

**OrCAD Layout<sup>®</sup> for Windows<sup>®</sup>**

**GerbTool<sup>™</sup> User's Guide**

Copyright © 1997 OrCAD, Inc. All rights reserved.

OrCAD, OrCAD Capture, OrCAD Design Desktop, OrCAD Layout, and OrCAD Simulate are registered trademarks, and PLD 386+, SDT 386+, SDT Release IV, and VST 386+ are trademarks of OrCAD, Inc.

GerbTool and Snoman are trademarks of WISE Software Solutions, Inc.

Microsoft, Windows, Windows NT and other names of Microsoft products referenced herein are trademarks or registered trademarks of Microsoft Corporation.

TrueType is a registered trademark of Apple Computer, Inc.

PostScript is a registered trademark of Adobe Systems, Inc.

All other brand and product names mentioned herein are used for identification purposes only, and are trademarks or registered trademarks of their respective holders.

MN-01-5048

Second Edition 30 June 97

Technical support	(503) 671-9400
Administration	(503) 671-9500
Fax	(503) 671-9501
General email	<a href="mailto:info@orcad.com">info@orcad.com</a>
Technical support email	<a href="mailto:techsupport@orcad.com">techsupport@orcad.com</a>
Web site	<a href="http://www.orcad.com">http://www.orcad.com</a>



9300 S.W. Nimbus Ave.  
Beaverton, OR 97008 • USA

# Contents

<b>Chapter 1</b>	<b>Introduction .....</b>	<b>1</b>
	Using this manual .....	1
	GerbTool features .....	2
<b>Chapter 2</b>	<b>Configuration .....</b>	<b>3</b>
	Color list file .....	3
<b>Chapter 3</b>	<b>Quick start.....</b>	<b>5</b>
	Starting GerbTool .....	5
	Creating a new aperture list .....	5
	Converting a CAD aperture list .....	6
	Creating a new design.....	8
	Opening an existing design.....	9
	Saving a modified layer .....	10
	Exiting GerbTool.....	10
<b>Chapter 4</b>	<b>GerbTool basics .....</b>	<b>11</b>
	GerbTool window.....	11
	Toolbars .....	12
	Settings bar .....	12
	Active layer/D-code.....	14
	Coordinate display .....	14
	Color chooser bar.....	14
	Birdseye view .....	14
	Split screen pane dividers .....	14
	Drawing area.....	14
	Crosshair cursor.....	14
	Film box.....	15
	Status bar .....	15
	Tool tips.....	15

	Design files .....	15
	Aperture list files .....	15
	Using GerbTool commands .....	16
	Mouse-button and function-key commands .....	16
	Shortcut Keys .....	17
	Interrupting a drawing process .....	18
	Ending a command .....	18
<b>Chapter 5</b>	<b>Performance tips .....</b>	<b>19</b>
	Speeding up GerbTool operations .....	19
	Using shortcut keys .....	19
	Interrupting, redrawing, and highlighting .....	19
	Undoing edits .....	19
	Programming mouse buttons and function keys .....	20
<b>Chapter 6</b>	<b>Uses for GerbTool .....</b>	<b>21</b>
	Layer alignment .....	21
	Creating NC drill files .....	22
	Importing drill files .....	22
	Panelizing .....	23
	Viewing or printing Gerber 274-D composite layers .....	23
	Merging designs .....	24
	Drawn pads .....	25
	Automatic silkscreen clean-up .....	25
	Creating a soldermask layer .....	26
	Transcoding .....	26
	Snoman pad and trace filleting .....	27
<b>Chapter 7</b>	<b>Command reference .....</b>	<b>29</b>
	File menu .....	29
	New .....	29
	Open .....	30
	Close .....	30
	Save .....	30
	Save As .....	30
	Save All .....	30
	Format .....	31



Merge.....	33
Import .....	34
Export .....	36
Page Setup .....	41
Print .....	42
Print Preview .....	42
Printer setup.....	42
Exit .....	42
Edit menu.....	43
Undo .....	44
Select .....	45
Item .....	47
Copy .....	47
Move .....	48
Delete .....	48
Clip .....	48
Join .....	49
Rotate .....	49
Mirror .....	49
Scale .....	50
D-code .....	50
Align Layers .....	51
Origin .....	51
Purge.....	51
View menu.....	52
Window .....	52
Zoom In .....	52
Zoom out .....	52
Pan .....	52
All .....	52
Film box .....	52
Redraw .....	53
Sketch .....	53
Overlay .....	53
Grid .....	53
Composites .....	53

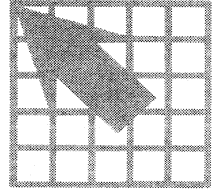
Virtual Panel.....	53
Clear Highlights.....	53
Highlights .....	53
Selections.....	54
Errors .....	54
Save .....	54
Recall.....	54
Previous.....	54
Toolbars.....	54
Split.....	54
Add.....	55
Layers menu.....	59
Edit .....	59
Colors.....	63
Create .....	63
Apertures menu.....	64
Edit .....	64
Report .....	67
Load.....	68
Unload .....	68
Merge.....	68
Compact.....	68
Convert .....	68
Save .....	68
Query menu .....	69
Item information .....	69
Net .....	70
UserData .....	71
Measure .....	71
Highlight .....	72
Copper .....	72
Extents .....	72
Options menu.....	73
Grid Snap .....	73
Ortho Line Snap .....	73
Arcs 360 .....	73

Metric .....	73
Configure .....	74
Macro menu .....	82
Run .....	82
Load .....	82
Edit .....	82
Tools menu .....	83
Panelize .....	83
DRC .....	88
Snoman .....	91
Teardrops .....	92
Netlist .....	94
Fix SilkScreen .....	96
Pad removal .....	97
Drill .....	97
Convert .....	99
Layer Spread .....	101
Vent .....	101

**Chapter 8      Macros .....** **103**

Creating a macro .....	103
Using variables .....	104
Coordinate lists .....	105
Repeating blocks of commands .....	105
Making decisions .....	106
Loading macros .....	106
Running macros .....	106
Macro language reference .....	107
Conventions used .....	107
Add functions .....	108
Aperture functions .....	117
Control statements .....	120
Database functions .....	125
Editing functions .....	138
Environment functions .....	165
File handling functions .....	191
File merging functions .....	195

	Mathematical functions .....	198
	Plotting functions.....	212
	Query functions .....	216
	String handling functions.....	219
	Tool functions.....	231
	User data entry functions.....	249
	Utilities and other functions.....	258
	Viewing functions.....	262
<b>Chapter 9</b>	<b>Aperture Conversion Rule files .....</b>	<b>273</b>
	Definition of an ACR file .....	273
	Creating an ACR file .....	273
<b>Chapter 10</b>	<b>Extended Gerber.....</b>	<b>283</b>
	Embedded apertures.....	283
	Aperture macros.....	284
	Layer compositing .....	285
	Viewing composites.....	285
	Converting from RS-274-D to extended Gerber format.....	285
<b>Chapter 11</b>	<b>Using custom apertures .....</b>	<b>287</b>
	Create a custom aperture.....	287
<b>Chapter 12</b>	<b>Working with text fonts.....</b>	<b>289</b>
	Editing a font .....	289
	Creating a new font.....	290
<b>Appendix A</b>	<b>Command ID values .....</b>	<b>291</b>
<b>Appendix B</b>	<b>Aperture list file format.....</b>	<b>297</b>
<b>Appendix C</b>	<b>Snoman concepts.....</b>	<b>299</b>
<b>Glossary</b>	<b>.....</b>	<b>301</b>
<b>Index</b>	<b>.....</b>	<b>303</b>



# Introduction

Welcome to GerbTool, the easiest, most powerful, and versatile CAM station available.

GerbTool provides a powerful set of Windows-based CAM tools, including a feature rich and robust Gerber/NC editor for ensuring a seamless link between PCB design and manufacturing. GerbTool is designed to provide CAD/CAM professionals with the tools they need for complete control over their CAM databases. From visual verification to high-level CAM tools, GerbTool simplifies and automates your PCB layout post processing and pre-manufacturing tasks.

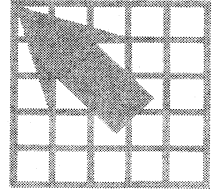
GerbTool's consistent and intuitive graphical user interface, and programmable mouse buttons and function keys, allow you to focus on accomplishing tasks, rather than on the technical details of operating the software.

## Using this manual

This manual was designed to assist you in using GerbTool's features. *Chapter 3: Quick start* is especially geared toward providing the information you need to become immediately productive. A prior knowledge of CAD/CAM concepts and your computer's operating system is assumed.

## GerbTool features

- Fast and easy to use.
- Unlimited file sizes.
- Accurate to 1/100 mil (.00001 in.).
- Fully automatic panelization and venting.
- Complete undo to beginning of session.
- Full design rule checking (DRC), including annular ring checking and stub detection.
- Snoman™ pad/trace filleting.
- Teardrop pads.
- NC drill optimizing, including step and repeat.
- Isolated pad removal.
- Automatic removal of silkscreen data from pads.
- Full support for true multilayer netlists, including net highlighting.
- Scalable check plots to HPGL, PostScript®, Laser printers, and all printers/plotters supported by Windows.
- Conversion of drawn pads to flashes.
- Macro language allows the addition of new commands.
- Metric and Imperial formats supported.
- Photo plotter support includes extended Gerber, FIRE9xxx, EIE, BARCO DPF and IPC-D-350.
- Accurate display of power and ground plane composites.
- Allows aperture scaling to create soldermasks, shrink/expand traces, and so on.
- Ability to scale layers to shrink or expand the database.
- Merge a complete design or a single Gerber file into another.
- Import NC Drill, HPGL, or BARCO files.
- View up to 999 layers simultaneously.
- Handles over 4000 apertures in up to 999 aperture lists.
- Aperture list conversion tools allow the addition of custom aperture list converters.
- Easily created custom apertures and custom fonts.



# Configuration

Unlike previous releases of GerbTool, this version does not require that you create a configuration file. Instead, use the Configure command from the Options menu.

## Color list file

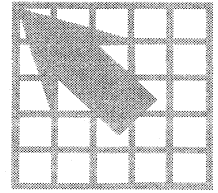
When starting up, GerbTool looks for a color list file named COLOR.RGB. Once the color list file is found, GerbTool reads the available colors from a red-green-blue (RGB) color and name pair list, then reads a list of the currently chosen colors. The currently chosen colors are those presented whenever you select colors from within GerbTool (for example, flash and draw colors).

```
# maximum 1024 colors available...
[RGB Color/Name pairs]
128  0  0      vga16red
   0 128 128   vga16cyan
   0 128  0    vga16green
245 245 245   WhiteSmoke
.
.
.
255 250 240   FloralWhite
253 245 230   OldLace
250 240 230   linen
250 235 215   AntiqueWhite
# maximum 32 current choice colors...
[Choice Colors]
blue
vga16green
white
black
coral
.
.
.
SteelBlue
SaddleBrown
DarkSalmon
DarkOrange
DeepPink
```

*Sample color list.*







# Quick start

In order to help you get started quickly, this chapter provides a quick overview of GerbTool and its processes. A more comprehensive description for each GerbTool function is provided in chapters 4 through 8.


## Starting GerbTool

To start GerbTool, choose GerbTool from the Tools menu in the OrCAD Layout for Windows session frame.

## Creating a new aperture list

GerbTool's most basic function is to read a Gerber file into memory and display it graphically on your screen. An aperture list file describes the shape and size of all the apertures used in the Gerber file. GerbTool automatically reads aperture list files from Layout.


---

 **Note** Aperture list files are not required for extended Gerber, FIRE9000 or EIE format Gerber files as they are embedded in the Gerber file.

---

You can also create your own aperture list files. There are two ways to do this. The easiest method is to convert your CAD aperture list into GerbTool format. Therefore, if you have an aperture list that is in a format listed in the table on the next page, you simply specify this as your aperture list and GerbTool automatically converts it for you.

---

 **Note** You do not need to convert aperture list files created by Layout (or by OrCAD PCB 386+).

---

If, on the other hand, you do not have a CAD aperture list, you can enter a new, non-existing filename when you specify an aperture list and let GerbTool create it for you. Although it will initially be empty, GerbTool creates new apertures as needed when reading in your Gerber file. You can then edit the aperture list, changing the shapes and sizes to meet the needs of your Gerber file.

## Converting a CAD aperture list

GerbTool provides aperture list conversion for most of the CAD and photoplotter aperture list formats in use today. The conversion process translates a CAD aperture list directly into GerbTool format, thereby reducing data-entry-related problems. Again, remember that you do not need to convert aperture files created by Layout. The following table shows the aperture list formats supported by GerbTool, along with the name of the Aperture Conversion Rule (ACR) file used for the conversion.

<i>Aperture list format</i>	<i>GerbTool ACR file</i>
ALLEGRO	ALLEGRO.ACR
CADSTAR	CADSTAR.ACR
CADSTAR 2	CADSTAR2.ACR
CONSULTEK	CONSULTK.ACR
CSI	CSI.ACR
CSI V4	CSI4.ACR
CSI Report	CSIRPT.ACR
DC-CAD	DC-CAD.ACR
DC-CAD 2	DC-CAD2.ACR
EAGLE	EAGLE.ACR
EDT	EDT.ACR
EDT 2	EDT2.ACR
EE Designer	EED.ACR
GraphiCode Report	GCREP.ACR
GerbTool Report	GTREP.ACR
HIWIRE	HIWIRE.ACR
IVEX	IVEX.ACR
Lavenir Report	LAVINER.ACR
Lavenir View	VIEW.ACR
MASSTECK	MASSTEK.ACR
OrCAD Layout (up to v6.42)	MASSTEK.ACR
McCAD	MCCAD.ACR

*Supported aperture list formats (page 1 of 2).*

---

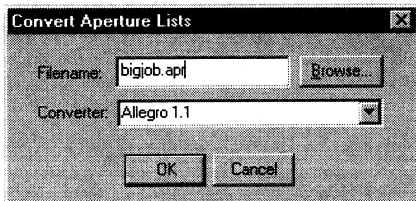
<i>Aperture list format</i>	<i>GerbTool ACR file</i>
MENTOR	MENTOR.ACR
OrCAD PCB II	ORCAD.ACR
PADS	PADS.ACR
P-CAD	PCAD.ACR
P-CAD V6	PCAD6.ACR
P-CAD V7/V8	PCAD7_8.ACR
P-CAD Report	PCADRPT.ACR
PRANCE	PRANCE.ACR
PRANCE 2	PRANCE2.ACR
PROTEL 1.0	PROTEL.ACR
PROTEL for Windows	PFW.ACR
SCICARDS 2	SCICARD2.ACR
SCICARDS	SCICARDS.ACR
TANGO	TANGO.ACR
ULTIBOARD	ULTIBRD.ACR
UNICAD	UNICAD.ACR
VALID	VALID.ACR

---

*Supported aperture list formats (page 2 of 2).*

When creating a design, specify your aperture list normally. GerbTool converts it to the proper format automatically. Or you may select the Convert command from the Apertures menu and convert it prior to loading it into GerbTool.

To convert a supported aperture list to GerbTool format, select the Convert command from the Apertures menu, specify an input filename, then select the appropriate converter in the Convert Aperture Lists dialog box.



*Convert Aperture Lists dialog box.*



**See also** For more information about converting aperture lists, see *Chapter 7: Command reference*.

---

## Creating a new design

To create a new design, choose either Auto or Manual mode after you choose New from the File menu.

