly the same. See the section on the resource file for more info on how to configure this.

Most of the interaction with `gschem` is fairly mode oriented (similar to the great text editor `vi`). If you select operations off of the menu, then you are placed into the corresponding mode (like copy or move mode). You must then select an anchor point (or whatever the appropriate point is) to continue the operation. Most of the commands off of the menu expect the objects to be already selected. Some of the modes persist after being execute while other immediately return you into select mode (the default mode).

The shortcuts are also mode like in nature. Most of the default shortcuts are for the various commands are not single keystrokes. There are a few which are single keystrokes (like zoom in: ‘z’ or pan: ‘x’), but most are typically two keystrokes long. As examples, to execute File/Save you would type ‘f’ and ‘s’

(without the quotes) or Add/Line is 'a' and 'l'. You can get a listing of the shortcuts by picking Help/Hotkeys. You can also see the hotkey assignments in the pulldown menus as well. The shortcuts are defined in the resource files (`system-gschemrc`, `/.gEDA/gschemrc`, or `'pwd'/gschemrc`). See the section on the resource file for more info.

The mouse button actions in gschem are mostly configurable. The first mouse button is always used to select objects or pick points. This button is not configurable. The second mouse button is either a copy/move action (when held down over an object), a repeat last command or used to draw a stroke to execute a command. The third mouse button is either a mouse pan (when held down as the mouse is moved) or a popup menu. The behavior of the second and third mouse buttons is controlled through the resource file (see the section below for more info).

6 Basic Operations

There are nine basic kinds of operations in gschem:

- File related (Open/Close/Save/Print etc...)
- Addition of objects (Add nets/lines/pins/components etc...)
- Manipulate objects (move/copy/delete/rotate/mirror etc...)
- View operations (Zoom/Pan/Redraw etc...)
- Hierarchy operations (Down Schematic/Down Symbol/Up)
- Attribute operations (Attach/Detach/Toggle visibility etc...)
- Buffer operations (Copy / Cut / Paste)
- Option changing (text size/gridding/snap etc...)
- Editing Grips

This section will describe each of these. The terms “page” and “schematic” both refer to the same thing (a collection of objects that are loaded and being displayed). A window is that big window in which schematics are displayed and manipulated.

6.1 File related operations:

- **New Window** Open a new window. Each window is totally separate from all other windows.

- **New Page** Open a new page. Usually this page will be called untitled_N.sch, where N is an incrementing number.
- **Open Page...** Open an page from disk. This will pop up a dialog box. Please see the later section on how to use the more advance features of this dialog box.
- **Close Page** Close the currently displayed page. This will prompt you to save if you have modified the page.
- **Revert Page** Close and reopen the currently displayed page. This will not prompt you to save the current page, but will quickly discard any changes you have made and reopen the saved schematic from disk. Use with caution.
- **Save Page** Save the page current page. If the page is called untitled_N.sch (where N is a integer), then a Save As... dialog box will appear.
- **Save Page as...** Opens the Save As.. dialog box. Clicking Save As will save the currently displayed schematic. The displayed schematic will be called the new specified name.
- **Save All** Unconditionally saves all schematics loaded in memory.
- **Print...** Brings up the Print dialog box. Please see the dialog box section for more info on this.
- **Write PNG...** Brings up the Write PNG dialog box. Please see the dialog box section for more info on this. Note you must have libgdgeda install (and any required dependencies) if you want to output images.
- **Close Window** Closes the current window. If there are any modified schematics, a "Are you sure" dialog box will appear. Clicking OK will cause all unsaved schematics to be lost.
- **Quit** Like Close window, but close all opened window. A "Are you sure" dialog box will appear for each window which has unsaved schematics
- **Page Manager** Brings up a dialog box which lets you pick pages as well as see various info about the pages loaded in memory. Please see the dialog box section for more info on this.
- **Page Next**
Page Prev These two options allow you to move in between opened pages.
- **Page Discard** Closes the current schematic forcefully. If the current schematic is NOT saved then the data is lost forever. You will NOT be prompted or warned if you have modified the current displayed schematic. Use with caution. Useful if you want to unload a schematic quickly.

6.2 Addition of objects:

An object is one of the following: line, circle, pin, net, box, bus, text/attribute, and component. A component is a collection of lines, circles, pins, boxes, text, and attributes.