### To create a design file automatically

- 1 From the File menu, choose New. GerbTool displays the Create New Design Wizard dialog box.
- 2 Enter a name for the new design in the Design Name text box, and specify a directory in which to create the design in the Design Folder text box, then choose the Next button. GerbTool displays the next dialog box in the Create New Design Wizard.
- 3 Choose Automatic, then choose the Next button. GerbTool displays the next dialog box in the Create New Design Wizard.
- 4 Specify the path to the directory that contains the Aperture list files for the new design in the text box, then choose the Next button. GerbTool displays a dialog box informing you that the design has been created.
- 5 Choose the Finish button. GerbTool displays the Edit dialog box and includes a list of all the layers found in the directory.
- 6 If you are using Gerber 274-D format, enter an aperture list for each layer by selecting that layer from the list, then entering the aperture list file (\*.APP) in the Aperture List field.
- 7 Choose the Edit button. GerbTool displays the Gerber format dialog box.
- 8 Select the appropriate format, m.n setting, and zero suppression, then choose the OK button. GerbTool returns you to the Edit dialog box.
- 9 Choose the OK button. GerbTool creates the new design according to your specifications.

### To create a design file manually

In order to create a design file manually in GerbTool, the layers and aperture lists must be in standard Gerber 274D format.

- 1 From the File menu, choose New. GerbTool displays the Create New Design Wizard dialog box.
- 2 Enter a name for the new design in the Design Name text box, and specify a directory in which to create the design in the Design Folder text box, then choose the Finish button. GerbTool displays the Edit dialog box and includes a list of all the layers found in the directory.
- 3 Enter a set of layers, and an aperture list for each layer in the appropriate fields.
- 4 Choose the Edit button. GerbTool displays the Gerber format dialog box.
- 5 Select the appropriate format, m.n setting, and zero suppression, then choose the OK button.
- 6 Choose the OK button. GerbTool creates the new design according to your specifications.

Whether creating design files automatically or manually, GerbTool creates a design file named UNTITLED.GTD in the current directory. You can use the Save command on the File menu to save your design file under a different name.

## Opening an existing design

Follow these steps to open an existing design.

### To open an existing design

- 1 From the File menu, choose the Open command. GerbTool displays the Open Design dialog box.
- 2 Locate and select the design you want to open, then choose the OK button. GerbTool displays the Edit dialog box.
- 3 If necessary, make any modifications, then choose the OK button. GerbTool loads the appropriate files.



---

**Note** Layout automatically generates a design file when post-processing (\*.GTD).

---

## Saving a modified layer

If a layer has been modified or changed, you are given an opportunity to save it when you select the Save, Save As, or Save All command from the File menu.

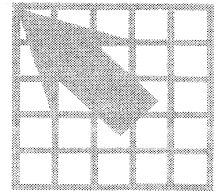


**Note** When GerbTool displays a list of files to save, you must select each file you want to save. Only those files selected are saved.

---

## Exiting GerbTool

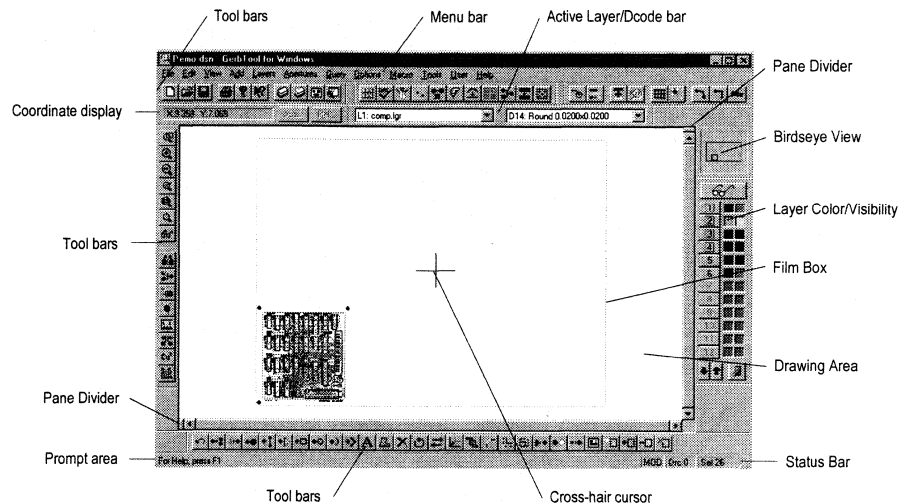
To exit GerbTool, choose the Exit command from the File menu. If any layers have been modified, GerbTool requests confirmation that you really want to exit.



## GerbTool basics

This chapter provides information on GerbTool basics.

### GerbTool window



The GerbTool window consists of the following components:

- Main menu bar where you can “pull down” the command menus.
- Toolbars where you can choose commands with a single click.
- Settings control bar where you control various program settings, such as metric display mode and grid snap.
- Color toolbars where you can change layer colors and visibility.
- Birdseye view that shows the current view window relative the extents of the loaded database.
- Adjustable pane dividers to split the drawing area into multiple views.
- Drawing area where GerbTool displays all database items.
- Crosshair cursor indicating the position of the mouse within the drawing area.
- Film box graphic that indicates the size of the current film box.
- Status bar with prompt area where GerbTool displays command messages.
- Tool tips on most window features including toolbar buttons, control bar buttons and menu items.

## Toolbars

Each icon within a toolbar represents a shortcut to a command. When you click on an icon in the toolbar, GerbTool executes the command associated with that icon.

## Settings bar

Use the settings bar to quickly and easily control various options with a single mouse click. This section describes each button on the settings bar.

### Sketch

This button toggles sketch mode on or off. When sketch mode is on, pads appear with an outline only, and traces appear as a single thin line. Besides speeding up redraw times, this mode can also help you spot stacked pads.

### Overlay

This button toggles overlay mode on or off. When overlay mode is on, items become transparent when drawn on top of each other. When overlay mode is off, items obscure whatever is drawn previously. Overlay mode makes it easier to spot stacked pads.

### View Composites

This button toggles the way GerbTool displays composite layers (extended Gerber and FIRExxxx only). When View Composites is on, the polarity of each layer, specified by the Polarity field within the Edit dialog box (from the Edit command on the Layers menu), is honored. If a layer is specified “Clear,” all data in that layer appears with the current background color.

### DRC Errors

This button toggles the display of rule violation errors on or off. If DRC errors exist and this setting is on, GerbTool displays the DRC View Errors dialog box.



**See** For information on error reporting, see the DRC section in *Chapter 7: Command reference*.

---



**Grid** 

This button toggles the system grid display on or off.



**See** For information on grids, see the Grid section in *Chapter 7: Command reference*.

**Grid Snap** 

This button toggles grid snap mode on or off. When grid snap mode is on, the crosshair cursor automatically jumps to the nearest grid point.



**See** For information on grid snapping, see the Grid Snapping section in *Chapter 7: Command reference*.

**Orthogonal Snap** 

Use this button to toggle orthogonal snap mode on or off. When on, lines drawn interactively are forced to the specified angle.



**Note** The current setting can be temporarily overridden by holding down the CTRL key.

**Arcs 360°** 

This button toggles the method of creating arcs used by the Arc and Circle commands (on the Add menu). If on all arcs are created using 360° circular interpolation. If off, all arcs are created using arcs of 90 or fewer degrees. This does *not* affect the way Gerber data is read from a disk file. It only pertains to adding new arcs with the Arc command (on the Add menu) and the Circle command (on the Edit menu).

**Metric** 

This button toggles metric mode on or off. When metric mode is enabled, GerbTool shows all information and editing fields that represent sizes and distances (for example, coordinates) in metric format.

### *Active layer/D-code*

You can dock the active layer/D-code control bar or allow it to “float” in a dialog box. Use this control bar to control the currently active layer and D-code. To change the active layer or D-code, simply use the drop-down list to select a different layer or D-code.

### *Coordinate display*

You can dock the coordinate display control bar or allow it to “float” in a dialog box. It shows at a glance the current location of the crosshair cursor. The format of the display is controlled by the current setting of the Me button within the Settings bar or the Metric command on the Options menu and the file format of the active layer.

### *Color chooser bar*

You can dock the color chooser bar or allow it to “float” in a dialog box. It is available at all times to change layer colors and visibility. When a layer is on, indicated by a red box around the layer number, it is both visible and editable. When a layer is off it is neither visible nor editable. When a layer is ref, indicated by a black box around the layer number, it is visible but not editable.

### *Birdseye view*

You can dock the birdseye view or allow it to “float” in a dialog box. The black rectangle represents the database extents while the red rectangle represents the current viewing window. Use it to determine at a glance exactly where your current view window is located.

### *Split screen pane dividers*

By adjusting the pane dividers, you can split the drawing area into up to four separate viewing windows. Each window represents a different view of your design. You can view and edit your data at multiple zoom levels or locations simultaneously.

### *Drawing area*

The drawing area is the area between the menu bar and the status bar. All database items are displayed here.


### *Crosshair cursor*

While the mouse position is within the drawing area, GerbTool displays the cursor as a crosshair. The exact location of the crosshair cursor appears in the Coordinate Display toolbar described above.

## Film box

The film box represents the size of the film on which you will plot, and is a graphic display only. It does *not* become part of your Gerber database(s).

---

 **Tip** You can control the size and color of the film box using the Options menu's Configure command, described in *Chapter 7: Command reference*.

---

## Status bar

GerbTool displays command status and prompts in this area.


## Tool tips

When you hold the mouse cursor over a window feature for a few seconds, a small popup window appears with a short description of the feature.

## Design files

GerbTool uses design files. A design file, as created by GerbTool (or Layout), contains information about the Gerber files, and their associated aperture list files, that constitute a single PCB. This usually includes filenames for inner and outer signal layers, silkscreen layers, soldermask layers, and so on. GerbTool also associates an operating environment with each design file. Thus, when you load an existing design file, GerbTool restores the environment to the state it was when you last saved the design file. This eliminates the need to continually reconfigure GerbTool each time you load a design.

---


 **Note** The default file extension for design files is configurable and is easily changed using the Configure command on the Options menu. Layout uses a .GTD extension for these files.

---

## Aperture list files


Aperture list files define the characteristics of each Gerber D-Code used in a design. For each D-Code the aperture list file defines a shape, size, type, and NC drill tool number (see *Chapter 7: Command reference*). GerbTool stores aperture lists in ASCII format. This makes it easy to create and modify aperture lists. It also allows easy conversion from most CAD system aperture lists.

---

 **See** For details of the aperture list format, along with an example aperture list, see *Appendix C: Aperture list file format*.

---

---

 **Note** The default file extension for aperture list files is configurable. You can change it using the Configure command on the Options menu. Layout uses a .APP extension for aperture list files it creates.

---

## Using GerbTool commands

This section describes the different ways to use GerbTool commands.

### *Mouse-button and function-key commands*

GerbTool comes pre-configured with the following mouse button and function key assignments.

---

<i>Key</i>	<i>Assignment</i>
Left mouse button	View Window
Middle mouse button	Zoom In
Right mouse button	None
F1	Redraw
F2	View Film Box
F3	View Previous
F4	View All
F5	Layers Edit
F6	Aperture Edit
F7	Aperture Report
F8	Query Highlight
F9	Query Item
F10	Measure End to End
F11	Add
F12	Remove

---

The assigned mouse and function key commands are available any time GerbTool is idle (for example, there is no command prompt in the prompt area).

## Shortcut Keys

Shortcut keys are available anytime GerbTool is idle, or when GerbTool prompts you to enter a point. Below is a list of the shortcut keys. GerbTool execute shortcut keys immediately without affecting the current command.

<i>Key</i>	<i>Action</i>
ENTER	Enter coordinate at cursor location
HOME	Snap cursor to center of item
PGUP	Zoom In
PGDN	Zoom Out
LEFT ARROW	Scroll page left
RIGHT ARROW	Scroll page right
UP ARROW	Scroll page up
DOWN ARROW	Scroll page down
+ or I	Zoom In
- or O	Zoom Out
0-9	Bring a layer to the top (1-10)
CTRL+0-9	Bring a layer to the top (11-20)
A	Turn on all layers
CTRL+A	Turn off all but active layer
B	Pop-up floating color box
C	Enter absolute coordinates
CTRL+C	Enter relative coordinates
D	Increment current D-code
CTRL+D	Decrement current D-code
CTRL+F	Edit configuration flags
CTRL+G	Edit system grid
H	Toggle highlights on or off

*Shortcut Keys (page 1 of 2).*

<i>Key</i>	<i>Action</i>
L	Increment active layer
CTRL+L	Decrement active layer
M	Run macro
CTRL+M	Toggle metric mode
P	Pan
CTRL+P	Toggle auto pan mode
CTRL+ALT+Q	Quit immediately without confirmation
R	Redraw
CTRL+R	View All
S	Toggle grid snap
CTRL+S	Screen print
U	Undo last edit
CTRL+U	Undo all edits
V	Toggle composite viewing
CTRL+V	Toggle virtual panel mode
1-9,0	Bring a layer to the top (1-10)
CTRL+[1-9,0]	Bring a layer to the top (11-20)

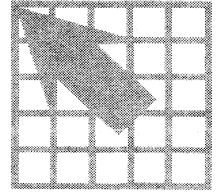
*Shortcut Keys (page 2 of 2).*

### ***Interrupting a drawing process***

Anytime GerbTool redraws the display or highlights a window of data, you can halt the drawing process by pressing the ESC key or clicking the right mouse button. This does not affect the operation of the command and in many cases speeds up its operation.

### ***Ending a command***

You can end a command, or end at least one level of a multistep command, by pressing the ESC key or right mouse button.



## Performance tips

This chapter provides tips for obtaining optimal performance from GerbTool.

### Speeding up GerbTool operations

#### *Using shortcut keys*

Shortcut keys are a powerful feature of GerbTool. These keys are available any time GerbTool is waiting for you to enter a coordinate (point) or whenever it is idle (that is, when no command has been selected). Using these keys you can snap to the center of a database item, change which layers are viewed, undo edits, and so on.



---

**See** For a complete list of available shortcut keys, see *Chapter 4: GerbTool basics*.

---

#### *Interrupting, redrawing, and highlighting*

You can speed up any command that redraws the database or highlights a group of items by canceling the drawing process. Click the right button or press the ESC key to halt the redraw. This doesn't affect the operation of the command; it affects only the redraw. Once you're comfortable with the operation of GerbTool commands you will find that this ability significantly speeds things up.

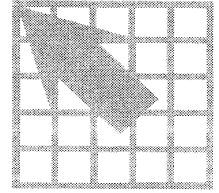
### Undoing edits

The Undo command provides a high level of freedom when making database edits. You can experiment and try different edits without fear of data loss. Since undo is available as the shortcut key U, you can undo edits immediately without having to exit the current command. Undo works for all edits regardless of size, and there is no limit to the number of edits you can undo. Remember to enable the undo capability with the Configure command (Options menu) *before* making your edits, then use the Edit menu's Undo command or the shortcut key U to undo as necessary.

## Programming mouse buttons and function keys

GerbTool's easy-to-use graphical user interface is further enhanced with the versatility of programmable mouse buttons and function keys. Using the Configure command on the Options menu, you can program the mouse button and function keys F1 through F12 with commands that you frequently use.





# Uses for GerbTool


This chapter provides several examples of uses for GerbTool.

## Layer alignment

Layer alignment involves lining up all layers so that when you view multiple layers simultaneously, they are correctly aligned. Proper layer alignment is also crucial to the successful creation of a multilayer netlist.