- **Add Component** Opens up a dialog box which lets you place components from the component libraries. To place a component do the following:
 1. Select a Component library (which, btw, is specified in the Resource files, see section below) from the left list
 2. Select a Component from the right list.
 3. Move the mouse into the main drawing window (you should see an outline).
 4. Press the first mouse button to place the component.
 5. Keep pressing the first mouse button to place additional instances of the component.

If a component name is already selected, hitting apply and moving the mouse into the main window will allow you to place that component again.

You can rotate the component before you place it by clicking the middle button. For every button click, the component will be rotate 90 degrees.

To cancel a component place press the last mouse button or the ESC key. Please see the dialog box section for detailed information on this dialog box.

- **Add Net** Draws a net segment. If you select this off of the top menu bar then:
 1. Press the first mouse button to start the net.
 2. Press the first mouse button to end the net.
 3. Another net will start at the last endpoint.
 4. Press the first mouse button to end that net etc...

Press the last mouse button or ESC to cancel any net in progress.

If you started this using the keyboard shortcut then the net start immediately at the last mouse position and then it behave exactly as above (except for line #1).

After drawing a net segment (or segments) and canceling the last net segment, you are automatically placed in select mode. You must pick add net again or type the shortcut to add more nets.

You can hold down the control key to draw non-orthogonal nets. Just keep in mind that you cannot connect anything to the middle of a non-orthogonal net.

The boxes at the end of the nets are connectivity cues. Filled boxes signified a dangling net (not connected to anything). Filled circles are midpoint connections/junctions. These cues are drawn automatically and are an indicator of electrical connectivity.

See the section on electrical connectivity below for more information.

- **Add Bus** Basically the same thing as nets, except that it draws buses. Buses are very new and there are many aspects which are not defined yet, so keep that in mind as you uses buses. *More to be added here eventually*
- **Add Attribute...** Brings up the add attribute dialog box. This dialog box is ONLY used to add attributes. It does not display or manipulate already placed attributes.

An attribute is nothing more than a text item which is in the form **name=value** (there cannot be any spaced to the left or right of the name,value pair). It can be either unattached or attached.

To add an unattached attribute do the following:

1. Select an attribute name off of the pulldown list –or– type in the attribute name into the name entry.
2. Type in a value for the attribute.
3. Pick any of the attribute options.
4. Click Apply and the attribute will be placed.

If you want to attach an attribute to an object then select the desired object first and then press the add attribute apply button. The text should be yellow which signifies an attached attribute. If you click on an object which has attached attributes, the attached attributes should be selected as well.

If you select Add/Attribute... off of the pull down menus then you do not have much control as to where the attribute gets placed (it gets places either at the lower left hand corner of the object extents or at the origin of any selected object). However, if you execute Add/Attribute using the hot key then the current mouse position is used as the anchor point for the attribute item.

You cannot place an incomplete attribute (an attribute without a name and value).

Please see the section on attributes below for more info on how to use attributes and more details.

- **Add Text** Opens up the text add dialog box. To place text:
 1. Type in text in entry field

2. Hit enter or click Apply
3. Move the mouse into the main window (an outline of the text should appear and follow the mouse)
4. Press the first mouse button to place the text

If you leave the text add dialog box open you can place the same text item again and again by just clicking apply (or pressing enter) and moving the mouse into the main window.

Text which is placed will be automatically capitalized. Please see the Resource file section below on how to control this behavior.

To cancel a text place press the last mouse button or the ESC key.

If you create text in the form `name=value`, then you are creating attributes. gEDA allows for general attributes to be free floating (or unattached). It is a good idea to change the color of these floating attributes to the current attribute color (which is also called the attached attribute color) to signify that this text item is an attribute.

You can rotate the text before you place it by clicking the middle button. For every button click, the text will be rotate 90 degrees.

- **Add Line** Draws a line in the same fashion as drawing nets with the following exceptions:
 - A line has no electrical significance
 - Only a single line is drawn
 - You keep drawing lines as long as you are in line drawing mode.

To cancel a line rubberband in progress press the last mouse button or the ESC key.

- **Add Circle** To draw a circle (picking Add Circle off of a menu):

1. Pick the center of the circle with the first button
2. Move the mouse to see an outline of the circle
3. Press the first mouse button to finalize the circle

To draw a circle (typing the shortcut), same as above except that the center of the circle is picked for you at the last mouse position when you type the shortcut.

To cancel a circle rubberband in progress press the last mouse button or the ESC key.

- **Add Arc** To draw an arc (picking Add Arc off of a menu):

1. Pick the center of the arc with first mouse button

2. Move the mouse and pick the next point. The rubberbanded line represents the radius
3. A dialog box will appear: specify the start angle (in degrees) and specify the degrees of sweep.
4. Press OK to finalize the values.

The start angle can be positive or negative. The degrees are specified using the standard Cartesian coordinate system. The degrees of sweep can be positive or negative.

To cancel an arc in progress (while rubberband the radius) press the last mouse button or the ESC key or press the Cancel button in the arc dialog box.

- **Add Pin** Draws a pin in the same fashion as drawing nets with the following exception:
 - You keep drawing pins as long as you are in pins drawing mode.

To cancel a pin rubberband in progress press the last mouse button or the ESC key.

6.3 Manipulate objects:

Most of these operations are modes which operate on the current selected objects. Before you can execute the desired function you must select the objects you want to rotate.

- **Undo/Redo** Undo does exactly that, it undos the last action which changed the schematic.

Here is how gschem implements undo: Basically after every action (including zooming and panning) the schematic is saved to disk (in /tmp). Gschem do clean up after itself when you exit. Should gschem crash, the saved files are left alone in /tmp.

Redo only applies when you do an Undo. You can undo something and then immediately redo it. However if you do anything in between you will lose the undo info. You can undo and redo to your hearts desire up and down till you reach the max undo levels.

You can change the behavior of undo and redo by changing the defaults in one of the *gschemrc files. Please see the system-gschemrc for more info.

- **Select Mode** Select mode is the initial mode gschem starts up in.

In select mode you can select objects.

You can pick single objects by clicking on them. If an object is already selected than clicking on that object will keep it selected. If multiple objects overlap then clicking in one spot will cycle through the objects.

If you hold down the SHIFT key and click, you can select and deselect multiple objects. Doing this with multiple overlapping objects will cause the selection to cycle among the possible object selections.

If you hold down the CONTROL key and click, you will toggle the object in and out of the current selection list.

To select multiple objects press and hold the first mouse button and drag the mouse till a selection box appears. Drag the mouse (selection box) till it encompasses the objects and release the first mouse button. The objects must be completely encompassed. Using the selection box takes some practice. If any previous objects were selected, they will be unselected.

If you hold down the SHIFT key while drawing a selection box then you will add to the currently selected objects. Objects cannot be removed using the selection box and holding down the SHIFT key.

If you hold down the CONTROL key while drawing a selection box then you will toggle any encompassed objects. If an object was selected then it will be unselected and vice versa.

If you pick an object which has attributes which are attached to it, then these attributes will be selected as well. If you just want to select the object, you must deselect the attributes. Invisible attached attributes are also selected when you pick the object This behavior is handy if you want to manipulate/change an object and all of its attributes (moving/copying the object around).

The selection mechanisms are not obvious and do require some practice. There are some quirks so please report them as you come across them.

- **Edit...** Allows you to edit: Text - brings up a dialog box which allows you to edit the text string and size

Attributes - a much more sophisticated version of above specifically for attributes.

All attributes attached to a component - brings up a multi-attribute editing dialog box.

1. Select the item (text/component) to edit.
2. Pick or type the shortcut for Edit/Edit...
3. Make the appropriate change and press OK

Note, only the item will be edited; bulk text/component editing is not supported (yet).

- **Edit Text...** Allows you to edit text only, regardless if that text is an attribute or just a plain text string.

1. Select the text item text to edit.
2. Pick or type the shortcut for Edit/Edit Text...

3. Make the appropriate change and press OK

Note, only the item will be edited; bulk text editing is not supported (yet).

- **Copy Mode** This mode allows you to copy the currently selected objects.

To copy objects (picking Edit/Copy Mode off of a menu):

1. Select the desired object(s)
2. Pick Edit/Copy Mode off of a menu
3. Pick an origin point for the copy (first button)
4. Move the outline to the desired position
5. Pick the destination point (first button)

To copy objects using the shortcut for copy mode is almost the same as above except that the origin point is selected automatically for you once you hit the copy mode shortcut.

After finishing the copy, you are automatically put back into to select mode.

Holding down the CONTROL key as you move the outline around will constrain the movement to be either horizontal or vertical.

- **Move Mode** Move mode is just like copy (above) except that instead of copying you are actually moving the objects around.
- **Delete** Delete allows you to remove objects off of the page.