First select a master layer with which all other layers should be aligned and select an item on that layer to use as a reference point. Choose the Align command from the Edit menu and select the item you chose as a reference point. Then, select an item on each layer to be aligned that corresponds to the reference point. As you select each item, GerbTool aligns the host layer.

---


 **Tip** You can use the shortcut zoom in, zoom out, and pan keys (see *Chapter 4: GerbTool basics*) to locate the reference and corresponding items.

---

## Creating NC drill files

Use the Drill command on the Tools menu to create an NC Drill file from any layer. Choose the format for the drill file by choosing the NC Format button within the Drill editing dialog box (shown in *Chapter 7: Command reference*). Usually, the layer you choose to create a drill file from represents the pad master for the entire design. When creating NC Drill files, GerbTool translates the Gerber flashes (except targets and thermals) into drill “hits.” The Tool field, in the corresponding aperture list for the selected layer, determines the tool call-out for each drill hit.

---

 **Note** Use the Report command from the Apertures menu to determine if you have a tool assigned to each flash. Edit the aperture list if required so all flashes are assigned a tool.

---

GerbTool then optimizes the drill hits, according to your specifications, for fastest through-put.

Perform panelization prior to executing the Tools menu’s Drill command. If your drilling equipment has a small memory capacity, perform a “virtual” panelization. This allows GerbTool to insert the needed step and repeat codes into the output drill file. Preferably, if your drilling equipment has enough memory, you should perform a normal non-virtual panelization. This results in a fully optimized panel for the maximum drilling efficiency.


## Importing drill files

Use the Drill command from the Import menu to load an NC Drill file into the active layer. Layout creates a drill file, THRUHOLE.TAP, that you can import in order to automatically include the drill information for your design.

You can also create a new empty layer first by selecting the Create command from the Layers menu. Make sure that the layer you choose is the active layer.

When loading a NC drill file, GerbTool converts the drill hits into Gerber flashes. Each tool called out in the drill file is located in the aperture list for the active layer. If GerbTool can’t find a tool, it adds an aperture to the list with an “Unknown” shape and the correct tool assignment. You can then edit the aperture to correct the shape, size, and so on.

---


 **Note** Use the Report command (from the Apertures menu) to determine if any apertures were added. Those added are highlighted.

---

## Panelizing

With GerbTool, panelizing is a simple, one-step process when using the Auto Panel feature. After activating the layers to be panelized (only), select the Panelize command from the Tools menu, ensure that the Auto Panel button is selected (shown in *Chapter 7: Command reference*), and enter the minimum image border-to-border spacing in the X and Y fields. The spacing you specify should be between adjoining edges of the intended images. GerbTool automatically calculates the maximum number of images that will fit inside the current film box. After asking for confirmation, GerbTool completes the panelization process. Depending on the setting of the Virtual button, GerbTool either copies the proper number of images into the database or notes the number of copies and their location for display purposes.

---


 **Note** You can use the right mouse button or press the ESC key to stop the drawing process anytime during the panelizing process. This usually provides a noticeable improvement in the overall time to complete the panelizing process without affecting the finished panel in any way.

---

## Viewing or printing Gerber 274-D composite layers

You can use black and white for layer colors, to provide accurate viewing of composite power and ground layers. Setting the negative layer to white on a black background and the positive layers to black results in a realistic depiction of the final film.


---

 **Note** Since the negative layer must be displayed first, it is important that the negative layer be before the positive layers .

---

To print a composite layer, view your composite layers as described above, then use the Print command (from the File menu). The printed image appears on the page exactly as it does in the display.

---

 **Note** Since the image for printing is created in a high resolution off screen bitmap, the film box and display grid may appear on the output page. You can disable this by setting the film box color to the background color using the Film box command (Options menu) and disabling the display of the grid using the Grid command (Options menu), or shortcut key G.

---

## Merging designs

You can merge two or more designs into a single Gerber file so they can be photoplotted simultaneously. This reduces manufacturing costs by making full use of photoplot film.

In order to merge designs in this manner, the following conditions must exist:

- Each file must be in the same Gerber format and have the same m.n values, and zero suppression.



**See** For information on Gerber formats, see the discussion of the File menu's Format command in *Chapter 7: Command reference*.

---

- Each file must use the same aperture list. That is, the size and shape of each D-code must be the same in each aperture list.
- The respective layers in each design must be the same. For example, layer 1 of each design must be silkscreen, layer 2 of each design must be the top layer, and so on.

### To merge design files

- 1 From the File menu choose the Open command and select the .GTD file for the first design. GerbTool displays a list of layers in the Layers-Edit dialog box.
- 2 Choose the OK button. GerbTool displays the design at coordinates 0:0 in the lower left of the Film Box.
- 3 From the Edit menu, choose the Scale command. The Scale and/or Offset Layers dialog box appears. Choose the desired offset value.
- 4 Select All Visible from the Layers drop down list.
- 5 Choose the OK button. GerbTool responds with an error message: "Command cannot be undone. Continue?"
- 6 Choose Yes. This shifts the first design.
- 7 From the File menu, choose Merge and then choose Design. In the Merge Design dialog box, choose the second design. The second file appears in the film box.

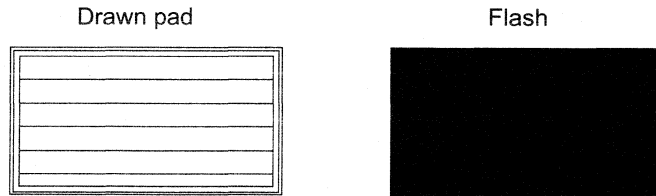


**Note** OrCAD recommends that you merge only Gerber 274-D files.

---

## Drawn pads

Occasionally, CAD systems may output an irregularly shaped or sized pad using multiple draws to “fill in” the shape, rather than a more efficient single flash. This results in larger Gerber files than necessary and increases processing times. Also, it is virtually impossible for high-level CAM tools such as DRC to recognize the drawn pads as pads rather than as collections of traces. The difference between a typical drawn pad and a comparable flash is shown below.




*Drawn pad versus a flash.*

The drawn pad in this example requires 27 separate Gerber commands to accomplish what one Gerber flash can accomplish. Thus, if you have 2000 of these drawn flashes, you’ll have a Gerber file with at least 54,000 lines when flashes could accomplish the same thing in only 2000 flashes.

Using the Pads command from the Convert menu, you can convert all your drawn pads to flashes. You do this by identifying one occurrence of a drawn pad and allowing GerbTool to find all drawn pads that match. And, to increase GerbTool’s ability to recognize matching drawn pads, you can specify a tolerance value to compensate for some CAD systems’ round-off errors. By specifying a tolerance, you allow GerbTool to relax its criteria for determining matching drawn pads.

---

 **Tip** Converting drawn pads to flashes should be the first thing you do to your designs. This usually ensures trouble-free conversion. Also, you must convert all drawn pads to flashes *before* generating a netlist or running most other CAM tools.

---

## Automatic silkscreen clean-up

GerbTool has the ability to automatically clean up a silkscreen where lines touch or are too close to the pads. Using the Fix Silkscreen command from the Tools menu, you specify the layer(s) that the silkscreen and pad master are on and the minimum spacing that must be maintained between the silkscreen data and the pads. If you want, you can use window mode to clean up isolated areas rather than the entire silkscreen layer. GerbTool then cleans up all areas where silkscreen lines are too close to a pad. Each offending line is moved just enough to eliminate the violation.



**See also** For more information on silkscreen cleanup, and to see before and after illustrations, see *Chapter 7: Command reference*.

---

## Creating a soldermask layer

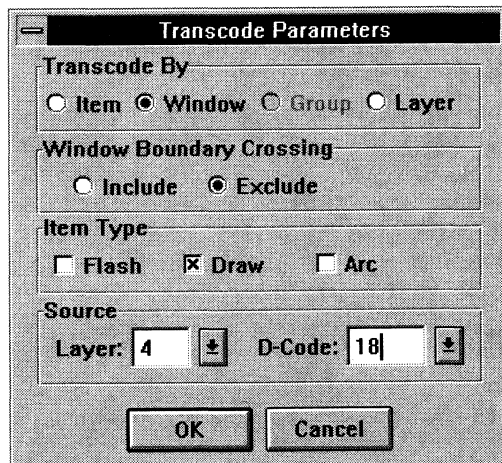
Creating a soldermask is a simple and easy process using the Scale command, from the D-Code selection on the Edit menu.

First, create the soldermask layer by copying the pad master layer onto a new layer. Use the Copy command to copy the pad master to the new layer. When copying, select Create Layer from the Copy to Layer fields drop-down list. This creates a new layer for the new soldermask data.

Now, select the Scale command, enter a scale factor for X and Y and select the Fixed Amount field in the D-Code Scale dialog box, then choose the OK button. GerbTool adds apertures to the corresponding aperture lists as necessary and replaces the D-codes with the new scaled D-codes. The original D-codes within the aperture lists are not modified.

## Transcoding

Using the Transcode command (from the D-Code selection on the Edit menu), you can transcode (transform D-Code) item by item or by selecting a group. Using selection criteria, you can choose exactly which D-Codes are transcoded. For example, to transcode only draws with a D-Code of D18 on layer 4 and within a particular window, specify the selection criteria as shown in the following example:



*Restrictive selection criteria.*

After selecting and highlighting the D-codes, GerbTool prompts you for the new D-code and then performs the actual transcoding.

## Snoman pad and trace filleting

Snoman is a highly configurable dialog box that specifies a method of optimizing pad/trace junction points. This is often referred to as *filleting* or *teardropping* (see *Appendix D: Snoman concepts* for a technical description of Snoman). The purpose of Snoman is to increase your manufacturing yield by adding more copper in the area of the pad/trace junction, thereby eliminating any possible pad/trace separation. Snoman is used primarily when dealing with small pads and traces (such as micro vias in the 30 mils or less range) but can be used anywhere to prevent pad/trace separation. Snoman provides additional versatility by allowing you control of the size and location of the generated Snoman pads, along with an integral DRC to eliminate any possible spacing violations.



**See** For a complete description of how to use the Snoman tool, see *Chapter 7: Command reference*.

---

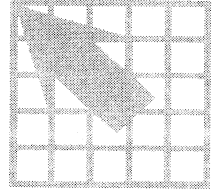


**Trivia** Snoman derives its unusual name from the appearance of a Snoman pad placed on top of a host pad, which resembles a “real” snowman.

---







## Command reference

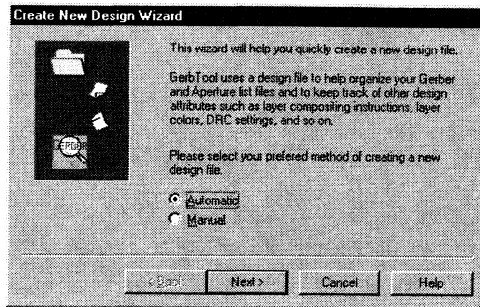
This chapter provides details for using each GerbTool command.

### File menu

The File menu selection displays commands for dealing primarily with files and directories. The menu commands are described in the following sections.

#### New

The New command presents the New Design Wizard as shown below.




#### *New Design Wizard*

There are two modes for the New command: Automatic and Manual.

#### **Automatic**

When you select Auto mode GerbTool builds a design file for you automatically. You specify the source folder and GerbTool determines which files are Gerber or aperture lists. The Gerber filenames are sorted first alphabetically and then by layer number if one is found. If an aperture list is found that it is not already in GerbTool format, GerbTool tries each configured aperture list converter until a match is found. Finally, each aperture list is matched to a suitable Gerber filename. GerbTool then displays the Edit dialog box where you can make any final adjustments if necessary.

 **Note** This command is affected by the number of aperture converters configured and by the filename extensions that are ignored. In general, fewer aperture list converters and more ignored filename extensions result in faster performance.

#### **Manual**

When you select Manual mode GerbTool creates an empty design file for you, and then displays the Edit dialog box for you to enter the Gerber files and aperture lists.

**Open** 

This command displays the file chooser and prompts you for a design file to load. You can use a wildcard specification to obtain a list of files from which to choose. After you specify a design file to load, GerbTool displays the Edit dialog box where you can define or modify the layer structure and, if needed, define or change the Gerber input format specification.

**Close**

This command closes the current design.

**Save** 

Select this command to save the current design file and optionally any modified layers or aperture lists. This command does not close the current design; you can continue to work on it after saving. You must use this command, Save As or Save All to save modified layer data.

**Save As**


Select this command to save the current design file under a different filename and optionally any modified layers or aperture lists. This command does not close the current design; you may continue to work on it after saving.

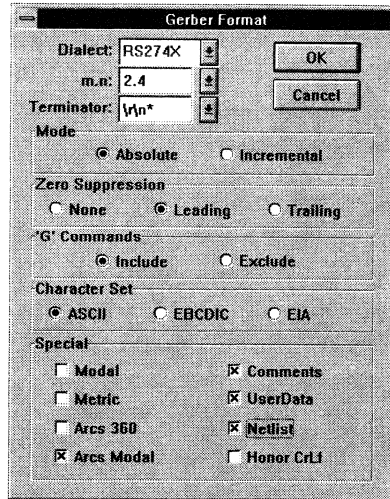
**Save All**

Select this menu item to save the current design and all modified layers or aperture lists. This command does not close the current design; you may continue to work on it after saving.

## Format

This menu selection has two commands: Gerber and Drill. When you select one of these commands, GerbTool displays a format dialog box from which you set the global formats for the file type in question.

 **Note** GerbTool supports both global and local formats. Global formats apply to all layers that do not have a local format assigned to them. Use this command to edit of the global formats only. See the Layers menu's Edit command in *Chapter 7: Command reference* for more information on local formats.



Typical format dialog box.

You can specify the correct format for that type of file (in this example, Gerber) by editing a format dialog box. The illustration above shows a Gerber Format editing dialog box, which includes the following fields:

**Dialect** Indicates the specific format of the Gerber language such as RS274D, extended Gerber, FIRE9xxx and EIE. If in doubt, choose RS274D.

**m.n** Coordinate format such as 2.3. This specifies 2 decimal digits before an implied decimal point and 3 following. (for example, 12250 represents 12.250 if the coordinate format is 2.3). Because of limitations in the representation of arcs in Gerber format, it is best to use at least a 3.4 coordinate precision for designs that include arcs.

**Terminator** Indicates the block terminator (EOB). Use \r to indicate a carriage return (ASCII 13) and \n to indicate a line feed (ASCII 10).

**Mode** Choose Absolute or Incremental (see Glossary for descriptions of these terms).

**Zero suppression** Indicates whether leading zeros or trailing zeros are suppressed, or there is no zero suppression.

**“G” commands** Indicates whether GerbTool includes “G” commands (for example, G01) when you output Gerber files.

**Special** You can enable Modal compression to reduce the size of your files by removing all redundant draft codes and coordinates. Or, you can enable Metric mode indicating that your files are in metric format. You can also specify whether all circular interpolated arcs should be considered 360° or quadrant, enable the saving of G04 comments, enable the output of UserData information, enable the output of Netlist information embedded within the Gerber file(s), and specify that carriage returns and line feeds should be honored as block terminators.

You can toggle between metric and inch format, as well as change m.n formats after loading a design. If you change formats after loading, all layers are marked as modified.



**Caution** If you change formats after loading and do not save all layers, the next time you load that design, the saved format may not match that of the unsaved Gerber files.

---

Selecting the Netlist button tells GerbTool to save netlist information within the Gerber file. If you have previously saved a Gerber file with netlist information, you can remove it by deselecting the Netlist button and saving.