To delete objects:

1. Pick the desired object(s).
2. Select or type the shortcut for Edit/Delete
3. Objects will be deleted immediately

Use this with caution since there is NO undo yet!

- **Rotate 90 Mode** Rotate mode allows you to rotate objects 90 degrees around a pivot/center point.

To rotate objects (picking Edit/Rotate 90 Mode off of a menu):

1. Pick the desired object(s).
2. Select Edit/Rotate 90 Mode off of a menu
3. Pick the pivot or center point of the rotate

Rotating objects using the shortcut is similar to above except that the center point is the last mouse position at which you typed the shortcut.

The objects will be rotate 90 counter clockwise. You can keep rotate objects 90 degrees till you have them in the wanted rotation. Text will always appear upright.

- **Mirror Mode** Mirror mode allows you to mirror objects horizontally around pivot.

To rotate objects (picking Edit/Rotate 90 Mode off of a menu):

1. Pick the desired object(s).
2. Select Edit/Mirror Mode off of a menu
3. Pick the pivot point of the mirror

Mirroring objects using the shortcut is similar to above except that the pivot point is the last mouse position at which you typed the shortcut.

Objects are mirrored horizontally about the pivot point. If you want to get a vertical mirror then rotate and mirror the object(s) till you get the desired position.

Mirroring of embedded components is not supported.

- **Slot...** Slot... allows you to change the slot of a multiple slot component. The component must support slotting (see the Components/Symbols section for more info on this).

To change the slot:

1. Select the desired component.
2. Pick or type the shortcut for Edit/Slot...
3. Type in the new slot number
4. Press OK.

Slotting is still in development so expect some quirks.

- **Color...** This option allows you to change the color of any selected object (with the exception of components).

To change the color of the currently selected objects:

1. Select the desired object(s).
2. Pick or type the shortcut for Edit/Color...
3. A dialog box with a drop down menu will appear.
4. Pick the new color
5. Press Apply.

The color change will only take effect once you press Apply. The dialog box will not change to the color of the current object(s) (ie it only lets you change the color and does not get the current color of selected object(s)).

You can leave this dialog box up and select new objects and change their color by pressing Apply.

- **Lock**

Unlock Lock and unlock allow you to lock/unlock components in a schematic. Locking means that a component cannot be selected by a single click. This is useful for title blocks and other components which should not be selectable because you might have other objects inside its boundaries and clicking inside the component (titleblock) would be a distraction.

To lock/unlock components:

1. Select the desired object(s)
2. Pick or type the hot key for lock/unlock

The components will not be selectable with a single click. To select a locked component, use a selection box to select it. This is the only way to select a locked component.

You can lock and unlock regular objects (lines/pins/boxes...) which is nice when you are drawing something and an object is in the way continuously. Just lock it and you will not have to think about it when you click to select other objects. However, locking an object is not preserved in the file format so once you quit any locked objects (lines/pins/boxes...) will become unlocked. The locked/unlocked state of components is preserved in the file format though.

- **Line Width & Type...** This dialog box lets you control the width and type of lines, boxes, circles, and arcs. This dialog box is still under development.

Select an object and then select this option will bring up a dialog box which lets you set the line width, line type, line dash length, and line dash spacing. Pressing OK will apply the changes.

- **Symbol Translate...** Symbol translate is used to take whatever is drawn and translate it around using a inputed value (in mils). This operation is mainly used to translate symbols around (mainly to the origin).

To translate a symbol, just select or type the shortcut for Edit/Symbol Translate... and a dialog box will appear. Enter the amount you want translate (positive or negative) in the X and Y directions (the same value will be applied to both directions).

If you enter a 0, then all the objects will be translated to the origin. If you are drawing a symbol, zoom in a bit and then execute this (0 for the amount), since there are still some bugs in this operation. It is a requirement that snap be on and that the grid snap spacing is set 100 mils when creating symbols.

- **Embed Component**

Unembed Component gschem supports a concept of embedded components which components in a schematics which do not require an external

symbol file of any sort. All the information necessary to display a component is placed in the schematic file. This causes schematic files to be significantly larger, but it makes it easy to share schematics with other people or archive schematics away, since you have any dependencies on symbol files. You should only embed components when absolutely necessary.