---




**Note** It is important that you specify the correct format *before* loading a new design. The critical format items are m.n, mode, and trailing zeros. If you load a design with an incorrect format, GerbTool will display it with unpredictable results. If you inadvertently load a design this way, reload the design and click on the Format button of the Edit dialog box to correct the format.

---

## Merge

The Merge command has two modes: Design and Gerber.

---

 **Note** All merge commands require that you ensure the critical format items (mode, m.n and zero suppression) of the file or files being merged match those of the currently loaded design.

---

### Design

Selecting this command allows another complete design to be merged layer by layer into the current design. If a layer from the external design doesn't exist in the current design, you will be prompted to create a new layer.

### Gerber

Use this command to merge a Gerber file into the currently active layer. GerbTool prompts you for a filename. You can use a wildcard specification to obtain a list of files from which to choose. The specified filename is *not* added to the list of loaded layers. Rather, the contents of the file are read in and appended to the active layer.

## Import

The Import command has a number of options: BARCO DPF, HPGL, IPC-D-356, and Drill.

### BARCO DPF

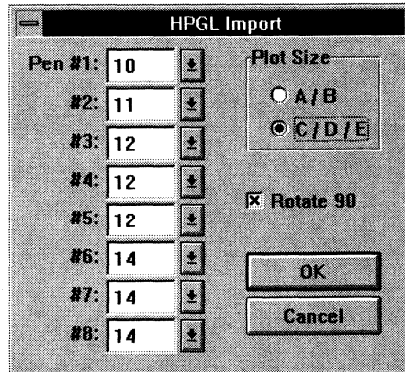
Use this command to import one or more BARCO files into the currently loaded design. This command begins importing the specified files into the active layer if it is empty. If it is not empty, GerbTool creates a new layer following the active layer. GerbTool creates as many layers as necessary to import all the files you specify.



**Note** Once a BARCO DPF file is imported into a layer it effectively becomes Gerber data and will indeed be saved as Gerber if the layer is subsequently saved. To output the layer in BARCO DPF format use the BARCO DPF command from the Export menu.

### HPGL

Use this command to merge an HPGL plot file into the currently active layer. After selecting a file to import, GerbTool displays the following HPGL import dialog box.



*HPGL Import dialog box.*

Using this dialog box, you can specify the expected plot size, whether to rotate the plot data and which D-codes to use for each HPGL pen.

### IPC-D-356

Use this command to import an IPC-D-356 netlist into your design. Since an IPC-D-356 netlist contains information pertaining to pads and not traces, GerbTool must generate an internal netlist prior to importing an IPC-D-356 netlist, to ensure that your database contains a full and complete netlist after importing. While this may sound redundant, the added benefit of an “automatic netlist comparison” is well worth it. The netlist comparison feature produces a report file detailing any differences between the internal netlist and the imported netlist, in addition to highlighting any differences. Optionally, you can update the database UserData fields with the component/net data from the IPC-D-356 file. Then, you can use GerbTool commands, including the Item Info command (on the Query menu), to examine and manipulate the true reference designators, pin numbers, and so on.

### Drill

Use this command to import an NC Drill file into the active layer.



**Note** This command requires that you insure the critical format items (mode, m.n and trailing zero suppression) for the file, or files, being loaded match those of the currently specified drill format.

---

## *Export*

The Export command includes these format options: IPC-D-350, IPC-D-356, BARCO DPF, HPGL and PostScript.

### **IPC-D-350**

GerbTool places designs exported to IPC-D-350 format into one file containing all layer data specified within the currently loaded design. The specified output file contains all data necessary to reproduce your design on any IPC-D-350 compatible device.

### **IPC-D-356**

GerbTool places designs exported to IPC-D-356 format into one file containing all layer data specified within the currently loaded design. The specified output file contains all netlist data associated with the current design.

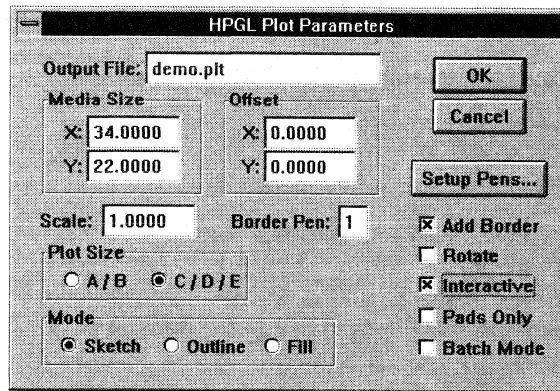
### **BARCO DPF**

GerbTool places designs exported to BARCO DPF format into a separate file for each layer. You select which layers to export and specify the output filenames. If you enable the Auto Rename button, GerbTool outputs all selected layers, renaming each layer automatically using the filename extension specified in the New Extension field.



## HPGL

When you select this command, GerbTool displays the HPGL Plot Parameters dialog box.



*HPGL Plot Parameters dialog box.*

**Output File** Specifies the output file that receives the HPGL plot data.

**Media Size** Specifies the page size (in inches) for the plot. By default, GerbTool sets these values according to the Plot Size setting, but you can change them as desired.

**Offset** Specifies the offset (in inches) from the edge of the paper.

**Scale** Specifies the scale for the plot. A scale of “1” indicates a 1:1 correspondence between the plot and the design. A scale of “2” indicates a 2:1 correspondence between the plot and the design, and so on.

**Border Pen** Specifies the pen size used to draw the plot border (if any).

**Plot Size** Specifies the plot size for the design. The Media Size field is initially set according to plot size.

**Mode** GerbTool provides three modes of output when plotting on a HPGL compatible plotter: Sketch, Outline, and Fill. Sketch mode is the fastest but does not show width on draws and some flashes. Outline mode shows true width on all objects but they are outlined only. Fill mode shows true width, and all objects are completely filled in as they would appear on a photoplot. Fill mode is the slowest and is extremely hard on plotter pens.

**Add Border** This option adds a border to your plots.

**Rotate** This option specifies that GerbTool rotates the plot 90 degrees.

**Interactive Mode** Use Interactive mode to interactively position each layer on the output page. To position an image on the page, select the image (using the left mouse button) and then drag it to the proper location and release the mouse button. During interactive plot positioning, a menu of buttons is provided along with several plot-specific shortcut keys.




*HPGL interactive control dialog box.*

The Plot button saves the page layout and plots the data. The OK button saves the page layout and quits the interactive session without plotting. Use the Reset button to reset the images to their initial positions for the session (if the dialog box has been pinned) or quit the interactive session without saving the page layout or plotting the data.

Shortcut keys are available during an interactive plot session. These are:

- C for absolute coordinate entry
- I for page layout initialization
- L to cycle the current layer forward
- CTRL+L to cycle the current layer backward
- S to snap (align) the current layer on top of another layer
- R to redraw the page layout

---

 **Note** There are two files within the GerbTool program directory that affect each HPGL plot. The files HPGL.INI and HPGL.DEI are prefixed and appended, respectively, to the plot output. If you have special requirements, you can edit these files as needed.

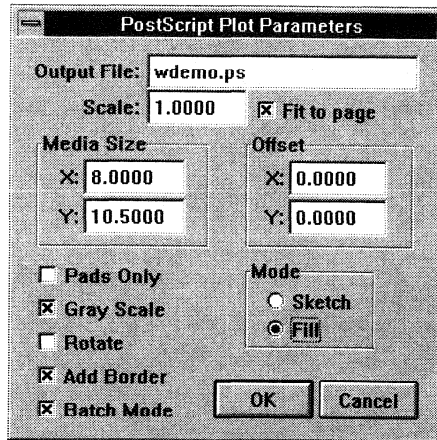
---

**Pads Only** This option instructs GerbTool to plot only pad information in the plot file.

**Batch Mode** This option instructs GerbTool to output each visible layer to a separate output file. During batch mode operation, if the Output File field is empty, GerbTool derives the output filenames from the filename associated with each layer and the currently configured HPGL filename extension (as explained in the Configure command discussion, later in this chapter). If, on the other hand, the output file field contains a filename, GerbTool appends a number representing the number of the input layer (demo.001, demo.002, and so on).

## PostScript

GerbTool provides PostScript output, allowing you to plot your data on any device that supports PostScript. This includes typesetters capable of producing production quality artwork.



*PostScript Plot Parameters dialog box.*

**Output File** Specifies the output file that receives the PostScript plot data.

**Scale** Specifies the scale for the plot. A scale of “1” indicates a 1:1 correspondence between the plot and the design. A scale of “2” indicates a 2:1 correspondence between the plot and the design, and so on.

**Fit to page** Specifies that the plot is automatically fitted to the page size.

**Media Size** Specifies the page size (in inches) for the plot. By default, GerbTool sets these values according to the Plot Size setting, but you can change them as desired.

**Offset** Specifies the offset (in inches) from the edge of the paper.

**Pads Only** This option instructs GerbTool to plot only pad information in the plot file.

**Gray Scale** This option tells GerbTool to produce accurate black and white composites as well as halftone images. When Gray Scale mode is disabled, all colors other than the background color are printed as black. When enabled, all colors (other than black/white) are converted to a different greyscale.

**Rotate** This option specifies that GerbTool rotates the plot 90 degrees.

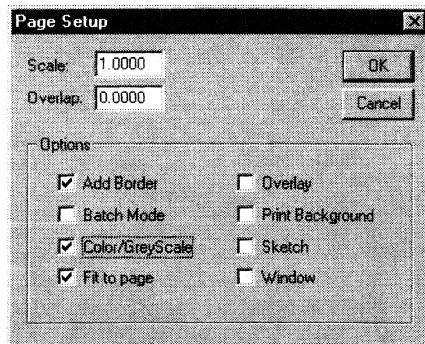
**Add Border** This option adds a border to your plots.

**Batch Mode** This option instructs GerbTool to output each visible layer to a separate output file. During batch mode operation, if the Output File field is empty, GerbTool derives the output filenames from the filename associated with each layer and the currently configured Postscript filename extension (as explained in the Configure command discussion, later in this chapter). If, on the other hand, the output file field contains a filename, GerbTool appends a number representing the number of the input layer (demo.001, demo.002, and so on).

**Mode** GerbTool provides two modes of output for PostScript: Outline and Fill. Outline mode shows true width on all objects, but with outlines only. This allows you to check for overlapping features. Fill mode shows true width, and all objects are completely filled in as they would appear on a photoplot. Fill mode may produce a larger output file.

## Page Setup

Use this command to configure how GerbTool will format each page printed by the Print command.



*Page Setup dialog box.*

**Scale** Specifies the scale for the print. A scale of “1” indicates a 1:1 correspondence between the print and the design. A scale of “2” indicates a 2:1 correspondence between the print and the design, and so on.

**Overlap** Indicates the amount of overlap for pages of a multi-page plot to allow proper alignment when taping the pages together.

**Add Border** This option adds a border to your prints.

**Batch Mode** This option instructs GerbTool to output each visible layer to a separate output file. During batch mode operation, if the Output File field is empty, GerbTool derives the output filenames from the filename associated with each layer and the currently configured Postscript filename extension (as explained in the Configure command discussion, later in this chapter). If, on the other hand, the output file field contains a filename, GerbTool appends a number representing the number of the input layer (demo.001, demo.002, and so on).

**Color/Grey Scale** If your printer is color-capable, this option tells GerbTool to produce a color print. Otherwise, (if your printer is not color-capable) GerbTool produces accurate black and white composites as well as halftone images. When this option is disabled, all colors other than the background color are printed as black. When enabled, all colors (other than black/white) are converted to a different greyscale.

**Fit to page** Specifies that the plot is automatically fitted to the page size.

**Overlay** Specifies that GerbTool prints a third color in areas where two differently colored objects are superimposed over each other. This option applies only if you have a color-capable printer.

**Print Background** Specifies that GerbTool prints the color for the design background. By default, this option is off.



**Note** Printing background colors uses a lot of ink (especially for C-, D-, or E-sized designs). Use this option with discretion.

**Sketch** Specifies that GerbTool draws only the outlines for objects in the design, rather than filling in those objects.

**Window** Specifies that GerbTool prints only that area of the design that falls within a user-defined window.

## *Print*

Select this command when you want to print the viewed layers. Use this command to print your design on any printer/plotter supported by Windows.

## *Print Preview*

Use this command to view how each page of your design would print before actual printing begins.

## *Printer setup*

Use the Printer Setup command to select and configure the current Windows default printer.

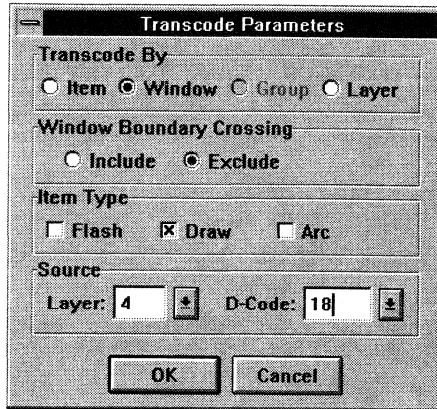
## *Exit*

Select this command when you want to exit GerbTool. The current design file can be saved, and you will be prompted to confirm that you want to quit if any layers have been modified.

## Edit menu

The Edit menu includes these commands: Undo, Select, Item, Copy, Move, Delete, Clip, Join, Rotate, Mirror, Scale, D-Code, Align Layers, Origin, and Purge.

You can modify the selection criteria for all editing commands that require you to modify one or more database items. GerbTool commands are flexible in the selection of data to modify. For example, you can select single items, or items within a window, group, or complete layer.



*Typical selection criteria.*

With this dialog box, you can control whether flashes, draws, arcs or any combination of all three are selected. Further, you can select whether a single item, window, group or layer is selected. In Window mode, you can select whether to include items that cross the window boundary. And finally, you can select whether to restrict the selection to a particular layer or D-code.

You can end any editing command by choosing the right button, pressing the ESC key, or selecting another menu item.




**See** For details on using GerbTool shortcut keys, see *Chapter 4: GerbTool basics*. Shortcut keys are selected with one keystroke and operate immediately, even during another command.

## Undo

Use this command to 'undo' changes you've made to the currently loaded database. Undo information is saved in a 'last in - first out' fashion. This means that you 'undo' changes in the reverse order in which the changes were made. This allows you to undo the most recent changes first. You may also use the shortcut key U to invoke the undo command even during another editing command.

---

 **Note** Undo must be enabled with the Configure command (Options menu) prior to making any edits if you plan to use this command.

Undo increases the amount of memory GerbTool requires. If you do not require the undo capability, you may disable undo with the Configure command (Options menu). Disabling undo will release any memory currently associated with undo information and prevent further undo memory use.

---



## Select

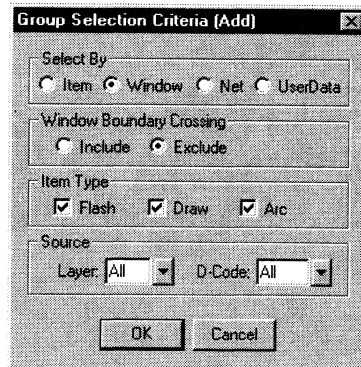
Most editing commands (such as Copy, Move, and so on) allow you to work with single items, windows of items or groups of items. The commands available in the Select menu allow you to manage the grouping of items for use by these editing commands. When a command allows group selection mode, it will use the currently selected group created and maintained by the different Select commands. Select groups are also persistent from one command to another. For example, if you rotate the current select group, the rotated data will remain selected ready for another command.

The Select menu contains the following commands for controlling select groups:

### New Group



Use this command to start a new group of selected items. When you choose this command, GerbTool prompts you for confirmation to clear the current select group (if any). This does *not* destroy any data. It simply deselects the current select group. If you respond affirmatively, GerbTool displays the Group Selection Criteria dialog box.



*Group Selection Criteria dialog box.*

### Add To



Use this command to select items and place them in the current select group. When you choose this command, GerbTool displays the Group Selection Criteria dialog box, from which you can choose those items you wish to add to the select group.

### Remove From



Use this command to select items and remove them from the current select group. When you choose this command, GerbTool displays the Group Selection Criteria dialog box, from which you can choose those items you wish to remove from the select group.


### Invert

Use this command to invert the current select group. That is, all currently selected items are deselected and all deselected items become selected. One use of this command is to allow you to quickly select all but a few items by first selecting the items you don't want and then inverting the select group.

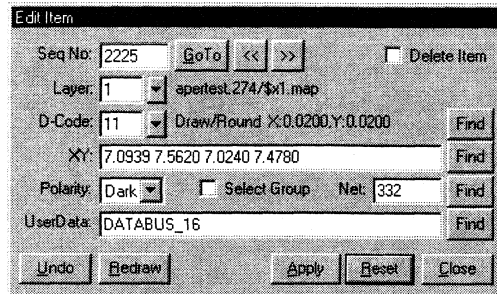
### Off



Use this command to clear the current select group and remove highlighting from any highlighted items. This does *not* destroy any data. It simply deselects the current select group.

Item 

This command displays a floating dialog box you can use to edit pertinent information associated with each database item. In addition, there are extensive controls for navigating from one item to another including the ability locate an item based on its sequential position in the database, D-code, X-Y coordinate, net and UserData value. You can also step forward and backward one item at a time using the supplied directional buttons. Lastly, you can use the 'N' hot key to automatically advance to the next item in the database.



*Edit Item dialog box.*

The UserData field is of special note as you can use this field to attach textual information to individual database items. Any text you associate with your database will automatically be saved within your Gerber files the next time you save them. This also allows you to pass data to other groups in your organization transparently.

An obvious use is to associate actual reference designators, pin numbers and net names with each pad thereby adding intelligence to your Gerber databases. Besides being able to see UserData using the Item command (Query menu), macros also have complete read/write access to each UserData field. This allows some powerful tools to be built upon GerbTool.

Other than a 256 character size limit, there are no restrictions on text associated with a database item.

Copy 

Use this command to copy single items, windows, or groups of items. By specifying a valid destination layer in the Copy to Layer field, you can copy all selected items to that layer.



**Note** If you select data from more than one layer and copy to a destination layer, GerbTool merges all into the destination layer. If you do not choose a destination layer, GerbTool copies the data into the respective source layers.


**Move** 

You can use this command to move a single item, a window or groups of items. By specifying a valid destination layer in the Move to Layer field, you can move all selected items to that layer. As with the Copy command, if you select data from more than one layer, GerbTool merges all moved data into the destination layer. If you do not choose a destination layer, GerbTool moves the data into the respective source layers.

**Delete** 

Select this command when you want to delete items from one or more layers. You can delete vertices, single items, a window or groups of items.

---


 **Note** If Undo is disabled, you will be prompted for confirmation when deleting items.

---

**Clip** 

Use this command to specify a window in which GerbTool deletes all data with automatic clipping of draws that pass through the window. If group mode is selected, GerbTool deletes only those items within the select group.

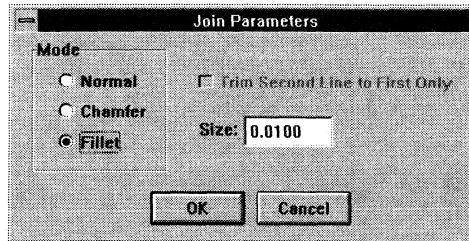
---

 **Note** The On Boundary selection controls whether flashes that straddle a window boundary are deleted.

---

## Join

Use this command to join two line segments using several different methods.




*Join Parameters dialog box.*

Using Normal mode, GerbTool extends or trims the two selected line segments as needed so that they connect. Naturally, this command will not work for parallel or near parallel lines. An option to Normal mode, Trim Second Line to First Only, helps when you have a long line in one direction and several lines intersecting the long line. With this option, GerbTool modifies only the second line you select. The remaining modes, Chamfer and Fillet, use the Size field to determine how far back to trim each of the two selected lines before adding the chamfer or fillet.

## Rotate

Use this command when you need to rotate a window or group of items. You can select Window mode or Group mode. You can also supply a pivot point (interactive) or allow automatic calculation of the center of the data for the required pivot point.

---

 **Note** If you enter a rotation factor of 90° or 270° and the Auto 90° button is selected, Rotate automatically compensates for asymmetrical pads, such as rectangles, by replacing the D-code with an equivalent D-code with opposite dimensions. You can add new D-codes to the appropriate aperture list.

---

## Mirror

Use this command if you need to mirror (flip) a group of items either horizontally or vertically. You can specify the direction to mirror and whether to prompt for the pivot point or automatically calculate it. This command can also be used to flip a secondary side layer that was designed as seen from the primary side.

## Scale

Use this command to apply coordinate offsets and scale to a loaded design.

The offsets and scale are applied to the selected layers. By applying a scale factor it is possible to expand or shrink the size of your database. For example, if you design your boards at 2X you can set both the X and Y scale factor to 0.5 to convert your files to 1X.



---

**Caution** You cannot “undo” this command. Make sure you save any important edits before running this command.

---

## D-code

The D-code menu item includes these commands: Transcode, Expand, Scale, and Polarity.

### Transcode

Use this command to change the D-Code of an individual item, window, group or complete layer. By changing the D-Code of an item, you can alter its size and shape. Another way to change the size and shape of an item is to edit the aperture list directly.

### Expand

Use this command to expand one or all custom apertures in a design. This command is required to plot a design that contains custom apertures when your photoplotter is unable to create the apertures you need.

When you choose this command, GerbTool asks you for the D-code you want to find. You may enter a specific D-code or you may enter zero to instruct GerbTool to expand all custom apertures.


### Scale

Use this command to shrink or expand the size of one or more D-codes. One use of this command is to create soldermasks automatically. GerbTool adds new apertures to the corresponding aperture list as needed based on your specified scale factor. If the Fixed Amount check button is enabled, GerbTool adds the scale values to each D-code. Otherwise, each D-Code size is multiplied by the scale values specified.

## Polarity

Use this command to control the item level polarity of EIE and BARCO format files as well as FIRE9XXX raster fill polygons. When using item level polarity, the ordering of the data is crucial. You may find that you need to move data ‘in place’, thereby placing the ‘moved’ data at the end of the database.

---

 **Note** Gerber RS274D does not support polarity at all. Extended Gerber files only support polarity at the layer level which is controlled using the Edit command from the Layers menu. FIRE9XXX format only supports raster fill polygons at the item level. Otherwise layer oriented polarity is assumed.

---

## Align Layers

Use this command to align any misaligned layers. First determine the layer to which all other layers should be aligned (the master layer) and select an item to use as a reference point. Then select an item, on each layer you want to align, that corresponds to the reference point. As you select each item, GerbTool automatically aligns the entire layer.

## Origin

Use this command to relocate the origin (0,0 point) of the database. GerbTool prompts you for a point to define the new origin. The film box is moved to the new origin.

---


 **Note** This command causes GerbTool to mark all layers as modified.

---

## Purge

Use this command to compact the currently loaded database for more efficient use of memory. Since GerbTool doesn’t actually remove data from memory during edits, memory may become fragmented and less efficient. Therefore, occasional purging can help GerbTool perform optimally.

---

 **Note** Purging destroys any “undo” information that currently exists. Do not use this command unless you are sure you don’t need to undo any previous edits.

---

## View menu

The View menu includes commands to control viewing window location and size.

### Window

Use this command to select a new viewing window. Two points are required to define a window. These points define a rectangle that encompass the area that is to become the new viewing window. Use this command when you want precise control over the viewing window.

### Zoom In

This menu item halves the size of the current viewing window using a center point that you supply. This command provides a closer look at the displayed data.

### Zoom out

Doubles the size of the current viewing window using a center point you supply. Use this command to increase the size of the viewing window.

### Pan

Moves the current viewing window to a new location. The new location is centered about a point you supply. This command does not change the size of the viewing window.

### All

This command adjusts the size of the viewing window to encompass all of the currently displayed layer(s). If you have deleted data from any displayed layers, you may need to use the Extents command (Query menu) to calculate the current extremes of the database.

### Film box

Select this command to adjust the size of the viewing window to display the contents of the currently specified film box. This command does not check to see that all data lies within the film box borders. Therefore, depending on the film box size, not all data may be displayed.



**Redraw** 

This command redraws the current viewing window.

**Sketch** 

This command toggles sketch mode viewing on and off. When sketch mode is enabled, pads are shown with an outline only, and traces are displayed as a single thin line. Besides speeding up redraw times considerably, this mode can also help you identify stacked pads.

**Overlay** 

This command toggles overlay viewing mode on and off. When overlay mode is enabled, items become transparent when drawn atop each other. When disabled, items obscure whatever is drawn previously. Overlay mode makes it easier to spot stacked pads.

**Grid** 

This command toggles the grid display on and off.



**Note** Grid is also available as a shortcut key by pressing CTRL+G.

---

**Composites** 

Enables the correct viewing of composite layers. When this button is enabled the polarity of each layer, specified by the Polarity field within the Edit dialog box, will be honored. If a layer is specified 'Clear,' all data in that layer is displayed with the current background color.

**Virtual Panel**

Virtual panel mode (and hence the display of virtual panels) may be toggled on and off using this command. You can also use the CTRL+V shortcut key.

**Clear Highlights**

This command clears any and all highlights that currently exist. Use this command after a command such as Highlight (Query menu) or Drill (Tools menu).

**Highlights**

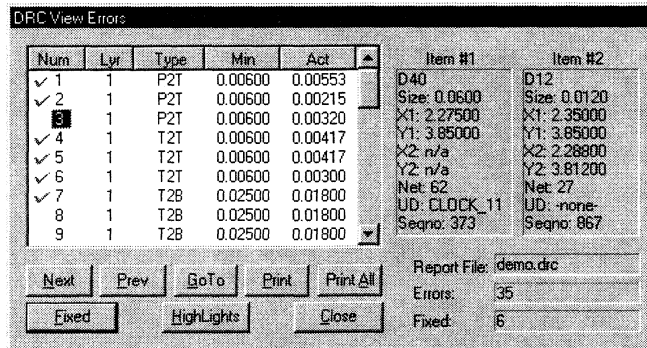
This command toggles the display of normal highlights on and off. Normal highlights are all highlights not indicating a select group or DRC error.

## Selections

This command toggles the display of select group highlighting. This does not change the actual select group. Only the highlight display is changed by this command.

## Errors

Use this command to view errors after performing a DRC. If DRC errors exist, the DRC View Errors dialog box appears.



DRC View Errors.

## Save

Use this command to save the current viewing window for later recall. There are eight positions available, 1-8, for saving. The current viewing window is saved in the position that you indicate. Use the Recall command (View menu) to recall any of the saved viewing windows.

## Recall

Use this command to recall a previously saved viewing window. If any of the eight possible positions does not have a viewing window associated with it, the corresponding positions in the menu are disabled.

## Previous

This command recalls the last viewing window. This allows you to toggle between two viewing locations.

## Toolbars

This selection presents a menu listing all of the available toolbars. You may toggle each toolbar on and off as desired.


## Split

This command splits the drawing area into multiple panes. By dragging the pane dividers to the desired location you can have up to four separate viewing panes. Each pane may have a different zoom level and/or location allowing you to view and edit multiple views of your design simultaneously.

## Add

The Add menu includes commands you can use to enter various types of new database items. This menu includes the Flash, Draw, Rectangle, Vertex, Circle, Arc Ctr, Arc 3 Pt, Polygon, and Text commands.

---

 **Note** GerbTool creates all circles and arcs using circular interpolation or multiple line segments depending on the style indicated by the Configure command (Options menu). Use circular interpolation with care as not all photoplotters support circular interpolation. Segmented circles and arcs use the chord angle specified using the Configure command.

---

### Flash

Use this command to add a flash to the active layer. GerbTool prompts for a point at which to add the flash. As you move the cursor about the screen an outline shape of the current D-code appears. Choose the left mouse button to add a flash at that location.

### Draw

Use this command to draw line segments in the active layer. GerbTool prompts you for a starting point and subsequent points to form continuous traces. Choose the right mouse button or press the [Esc] key to start a new trace.

### Rectangle

Use this command to draw line segments in the shape of a rectangle to the active layer. GerbTool prompts you for a starting corner point and an opposite corner point.

### Vertex

Use this command to add (and move by dragging the mouse) a vertex anywhere on an existing line segment.

### Circle

Use this command to draw a circle by entering a center point and a point on the radius. The circle is drawn on the active layer, using the current D-code, in a counter-clockwise direction.


### Arc Ctr

Use this command to define an arc by entering a center point, a point defining the radius and starting angle, and a point defining the ending angle. The arc is drawn on the active layer, using the current D-code, in a counter-clockwise direction.

### Arc 3 Pt

Use this command to define an arc by entering its end points and a point on the arc's circumference. The arc is drawn on the active layer, using the current D-code, in a counter-clockwise direction.

---


 **Note** To create 90° arcs easily and quickly, press the '9' key. This automatically constructs a 90° degree arc.

---

### Polygon

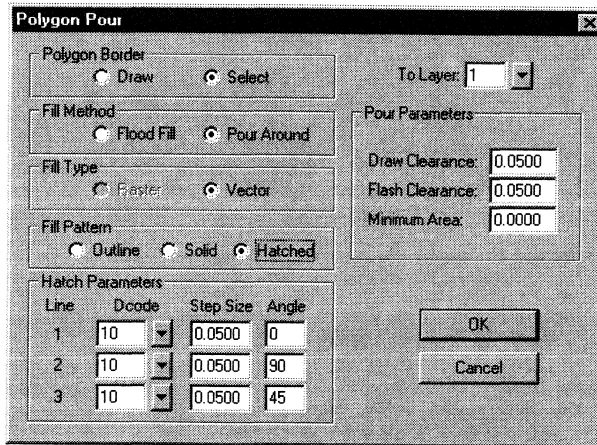
Use this command to select or enter a closed polygon. GerbTool fills or pours the interior of the polygon using either a raster fill or vector fill method. This command is commonly used to create ground plane areas.

---

 **Note** Raster filling is not supported in RS274D format Gerber files.

---

When entering a polygon you can automatically close the polygon by pressing the [End] key. You can also close the polygon manually by entering a point at the point which began the polygon. Regardless of the method, GerbTool then outlines the polygon with the current D-code, as displayed in the status bar, and fills the interior of the polygon. In Flood Fill mode, GerbTool fills the interior of the polygon with increasing aperture sizes. As it fills toward the center of the polygon the aperture sizes become larger. In Pour Around mode, GerbTool fills the interior of the polygon, as above, while maintaining clearance, as specified by the Draw Clearance and Flash Clearance fields, around all circuitry.



*Polygon edit dialog box.*

Since you can generate many smaller polygons to effectively “pour” around the circuitry, you can use the Min Area parameter to specify the minimum size area. GerbTool eliminates any filled areas that would be smaller than the specified Min Area.

The Pour Around option also supports three additional modes: Outline, Solid and Hatch mode.