Embed Component and Unembed component work exactly alike:

1. Select any components you wish to (un)embed
2. Pick or type the shortcut for Edit/(Un)Embed Component

If you want to unembed a component you must have a symbol with the same name in the component library search path (otherwise the unembed will not work).

You can only embed and unembed components. Also, you cannot embed and then mirror a component (this is a limitation of gschem and will eventually be fixed).

You can also place embedded components directly in the Add Component... dialog box.

- **Show Hidden Text** This operation makes all hidden/invisible text visible. To use it, simply pick the option (or type its shortcut) and all the invisible text will appear. Visible text is unaffected by this operation.

This operation is useful when drawing/debugging symbols.

When hidden text is visible, the text "Show Hidden" will appear in the lower, righthand corner.

6.4 View operations (Zoom/Pan/Redraw etc...)

- **Redraw** This option redraws the current display. This is useful when you have mouse/component/line/text etc... droppings left over from a previous action. It is also useful when you want to update all visual connectivity cues.

- **Pan** Pan lets you move around the display.

To pan the display (picking View/Pan off of a menu):

1. Select View/Pan off the menu
2. Click the first mouse button at the new center of the display.

To pan the display using the shortcut is much simpler, simply type the shortcut and the display will pan the display to the current mouse location.

You can also enable mouse panning if you add:

```
(third-button "mousepan")
```

to one of the gschemrc files. See the section on the resource file for more info. Mouse panning is very nice for small schematics, but tends to lag on larger ones.

- **Zoom box** Zoom box allows you to specify a zoom window for zooming in.

To use the zoom box (picking View/Zoom box off of a menu):

1. Select View/Zoom box off the menu
2. Click and hold the first mouse button
3. Drag the mouse drawing the zoom box around the area you want to zoom
4. Release the mouse button and the display will zoom

To use the zoom box by typing the shortcut is similar. Once you type the shortcut, the zoom box will start immediately using the current mouse location as the first corner of the box.

Zoom box will attempt to zoom the requested area, but some boxes are not legal and gschem will do it's best to zoom the requested area.

- **Zoom limits** Zoom limits will zoom the display attempting to fit all the placed objects on the screen.

Simply pick View/Zoom limits off of a menu or type its shortcut and the display will be redrawn.

There are the special cases (like a single horizontal line) which sometimes do not display correctly (or as expected). Hopefully these exceptions will be fixed someday.

- **Zoom In**

Zoom out Zoom In/Out zoom the display using the picked mouse location as the center of the display (or the current mouse location if using the shortcut). These commands always zoom in/out by a factor.

To use Zoom In/Out (picking View/Zoom box off of a menu):

1. Pick Zoom In/Out off of the menu
2. Click the first mouse button as the center of the zoom

Using the shortcut is similar except that the current mouse position serves as the center of the new display.

- **Zoom full** Zoom full will zoom the display to the maximum possible displayable view.

Simple select it off of a menu or type it's shortcut and the display will be zoomed.

6.5 Hierarchy operations (...)

- **Down Schematic** Go down into a symbol, opening up any underlying schematics. Basically this will open up an underlying schematic of the selected component if it exists in the source library search path. See the Resource File section on how to define this path.

There are currently two ways of specifying that a symbol has an underlying schematic or schematics:

1. The underlying schematic must have the same name as the symbol but have a `.sch` extension and must follow the `_{#}` suffix naming convention. See the Files section below on this convention.
2. Attach an attribute to the symbol called `source=filename.sch` filename.sch is not a path to the symbol, but rather the basename (last file in the path specifier) of the symbol path. The underlying schematic will still be searched in the source-library path. You can specify multiple `source=` attributes. The underlying schematics will be opened in the order that the `source=` attribute is found.

If there multiple underlying schematics, they will be loaded. Movement between the schematic pages is restricted (to the same level of the same set of underlying schematics) unless the `rc` keyword `enforce-hierarchy` is modified to allow for a freer hierarchy traversal mode. See the Resource File section for more info.

It is also recommend that you maintain unique names for the various levels (when using the `source=` attribute) to avoid possible confusion. The hierarchy mechanisms are fairly new so expect some odd behavior (and please report it)

- **Down Symbol** This option will open up the symbol of the selected component.

Once the symbol is open, the user can edit it and save it.

At this time, the toplevel schematic will not see the symbol change unless the toplevel schematic is reloaded or `File/Revert` is executed. This will be fixed eventually.

- **Up** This option will move up the hierarchy (if there are pages above the currently displayed page).
- **Documentation** Open any documentation available for the selected symbol/component.