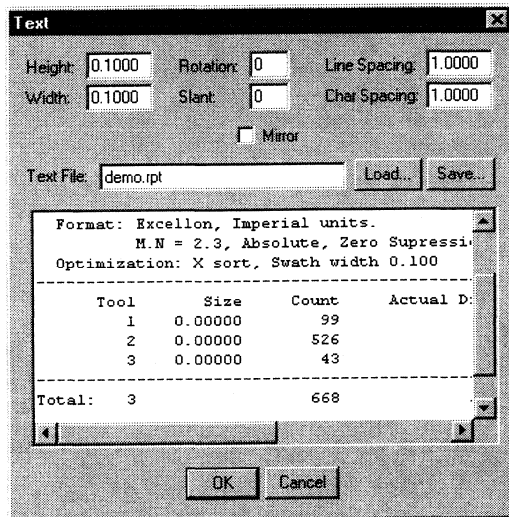
If you select Outline mode, GerbTool does not fill the resultant polygons. You can use this type of output to drive PCB prototyping equipment.

If you select Solid mode, GerbTool completely fills the resultant polygons using the same methods described for the Fill command above.

If you select Hatched mode, GerbTool fills the resultant polygons with a cross hatched pattern as specified in the Hatch Parameters section of the Edit dialog box. Up to three lines may be used with different sizes and angles for each line.

## Text

Use the Text command to insert text into the database as a sequence of line segments. Therefore, you can control the line thickness of the inserted text by changing the current D-code. You can rotate or slant text. You can also specify the height and width of the text and the intercharacter and line spacing. GerbTool provides a text editing window where you can enter as many lines of text as needed. You have full editing and scrolling capabilities. You can even load and save text files. The Text command presents the Edit dialog box shown below:



*Text Parameters dialog box.*

## Layers menu

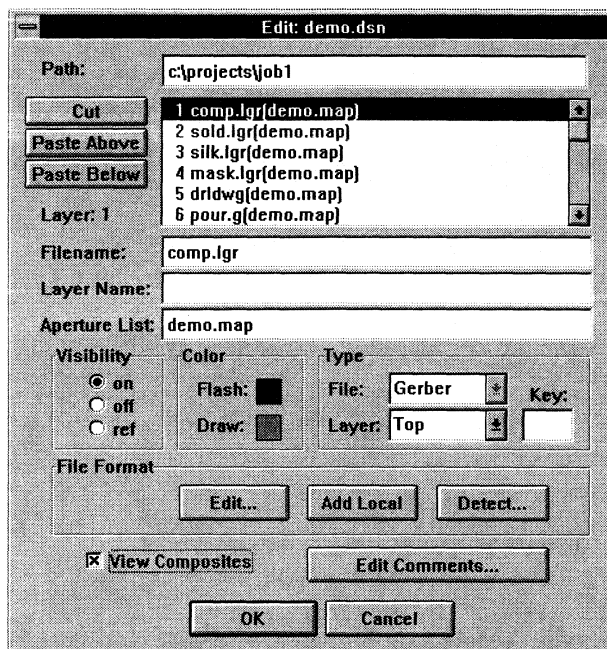
The Layers menu includes commands for managing the individual layers within your design. The menu selections include Edit, Colors, and Create.

### *Edit*

Use the Edit command to edit layers in your design with the Layers dialog box. The effects of editing certain fields within the dialog box differ, depending on whether you are loading a design or editing after loading.

Within the Layers dialog box you specify:

- The path used to locate the Gerber and aperture list files
- Gerber files
- Aperture list files
- Layer names
- Layer visibility
- Flash/draw colors
- Layer type
- File format
- Extended Gerber compositing instructions



*Layers dialog box.*

While loading, edit the Path field to tell GerbTool where to find the specified files if they do not contain a path as part of the filename. Entering a wildcard specification (for example, \*.GBR) in the Filename field displays the file chooser. You can choose more than one filename in which case all selected filenames are entered in one step. You can also enter a wildcard in an Aperture List field to obtain the file chooser. If you select a filename, GerbTool enters it in the current field.


**Note** You do not need to fill in the Aperture list field for each specified Gerber file. If an Aperture List field is left blank, it assumes the contents of the previous Aperture List field. If the Aperture List field for the first specified layer is blank then it assumes the currently configured default aperture list file.

If you change the Path field after loading, GerbTool marks all Gerber and aperture list files as modified. Then, you can save them in a location other than the location from which you loaded them. Changing the contents of a Filename field after a design is loaded causes that layer to be marked as modified. Then, you can save a layer under a new filename. If you enter a filename into a previously empty Filename field, GerbTool attempts to load the newly specified Gerber file. If it does not exist, you can create it. Changing the contents of an Aperture List field causes GerbTool to load the specified aperture list, if it is not already loaded, and link it to the corresponding Gerber file.



Regardless of whether you are loading or not, the Visibility button controls the visibility of the specified layer, the Flash and Draw color buttons control the color of flashes and draws respectively and the Layer Type button displays a menu of layer types from which you can choose.

---

 **Note** It is important to specify the Layer type for each layer, as several GerbTool commands check this field for the proper type before processing each layer. For example, the Pad Removal command (Tools menu) only operates on Inner-type layers.

---

The following is a description of each field within the Edit dialog box (from the Layers menu).

**Path** Specifies the path to the directory containing the Gerber and aperture list files.


**Cut, paste above, and paste below** Use these to re-order the layer structure both before and after a design is loaded. In addition, if you cut a layer from a loaded design without pasting the layer, GerbTool prompts you to unload that layer from memory. Then, you can free memory if your resources become low.

**Layer** Specifies the current layer. To make a layer current, click on the layer within the scrollable layer list.

**Filename** Specifies the name of a Gerber file loaded into the current layer. If you do not include an explicit path, GerbTool uses the contents of the Path field to locate the file.

**Aperture list** Specifies the aperture list file to associate with the current layer.

---

 **Note** If the specified aperture list is not already in GerbTool format, it is converted automatically.

You do not need to fill in the Aperture list field for each specified Gerber file. If an Aperture List field is left blank, it assumes the contents of the previous Aperture List field. If the Aperture List field for the first specified layer is blank then it assumes the currently configured default aperture list file.

---


**Layer name** Specifies a composite layer name used by the extended Gerber format files. This is *not* a file name.

**Visibility** Controls the visibility of the specified layer. Options are On, Off or Ref.

**Flash/Draw** Control the color of flashes and draws, respectively.

**Layer (Type)** Specifies a layer type of Top, Inner, Bottom, Plane, Composite, or Other.

---

 **Note** It is important to specify a type for each layer, as several GerbTool commands check this field for the proper type before processing each layer. For example, the Pad Removal command (Tools menu) only operates on Inner layers.

---

**Polarity** Specifies the polarity of layers to form composites. Select either Dark or Clear. This field is only valid for extended Gerber and FIRE9000 layers.

**Link** Links layers together to form composites. Enter a numeric value. GerbTool links together layers with identical link numbers to form a composite. This field is only valid for extended Gerber and FIRE9000 layers.

**File format** Specifies the correct data format *before* loading begins. With these buttons, you edit the selected layers format, whether global or local, add or remove local formats, and automatically detect the format of one or more layers.

**Edit** Use this button to edit the file format of the selected layer. If that layer has a local format added (the File Format Edit button will have Local to its left), the format GerbTool displays for editing is specific to the selected layer. Otherwise, GerbTool uses the global format. (See the File menu's Format command in *Chapter 7: Command reference* for more information on editing file formats.)

**Add Local** Adds a local format to the currently selected layer, which you can use to specify that the layer has a different format than other layers of the same file type. By default, each layer references a global format common to all layers of a particular type (for example, Gerber). You can use local formats to load different file types into the same design. Then you can simultaneously view and edit any files in the same design regardless of their file type.

**Del Local** Removes a local format.

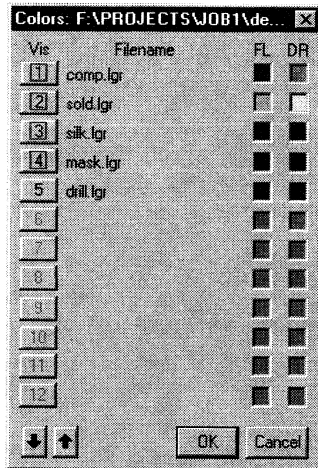
**Detect** Detects the file format of the selected layer and updates the format associated with that layer.

The effects of editing certain fields within the Edit dialog box differ depending on whether you are loading a design or editing after loading.

## Colors


The Colors command displays the Layers Color dialog box. Use this dialog box to specify:

- Visibility; ON, OFF or REF.
- Draw and Flash Color.



*Layers Color dialog box.*

When a layer is on, indicated by a red box around the layer number, it is both visible and editable. When a layer is off, it is neither visible nor editable. A black box around the layer number, indicates that it is the reference layer, in which case it is visible but not editable.

 **Note** If you find that you don't use the reference visibility setting, you may disable the availability of the reference status with the Configure command (Options menu).

This command is not available if the ColorBar is visible on the window.

## Create

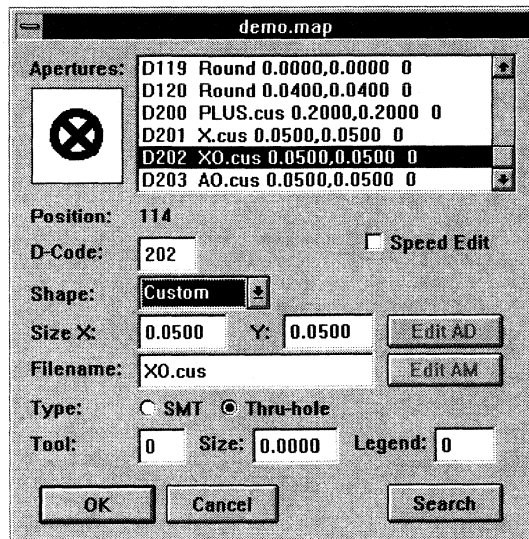
This command creates a new empty layer with the filename TEMP<#####>. <ext>, where <#####> is the layer number and <ext> is the currently configured extension for Gerber files. If you later decide to save the contents of this layer you may use the Edit command (Layers menu) or the Save command (File menu), using the 'Save As' option, to change the filename to something more meaningful.

## Apertures menu

The Apertures command displays the Edit, Report, Load, Unload, Merge, Compact, Convert, and Save commands.

Edit 

Use this command to edit a previously loaded aperture list. A list of currently loaded aperture lists, if such exists, appears for you to choose from. GerbTool displays the selected aperture list for you to edit.



*Aperture editing dialog box.*

If, after editing an aperture list, you want to discard the changes you've made, you can choose the Cancel button. To keep your changes, at least temporarily, choose the OK button. GerbTool updates the in-memory copy of the aperture list. To save an aperture list, you must use the Save command (Apertures menu).

The Aperture editing dialog box contains two sections; a list for selecting apertures to edit and the actual editable fields. The Position field is for reference only and is not editable. The nine remaining fields are:

<i>Field</i>	<i>Possible values</i>
D-Code	10 - 4095
Shape	Round, Square, Rectangle, Oblong, Donut, Diamond, Octagon, Thermal, Therm45, Target, Complex, Custom
Size X	0.0 - 9.9999
Size Y	0.0 - 9.9999
Filename	Custom aperture filename or aperture macro
Type	Surface-mount or through-hole
Tool	0 - 999
Tool Size	0.0 - 9.9999
Legend	0 - 4095

**D-Code** Normally, you select a D-code from the apertures list, but you can change this field to add new apertures.

**Shape** Choose a shape for your aperture. If you choose Custom, the Filename field becomes available for you to specify the filename of the custom aperture (see *Chapter 11: Using custom apertures*). GerbTool automatically adds the required .CUS extension, if needed, when loading the custom aperture. If you choose Complex, the Filename field becomes available for you to specify a valid aperture macro name.

**Size X/Size Y** When editing the Size X field, the Size Y field, if currently set to 0.0 also assumes the value of the X field.

**Filename** If the current aperture shape is Custom, enter the filename of a custom aperture file. You can use a wildcard to invoke the file chooser. If the current aperture shape is Complex, enter a valid aperture macro name.

**Type** Specifies whether the D-code represents a surface mount or through-hole pad. This information is needed when building multilayer netlists (see *Chapter 7: Command reference*).

**Tool** Edit this field if you intend to extract NC Drill information from a layer, or merge a NC Drill file into a layer, using this aperture list.

**Size** Specifies the size of the tool indicated in the Tool field.

**Legend** Used when creating a drill drawing using the Drawing command from the Drill menu. You can enter a D-code to represent this tool in a drill legend.


**Speed edit** Specifies that GerbTool change the operation of this dialog box to make it easier to enter aperture lists manually. Normally, when editing an aperture list, pressing the ENTER key updates the current aperture and advances to the next aperture. When you reach the end of the aperture list, new apertures are added automatically. Moving from field to field is accomplished using the TAB key or mouse. The Speed edit option activates only the Shape and X/Y size fields. Furthermore, the ENTER key moves from field to field except for the Y size field. While editing the Y size field, the ENTER key advances to the next record, as usual, before moving to the Shape field. This change in operation allows fast aperture list creation using only the ENTER key to move from field to field and to advance to the next record.

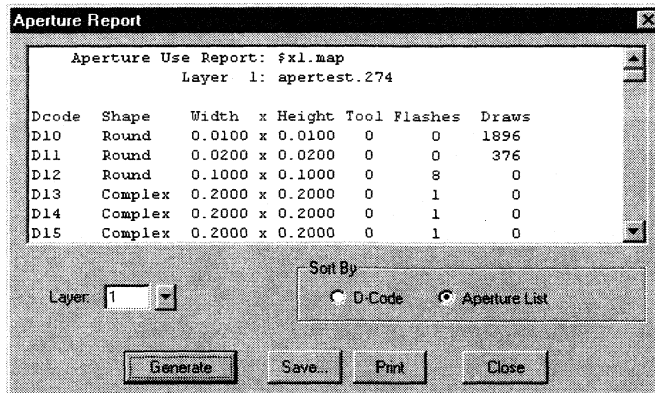
**Edit AD and Edit AM** These buttons are only active if the shape is Complex. Use them to edit the extended Gerber aperture definition (AD) and the aperture macro (AM) respectively. For FIRE9xxx aperture lists, use the Edit AD button to edit an aperture definition in native FIRE9xxx format.

**Search** Use this to search for an aperture that contains the text string you specify. You can search for any text appearing in the scrollable aperture list. For example, you could enter D200 to find that particular D-code or you could enter `rect` to find the next occurrence of a Rectangular aperture. You could also enter `.05` to find the next occurrence of a 50-mils aperture.

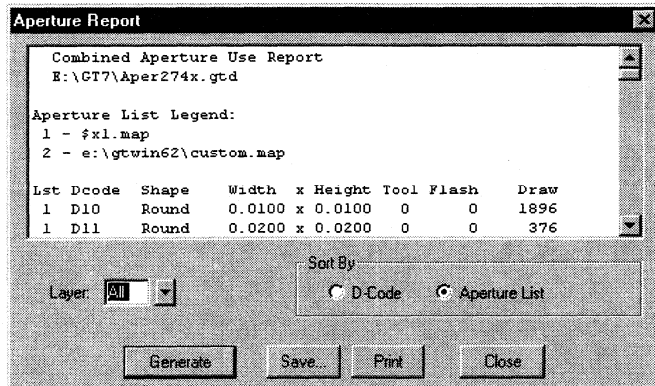
## Report

Select this command to generate an Aperture Report. An aperture report details the D-codes, along with their definitions, used on a per layer basis. Included in the report are use counts for both flashes and draws.