The job is handed over to "gschemdoc", which makes a best-effort attempt of finding relevant documentation.

The documentation would normally be in PDF, HTML, text or image format, but gschemdoc tries to be as transparent as possible on this account.

First and foremost, the attribute "documentation=" is assumed to point to the documentation. This attribute should either be the filename (base-name) of the document, or it should be a complete URL.

If it is a filename, and the file is found locally (in /usr/share/gEDA/documentation or otherwise), the relevant viewer will be initiated. Otherwise, a Google search for the document will be initiated.

If there is no documentation attribute, the attributes "device" and possibly "value" will be consulted in much the same way as for "documentation". File searches will be made in forms of filenames like "device-value.pdf" and "device.pdf".

Failing that, the file name for the symbol itself will be used as basis for the search.

6.6 Attribute operations (Attach/Detach/Toggle visibility etc...)

An attribute is nothing more than a text item which is in the form `name=value`. It can be either unattached or attached.

The operations in this group manipulate attributes only. Most of these operations have no effect on plain text objects.

- **Attach** The Attach command allows you to take a text item (in the proper form; `name=value`) and attach it to another object.

To use Attributes/Attach:

1. Select the object which will receive the attributes
2. Select the text object(s) which will be attached to the above object
3. Pick or type the shortcut for Attributes/Attach

The order of the sequence of selecting the object and then the text items is important; gschem will not allow you to select the text items first and then the object. After going through the above sequence the text item will turn yellow (or the current attached attribute color) signifying that the text item is an attached attribute.

You cannot attach a single attribute to several different objects. You cannot attach non-text items as attributes.

- **Detach** Detach allows you to deassociate attributes from objects.

To deselect an object of all attributes:

1. Select the object of interest
2. Pick or type the shortcut for Attributes/Detach

All the attached attributes (even if they are not selected) will be detached from the object. This behavior is probably broken and will eventually be fixed (so that only selected attributes are detached).

When you detach attributes then they turn red (or the current detached attribute color). This color changes allows you to spot text which was an attribute and is now dangling (unattached).

- **Show Value**

- **Show Name**

- **Show Both** These operations allow you to control which part of the attribute string is visible. Usually you are just interested in seeing the value of the attribute, but there are circumstances where seeing the name and value (or maybe just the name) would be useful.

- To use the options:

- 1. Select the attribute(s) of interest
 2. Pick or type the shortcut for Attributes/Show *

- The text item(s) should immediately change.

- These operations only work on text items which are in the form **name=value**

- **Toggle Vis** This operation allows you to toggle the visibility of attributes.

- To use this option:

- 1. Select the text item(s) of interest
 2. Pick or type the shortcut for Attributes/Toggle Vis

- The text item(s) should change their visibility immediately.

- If you make an attached attribute invisible, then you can simply select the parent object and select Toggle Vis and the attribute will be come visible (likewise any visible attributes attached to that object will become invisible).

- If you make a free floating (unattached) attribute invisible, then the only way to make it visible (and all other invisible attributes) is to use the Edit/Show Hidden Text option.

6.7 Buffer operations

Gschem supports 5 copy/cut/paste buffers which are visible across all opened pages and windows.

- **Copy** To copy something into a buffer:

- 1. Select the objects you want to copy.
 2. Select Buffer/Copy/Copy into buffer #.

- **Cut** Cut is like copy in that it removes the objects from the schematic
- **Paste** To paste a buffer into the current schematic:
 1. Fill the buffer using the above Copy or Cut.
 2. Go to the new schematic page/window.
 3. Select Buffer/Paste/Paste from buffer #.
 4. Click the first mouse button to pick an anchor point.
 5. Move the mouse to the final spot.
 6. Click the first mouse button again.

6.8 Option changing (text size/gridding/snap etc...)

- **Text Size...** This command pops up a dialog box which allows you to specify the text size of all text (including attributes placed with the Add/Attribute... dialog box).
The text size is in points (1/72"). The default text size is 10 point text. The smallest text size is 2 points.
- **Toggle Grid** Toggles the visible grid
- **Toggle Snap** Toggles the snap. Be very careful using this. Connections between pins and nets (and nets to nets) depends on being exactly connected. Turning of the grid will almost guarantee that nets/pins do not connect.
Before you translate a symbol using Edit/Symbol Translate, make sure the snap is on.
When snap mode is off, the text "Snap Off" will appear in the lower, righthand corner.
- **Snap Grid Spacing...** This option brings up a dialog box which allows you to change the snap grid spacing (not the grid spacing). The units for this spacing are mils.
Before you translate a symbol using Edit/Symbol Translate, make sure this spacing is set to 100.
- **Toggle Outline** Toggles between drawing the outline of the current selection or just drawing a box when doing moves/copies/component and text places. The outline mode looks better, but tends to be significantly slower than using the box (bounding box) mode.
- **Show Log Window** This option displays the log window if it has been closed or disabled from being displayed when you start up gschem.
- **Show Coord Window** This option displays a dialog box which displays the current x, y location of the mouse pointer in screen (pixels) and world coordinates (mils)

6.9 Grips

Grips are a mechanism used in gschem to provide an easy way of modifying objects inside schematics. When you select an object, little squares are placed in strategic locations (line end points or circle radius point or corners of a box) which allow you to change the object quickly. Grip support currently exists for lines, nets, pins, buses, circles, and boxes. Arcs do not yet have grips, but will eventually have them.

Using grips is easy:

1. Select the object you want to change. The grips (the little boxes) will appear.
2. Click and hold the first mouse button inside the box.
3. Move the mouse around till you have the object where you want it
4. Release the mouse button.

7 Files used/created by gschem

There are several files which gschem uses. Here is a list and a brief explanation of each:

- ***.sch** Schematic files. These files contain components, nets, text, and sometimes primitive objects (like lines, circles, box etc...) Schematics do not contain pins. Schematic filenames should follow this convention: **name_#.sch** where:
 - **name** is a text string which describes what this schematic contains.
 - **_#** is an underscore and a number (like **_1**, **_2**, **_7**, **_13**, etc...) This number is used to sequence schematic pages in a multiple page schematic.
 - **.sch** is the schematic extension/suffix. It is important the schematic pages have this extension.

Schematic files are pure ASCII and will always be pure ASCII. gEDA does not support any binary file formats. The file format for schematics is described in the gEDA file formats document.

- ***.sym** Symbol files. The schematic and symbol file formats are identical. gschem (or a text editor) is used to create symbol files as well as schematics. Symbol files contain lines, circles, boxes, arcs, pins, text, and attributes.

The naming convention for symbol files is: **name-#.sym** where:

- **name** is a text string which describes what the symbol represents.

- `-#` is a dash and a number (like `-1`, `-2` etc...) The number is used to allow for a symbols to have the same name yet different contents. There might be multiple representations for resistors so these symbols should be called: `resistor-1.sym`, `resistor-2.sym`, and `resistor-3.sym`.
- `.sym` is the symbol extension/suffix. It is important the symbols have this extension.

The way of specifying hierarchy is by using the `source= attribute`. Please see the master attribute document for info on this mechanism.

The hierarchy mechanism is still in heavy flux, so there might be some more changes.

- `system-gschemrc`
`/.gEDA/gschemrc`
`gschemrc` Resource files. These contain guile (scheme) keywords which control the various configurable parameters. All three files may exist and are searched in the above listed order. `system-gschemrc` contains system wide defaults. `/.gEDA/gschemrc` lives in each users home directory and contains project wide defaults. `gschemrc` lives in each project directory and contains project specific defaults. See the section below on the resource file for more info.
- `gschem.log` Log file. This file contains informative, error, warnings etc... messages when `gschem` was run. This file is created in the working directory that `gschem` was started in. This allows the user to preserve log files between independent projects.

TBA: Suggested format for projects

8 Electrical connectivity

As you draw schematics you need be aware of what is considered to be electrically connected by the `gEDA` programs.

Nets which are visually connected to other nets are electrically connected. This connection may be endpoint to endpoint or endpoint to midpoint. When a single endpoint to endpoint (net or pin endpoint) connection is drawn then the visual dangling net cue disappears. When an endpoint ends in the middle of another net (or multiple endpoints coming together at a single point) then a circular filled connectivity cue is drawn. You cannot connect a net to the middle of a pin. Nets can only be connected to the endpoints of pins. You cannot connect to a net if that net is not orthogonal (horizontal or vertical). The visual cues are the primary way of telling if nets/pins are connected.

Bus are similar to nets with the exception that you cannot connect a net to the endpoint of a bus (only to the middle). If you do try to connect a net to the

end of a bus you will see a big red X at the invalid endpoint connection. Buses are still very new so there are still many quirks.