 **Note** If an aperture has an unknown shape, or is zero in size, GerbTool highlights it for easy recognition.



*Single Layer Aperture Use Report.*



*Combined Layer Aperture Use Report.*

**Generate** Displays a report for the layer specified in the Layer field. Entering 'all' or '0' in the Layer field instructs GerbTool to generate a Combined Aperture Report for all loaded layers. Use the scroll bar to view all of the report if it does not fit entirely within the window. You can edit the report.


**Save** Saves the report to a file.

**Print** Prints the report to the default Windows printer.

## Load

Select this menu item when you need to load or create an aperture list. GerbTool prompts you to specify a file. You can use a wild card specification to obtain a list of files from which to choose. If the specified aperture list doesn't exist, you can create a new one. GerbTool creates an aperture list, using default values, then loads it. If you specify an existing aperture list, GerbTool simply loads the specified aperture list.

---

 **Note** You may load and edit aperture lists independent of a design.

---


## Unload

Use this command to remove a previously loaded aperture list. GerbTool removes the aperture list you select, if it is not required by the currently loaded design. If you have modified the aperture but have not saved the changes, GerbTool prompts you to do so.

## Merge

Use this command to merge two or more loaded aperture lists. GerbTool merges all aperture lists associated with the currently viewed layers into a new aperture list. Each layer is then associated with the new aperture list. GerbTool remaps the D-codes of each layer accordingly.

---


 **Note** Save the new aperture list if any remapped layers are saved.

---

## Compact

Use this command to remove unused and redundant apertures within an aperture list. Select an aperture list to compact. Each layer associated with the selected aperture list is then re-associated with the new aperture list and the D-codes remapped accordingly.

---

 **Note** It is important that the new aperture list be saved if any of the remapped layers are saved.

---

## Convert

GerbTool has the ability to convert most CAD and photo plotter aperture list formats directly into GerbTool format. (See *Chapter 3: Quick start* for a complete list.)

You can specify an input filename and select the appropriate converter using the pull-down list.

## Save

Use this command to optionally save any modified aperture lists.

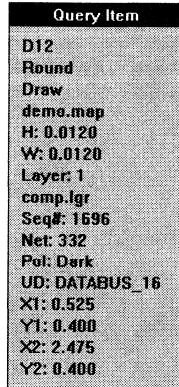


## Query menu

The Query command displays the Item, Net, UserData, Highlight, Measure, Copper, and Extents commands, which are described in the following sections.

### Item information

Use the Item Info command to obtain information on individual items within the database. The information appears in a dialog box as shown below.

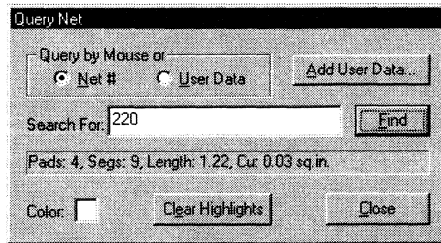


*Query Item display.*

As you select items, GerbTool highlights each item and its D-Code definition, along with the X-Y location and other information. You can select items with the mouse or using the N key to automatically advance to the next sequential item in the database.

## Net

Use this command to highlight true multi-layer nets using a variety of colors. Selected nets remain highlighted until you specifically clear them.



*Query Net dialog box.*

**Query by Mouse, Net or UserData** You can select a net at anytime by pressing the left mouse button anywhere on a line segment or flash. You can also search for nets by their GerbTool net number or their UserData.

**Search For** Specifies the net or UserData value for which to search.


**Find** Choose this button to have GerbTool find and highlight the net that contains the value in the Search For field.

**Clear Highlights** Choose this button to clear any nets currently highlighted. This does not change the net, it merely removes the highlighting.

**Color** This changes the color of subsequent nets selections when you select a new color. Previous selections are not altered.

**Add UserData** Once you have a net selected you may globally add a UserData value to all items in the selected net. This provides an easy way of assigning meaningful net 'names' to your nets.

---

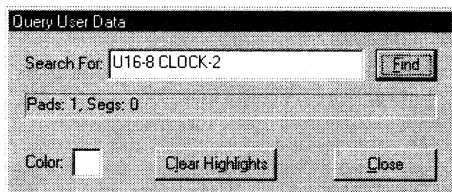
 **Note** This command relies on the netlist information supplied by a previous invocation of the Generate command. If a netlist does not exist, GerbTool prompts you to create one.

---

## UserData



Use this command to highlight all items that contain a specific UserData value. Each selected item remains highlighted until you specifically clear the highlighting.



*Query UserData dialog box.*

**Search For** Specifies the net or UserData value for which to search.

**Find** Choose this button to have GerbTool find and highlight the net that contains the value in the Search For field.

---

**Note** For a successful match, the search value can be located anywhere within an item's UserData value. This means that a search value "DATA" would match both "DATA32" and "MYDATA". The comparison is not case-sensitive.

---

**Clear Highlights** Choose this button to clear highlighting. This does not change any items, it merely clears the highlighting.

**Color** Changes the color of subsequent nets selections when you select a new color. Previous selections are not altered.

## Measure

The Measure command displays the Point to Point and Edge to Edge commands, which are described in the following sections.

### Point to point



Use this command to obtain accurate measurements of your data. GerbTool first prompts for a base point from which to measure. As you move the cursor away from the base point, GerbTool displays the distance in X and Y and true length in the prompt area. Choosing the left mouse button changes the base point to the current cursor position.

### Edge to edge



This command measures the actual minimum distance between two Gerber data items. GerbTool first prompts you to select a base item. As you select additional items, GerbTool displays the actual minimum distance between items in X and Y as well as true length in the prompt area.

*Highlight* 

Use this command to highlight all occurrences of a specified D-code. You can restrict your selection to flashes or draws or you can select both. You can also specify a particular layer. The selected D-codes remain highlighted until you turn off the highlight with the Highlights command (View menu) or 'H' shortcut key or when you select another group of items with this command.

*Copper* 

This command accurately calculates the amount of copper used on a layer using a high resolution bitmap method. GerbTool scans all visible layers.

*Extents* 

Use this command to determine the data extents of all loaded layers. In addition to displaying the extents information, GerbTool also updates its internal data extent information. This allows the All command (View menu) to correctly center the data after you've made edits to the database.


## Options menu

The Options command displays the Grid Snap, Ortho Line Snap, Arcs 360, Metric, and Configure commands.

### Grid Snap

Use this command to toggle grid snapping on or off. See the Configure command (Options menu) for information on changing the appearance of the grid.

---


 **Note** This command is also available as a shortcut key, CTRL+S.

---

### Ortho Line Snap

Use this command to toggle orthogonal snap mode on or off. When enabled, GerbTool forces all lines drawn interactively to the specified angle. See the Configure command (Options menu) for information on changing the snap angle.

---


 **Note** You can temporarily override the current setting by holding down the CTRL key.

---

### Arcs 360

Use this command to toggle interpolated arcs on or off. This setting affects the method of creating arcs used by the Arc Ctr, Arc 3 Pt, and Circle commands (Add menu). If enabled, GerbTool creates all arcs using 360° circular interpolation. If disabled, GerbTool creates all arcs using arcs of 90 or fewer degrees. This does *not* affect the way Gerber data is read from a file. It only pertains to adding new arcs with the above mentioned commands.

---

 **Note** Not all photoplotters support circular interpolation.

---

### Metric

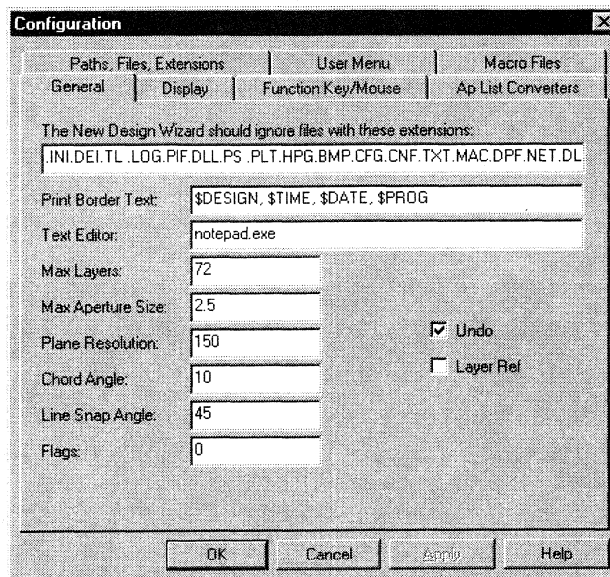
This menu item toggles metric mode on or off. When metric mode is enabled, GerbTool displays all information and editing fields that represent sizes and distances (for example, coordinates) in metric format.

## Configure

Use this command to change GerbTool's configuration and many of the default startup settings. When you choose this command, GerbTool displays a tabbed dialog box that contains the following topics: General, Display, Function Key/Mouse, Ap List Converters, Paths-Files-Extensions, Macro Files, User Menu.

### General

This tab displays a dialog box showing the current values of various general program settings.



*General Program settings.*

### The New Design Wizard should ignore files with these extensions

Enter any filename extensions that you know are not used for Gerber or aperture list files. The more extensions that GerbTool is able to ignore, the faster the New Design Wizard builds a new design file.

**Print Border Text** Specifies the text that appears in the border of check plots generated when printing and exporting to HPGL and PostScript. GerbTool looks for the key words \$DATE, \$TIME, \$DESIGN and \$PROG. If any of these keywords are found, GerbTool replaces them with the appropriate text. All other text specified appears in the border verbatim.

**Text Editor** Specifies the text editor that GerbTool starts when you are presented with a file to view or edit.

**Max Layers** Controls the number of layers that GerbTool can handle. The valid range of values is 12-999. Use the minimum value that satisfies your requirements in order to conserve memory.



**Note** This parameter will not become effective until the next time you start GerbTool.

**Max Aperture Size** Specifies the maximum aperture size that GerbTool creates for filled polygons.

**Plane Resolution** Specifies the “Dots Per Inch” resolution of the bitmap created when processing a pwr/gnd plane during netlist generation. To allow maximum speed, keep this value to a minimum. Default is 150 DPI.

**Chord Angle** Specifies the chord angle used when creating segmented arcs using editing commands. For example, a chord angle of 5° would result in a 18 separate line segments for a 90° arc.

**Line Snap Angle** Specifies the angle to which lines are forced if Ortho Line Snap is enabled.

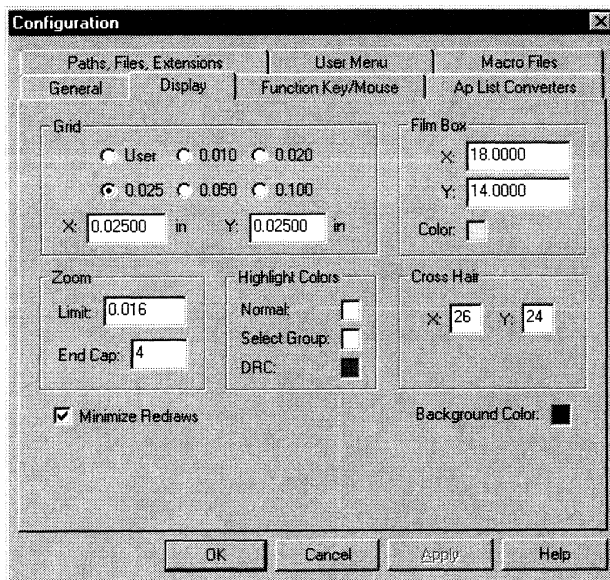
**Flags** Use this field to control some aspects of GerbTool's low level operations in the field. Typically you would be instructed by GerbTool Technical Support personnel on how to modify this parameter. The value is entered as a hexadecimal number.

**Undo** Specifies whether or not GerbTool retains “undo” information. If you disable this option, any current “undo” information is destroyed.

**Layer Ref** Determines whether or not ‘ref’ layer status is available. If you find that you don't use ref status you may disable it by deselecting this box.

## Display

This tab displays a dialog box showing the current settings that affect the GerbTool window.



*Display Parameters dialog box.*

**Grid** Specifies the grid size. You may select a pre-defined grid size or enter a value in the size X/Y fields.

**Film Box** Specifies the film box size and color. Change the current film box size by editing the size X/Y fields. Change the film box color by choosing the Color button.

**Zoom, Limit or End Cap** Use the Limit field to limit how far GerbTool can zoom in. On certain combinations of screen resolution and file format the display of items at extreme magnification can appear distorted. Use this setting to prevent this from occurring. The End Cap field specifies when GerbTool should stop attempting to draw end caps on drawn lines. If the thickness of a line (in pixels) is less than or equal to this parameter, no end caps will be drawn. Higher values provide decreased redraw times at minimum zoom levels.



**Note** The End Cap parameter only affects redraw speed and has no effect on your database.


**Highlight Colors** Use these buttons to control the colors used when highlighting database items.

**Cross Hair** The X and Y fields provide control over the size of the drawing area cross hair cursor. Enter 0,0 for a full screen cursor.



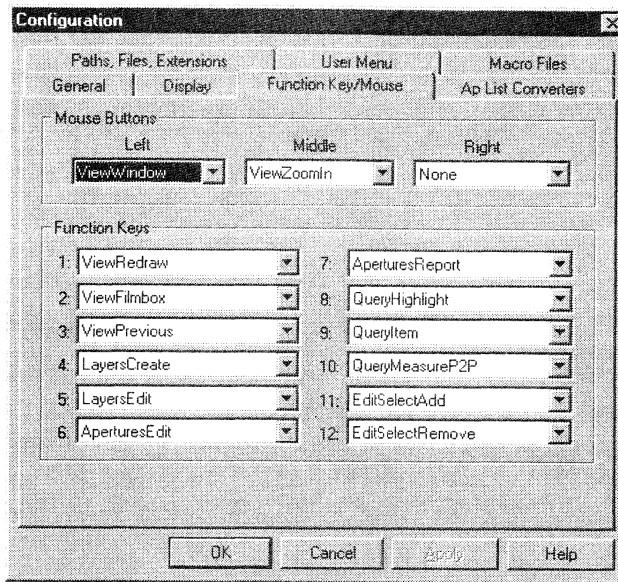
**Background Color** Use this button to change the Drawing Area background color. As with all color buttons within GerbTool, simply click on the button for a list of available colors.

**Minimize Redraws** This check box controls whether or not GerbTool minimizes the amount of redrawing it does. If you are an experienced user you may be comfortable redrawing the screen only when you want.

 **Note** Regardless of this setting, you may always interrupt a redraw, without affecting the current command, by pressing the [Esc] key.


## Function Key / Mouse

This tab displays a dialog box showing the current function key/mouse command assignments.



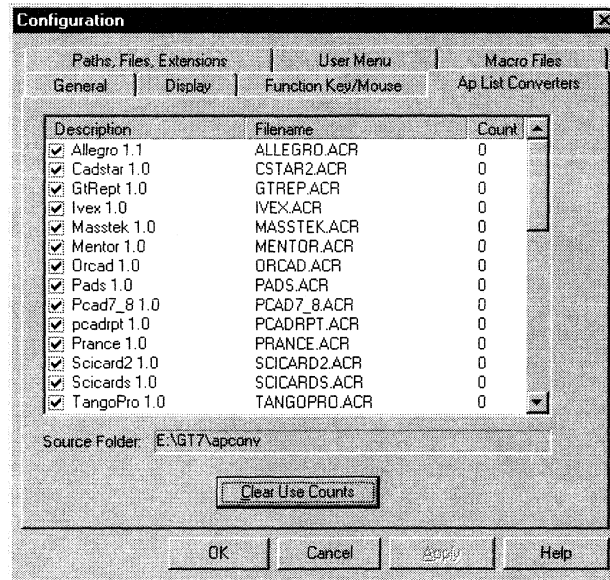
*Function key/mouse assignment dialog box.*

You can change any of the commands assigned to the mouse and function keys by selecting from the drop down lists. Any changes you make become effective immediately after choosing the OK button. This also saves the current key assignments so they are available the next time you start GerbTool.