You can label nets by using the `label=` attribute. Do not attach more than one `label=` to a net. You only need to attach the `label=` attribute to one net segment. Different nets (i.e. multiple net segments which aren't connected together) which have the same attribute `label=` attached to them are also considered electrically connected. You will not get any indication of this connection by `gschem`, but the `netlister` (`gnetlist`) considers nets with the same `label=` attribute electrically connected. The naming convention for buses has not been formalized yet.

More TBA

9 Dialog boxes

TBA

10 Components/symbols

A component or symbol represents something. Usually it represents a gate, a black box or block, or an entire device. A symbol is a collection of primitives which are grouped together. You can use lines, boxes, circles, arcs, text, attributes and pins as the primitives for building symbols. You cannot have nets, buses, or other symbols inside a component.

Components are searched for by specifying (component-library "...") inside one of the `*rc` files. See below for more info.

More to TBA

11 Attributes

An attribute is nothing more than a text item which is in the form `name=value` (there cannot be any spaced to the left or right of the name,value pair). It can be either unattached or attached. Attributes are used extensively in the `gEDA` project to convey information. Things like device name, pin numbers, hidden nets, and unit reference numbers are specified using attributes. For a list of attributes be sure to look at the `attributes.txt` document.

There are two kinds of attributes:

1. Regular attached attributes. These are attributes which take on the standard form and are attached to some object (pin, net, component, or box etc...). These attributes are usually yellow in color.

2. Unattached attributes. These are attributes which take on the standard form, but are not attached to any object. These attributes are also known as floating or toplevel attributes.

Regular attached attributes are attached to an object to associate the info with a specific object. For example: a pin number associated with a pin. Unattached attributes usually convey some information which is global in nature. For example: a `device=` attribute (which lives inside symbols) and specifies what device the entire symbol represents.

There is a third type of attribute which is a special case of #2 but turns into #1. This special type of attribute is known as a prompted attribute. If you place an unattached visible attribute inside a symbol and then instantiate that symbol, then that unattached attribute gets "promoted" to an attached attribute. This newly promoted attribute gets attached to the symbol. This mechanism of attribute reattachment (from within a symbol) is known as attribute promotion.

There are some gotchas about attribute promotion:

- Promotion **only** happens when the symbol is placed. That means that if you place a symbol (`sym1`) and then change it on disk (by adding or removing floating attributes), existing `sym1`'s will not reflect these new floating attributes (ie they won't be promoted) in any schematic.
- The `device=` attribute is not promoted.
- Invisible attributes are not promoted by default. If you attach a floating attribute (like `numsLots=#`) and make it invisible, it will not be promoted. Now, in order to make everybody happy, this behavior is configurable. If you add:

```
(promote-invisible "enabled")
```

to the `*gschemrc` files (or editing `system-gschemrc`), invisible floating attributes will also be promoted (and in memory removed)

However, if you enable this, then component slotting will break, because `gschem` expects certain floating attributes to be inside the symbol (in memory even though they are invisible). So you can add:

```
(keep-invisible "enabled")
```

to the `*gschemrc` files (or editing `system-gschemrc`). This is enabled by default, but has no effect unless `promote-invisible` is enabled.

So, to summarize, attribute promotion takes floating attributes inside symbols and attaches them to the outside of a placed symbol. Three `*rc` keywords control this behavior: `attribute-promotion`, `promote-invisible`, and `keep-invisible`.

12 Resource File

Gschem is highly configurable. All configuration is handled through a scheme based rc file. Gschem looks for three rc files:

- `system-gschemrc`: This is usually installed in `/usr/share/share/gEDA` and is required for gschem to run.
- `$HOME/.gEDA/gschemrc`: A per user file. Users should put defaults in this file they want to apply to all sessions.
- local directory `gschemrc`: This file holds the per project defaults. Things like `component-library` or `source-library` keywords go into this file.

The order of searching for these three files is as above (first `system-gschemrc`, then `$HOME/.gEDA/gschemrc`, and finally the local `gschemrc`).

A few rules about changing the files:

- Don't break any syntax rules. Doing so will cause the scheme interpreter (guile) to stop interpreting.
- Keywords/defaults always override what came before, with the exception of cumulative keywords (like `component-library`).

For more info as to what can be configured, please look in `system-gschemrc`.

More TBA.