 **Note** In addition to command names, you can also program function keys with GerbTool Macros allowing virtually all of GerbTool's power to be within one keystroke.

## Ap List Converters

This tab displays a list dialog box where you select the aperture list converters you require.



### Aperture list converters.

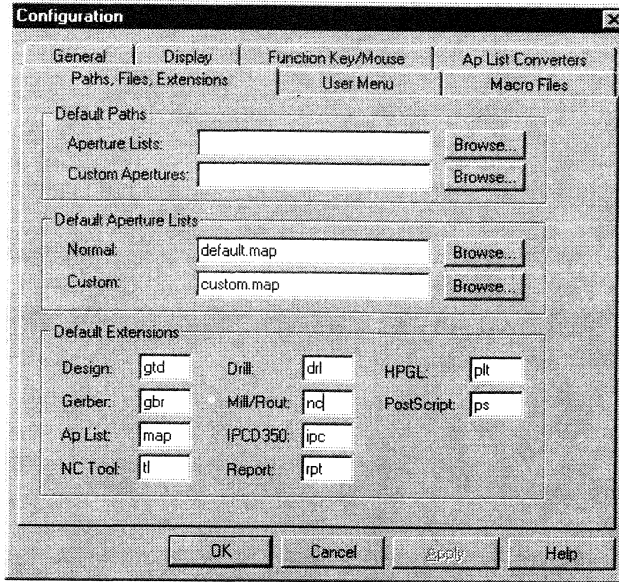
This list shows all aperture list converters found in the GerbTool program folder. The current converters are at the top of the list with a check mark by them. You can select or deselect converters as required. As a general rule, less is better. You can also change the title of a converter by clicking on an item in the list and then clicking on the title. You can then type in a new name. The use counts associated with each converter allows GerbTool to try the most popular converters first during automatic aperture list conversion. This can speed up the process considerably.



**Note** To add additional ACR files to GerbTool, simply copy them into the folder specified by the Source Folder field.

## Paths, Files, Extensions

This tab displays a dialog box containing edit fields for various program default values regarding paths, files, and extensions.



*Paths, files, and extensions.*

**Aperture Lists** This field tells GerbTool to look for all aperture list files, that don't have an explicit path, in this directory. This field replaces the GTMAPDIR environment variable used in previous versions of GerbTool.

**Custom Apertures** This field tells GerbTool to always look for custom aperture files (.cus) in this directory.

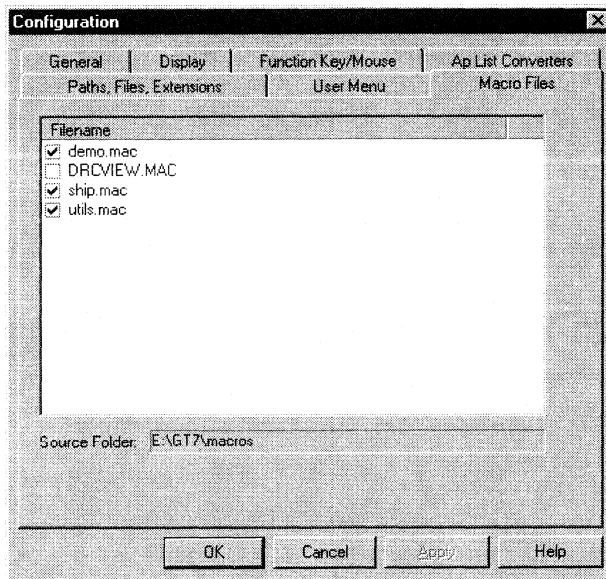
**Normal** This field specifies the default aperture list file that GerbTool loads if no other aperture list is specified.

**Custom** This field specifies the aperture list used by all custom aperture files (.cus) loaded.

**Default Extensions** This group of edit fields specify the default filename extensions for the indicated file types.

## Macro Files

This tab displays a list where you select which Macro files you want to load the next time you start GerbTool.



*Macro files.*

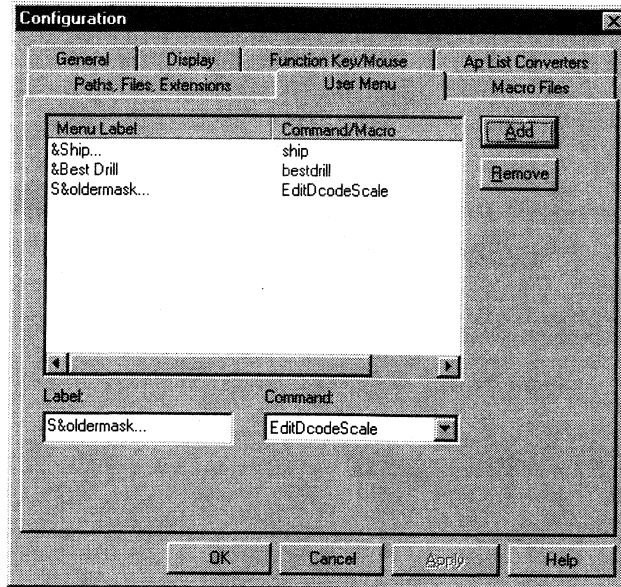
This list shows all Macro files found in the GerbTool program folder. You may select or deselect Macro files as required.



**Note** To add additional Macro files to GerbTool, simply copy them into the folder specified by the Source Folder field.

## User Menu

This tab displays the current user menu configuration so you can make changes as desired.



*User Menu setup.*

**Label** Use this field to type the menu item label text. A character prefixed with the ampersand (&) is considered the menu item hot key.

**Command** Use this list to select either a macro name or command name.

**Add** Choose this button to add a new item to the User menu. GerbTool uses the current value in the Label and Command fields to construct the menu item.

**Remove** Choose an existing menu item in the list, then choose this button to remove the item. If you add an item and then want to change it, you must remove it and add it again.

## Macro menu

The Macro menu includes the Run, Load, and Edit commands.

### *Run*

This command prompts you to select a macro to run. All macros loaded at program startup and through the Load command are available for execution.

### *Load*

Use this command to load additional macro files into GerbTool. You can include any macros present in the specified file in GerbTool's list of available macros.



**Note** To have GerbTool automatically load a Macro file at startup, see the Configure command described earlier in this chapter.


---

### *Edit*

This command prompts you to select a macro file to edit. GerbTool then loads the selected macro file into the currently configured text editor and displays it on the screen for you to edit.

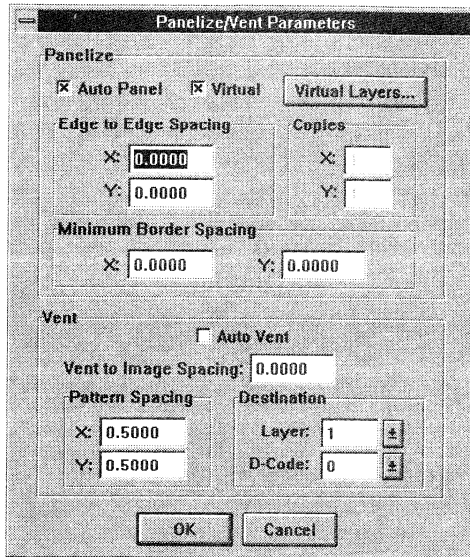
## Tools menu

The Tools command displays the Panelize, DRC, Snoman, Teardrops, Netlist, Test Points, Pad Removal, NC Drill, Vent, Convert, Lyr Spread, Fix SS, and Macros commands.

 **Note** The Test Points tool is not available with OrCAD Layout for Windows.

### Panelize

The Panelize command creates multiple copies of a design in one film box. This allows multiple copies of the design to be manufactured as one panel. When you choose the Panelize command, GerbTool displays the Panelize editing dialog box.



*Panelize editing dialog box.*

### To perform automatic panelization

Automatic panelization provides automatic placement of the maximum number of copies on a single panel.

The Edge-to-Edge values set the minimum distance between adjoining edges of the image copies in the X- and Y-axes. The Minimum Border Spacing values set the minimum distance between the edges of the image copies and the edge of the panel.



**Note** Although GerbTool copies only visible layers, all layers of the original image remain aligned after panelization.

---

Follow these steps to perform automatic panelization:

- 1 Activate those layers you want to include in the panelization.
- 2 Select the Panelize command from the Tools menu. GerbTool displays the Panelize editing dialog box.
- 3 Select the Auto Panel button.
- 4 Enter the Edge to Edge and Minimum Border Spacing values.
- 5 Choose the OK button.


GerbTool automatically calculates the maximum number of images that fit inside the current film box, then previews the panel layout. After asking for confirmation, GerbTool completes the panelization process. Depending on the setting of the Virtual button, GerbTool either copies the proper number images into the database or notes the number of copies and their location for display purposes.



### To perform manual panelization

Use manual panelization to control the layout of the copies on the panel. The Copy values set the number of copies in the X- and Y-axes. The Point to Point values set the minimum distance between adjoining edges of the image copies in the X- and Y-axes. The Minimum Border Spacing values set the minimum distance between the edges of the image copies and the edge of the panel.

---

 **Note** The Point-to-Point option appears only when Auto Panel is deselected.

---

To perform manual panelization, follow these steps:

- 1 Activate those layers you want to include in the panelization.
- 2 Select the Panelize command from the Tools menu. GerbTool displays the Panelize editing dialog box.
- 3 Deselect the Auto Panel button.
- 4 Enter the number of rows and columns in the Copy fields.
- 5 Enter the Edge to Edge and Minimum Border Spacing values.
- 6 Choose the OK button.
- 7 Draw a selection box around the area you want to copy.

GerbTool previews the panel layout. After asking for confirmation, GerbTool completes the panelization process. Depending on the setting of the Virtual button, GerbTool either copies the proper number images into the database or notes the number of copies and their location for display purposes.

### **To perform automatic venting**

Automatic venting defines the shape and placement of flashes on the panel outside the image areas. GerbTool adds the pattern spacing and aperture selection to the database.

Automatic venting can occur during panelization, regardless of whether or not the panelization is automatic. Venting may be targeted to any layers in the Gerber file.

The Vent to Image Spacing field sets the spacing between the image copies and the venting area. The Pattern Spacing field sets the spacing between the flashes in the vent pattern.

The D-Code field sets the size and shape of the flashes. In both automatic and manual venting, the style of vent pattern is customized using custom apertures. For example, you can create a hatch or cross-hatch pattern using a diagonal or cross shape custom aperture. Be sure to set the height and width of the overall size of the custom aperture in the aperture list.

To perform automatic venting, follow these steps:


- 1 Check the Auto Vent button within the Panelize editing dialog box.
- 2 Enter the Vent to Image Spacing values and the Pattern Spacing values.
- 3 Enter the D-Code values.

After panelization, GerbTool fills the specified area with the defined pattern of flashes.

## To perform virtual panelization

Virtual Panelization allows GerbTool to panelize your design without actually duplicating layer data in the database. Virtual panelization provides many benefits including automatic updating of all images during edits and drastically reduced file sizes. Furthermore, if you want to plot your design on an extended Gerber or FIRE9xxx compatible plotter, GerbTool automatically inserts the proper step and repeat codes into your Gerber data.

---

 **Note** Although no data is duplicated during virtual panelization, the data origin is modified to center the images within the panel. Therefore, it is still necessary to save your design after panelization.

If your designs are plotted on a plotter that does not support step-and-repeat codes, you must execute the Panelize command without the Virtual button selected and save your panelized Gerber files before you send them to the plotter.

---

To perform virtual panelization, follow these steps:

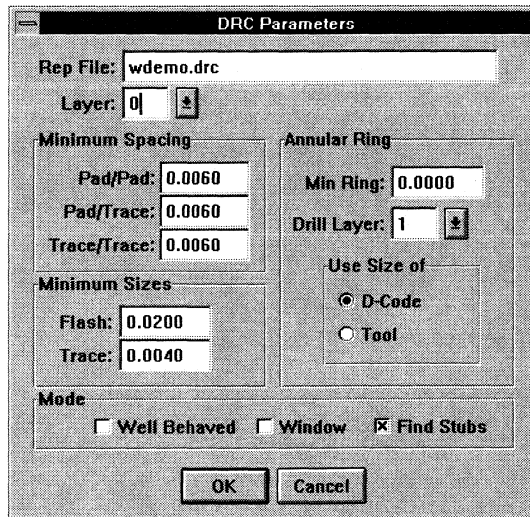
- 1 Activate those layers you want to include in the panelization.
- 2 Select the Panelize command from the Tools menu. GerbTool displays the Panelization editing dialog box.
- 3 Select the Virtual Layers button. GerbTool displays a dialog box listing the loaded layers.
- 4 Select those layers you want to include in the virtual panelization and choose the OK button.
- 5 Proceed with either automatic or manual panelization.

GerbTool modifies the database with the number of copies and their location for display purposes. Virtual panel mode (and hence the display of virtual panels) may be toggled on or off using the CTRL+V shortcut key.

GerbTool also inserts step and repeat codes into NC Drill output data if you select the Virtual button. This may be necessary to drill large panels if your NC equipment is memory limited.



The DRC command checks to see that your design meets minimum item to item spacing requirements. When you select DRC from the Tools menu GerbTool displays the DRC Parameters dialog box.




*DRC Parameters dialog box.*

**Report File** This is the file to which all errors are logged. Enter a valid filename to set the filename for the DRC error file.

**Layer** This is the target layer for the DRC. By default, GerbTool uses the current active layer in the Layer field. You may override this by entering a different layer.

---

 **Note** If you enter 'all' or '0' in this field, GerbTool processes all viewed layers.

---

**Highlight Layer** Specifies an optional layer in which GerbTool copies all database items that are part of a DRC error. That way, you can easily see the whole picture and print all errors at once. To disable this feature enter 'none' or '0' in this field.

**Border Layer** Specifies the layer that contains the design border. If you have a layer with just the border on it you may specify this layer here. To disable border spacing checks enter 'none' or '0' in this field.

**Pad/Pad** Specifies the minimum spacing allowed between pads.

**Pad/Trace** Specifies the minimum spacing allowed between pads and traces.

**Trace/Trace** Specifies the minimum spacing allowed between traces.

**Border** Specifies the minimum spacing allowed between any item and the border specified in the Border Layer field.

**Flash** Specifies the minimum pad size.

**Trace** Specifies the minimum trace size.

**Min Ring** Specifies the minimum annular ring required. The Annular Ring option compares the diameter of the flash on the DRC layer to the diameter of the flash on the drill layer with the assumption that the drill layer normally contains a flash at each pad location using a smaller size than the DRC layer. The Min Ring value is the difference in diameters. Select whether the size of the drill layer flashes are taken from the D-code size or the Tool size. Through-hole pads that do not have a corresponding drill flash are reported as "missing" drills. If either the Min Ring value or drill layer are 0, GerbTool does not perform the annular ring.



**Tip** You can also use the annular ring check to verify a soldermask layer. Use the soldermask layer as the active layer; use the copper layer on which the pads are defined as the drill layer.

**Drill Layer** Specifies the drill layer used in the annular ring check. A '0' disables annular ring checking.

**Use Size Of** Specifies whether the size of the drill layer flashes are taken from the D-code size or the Tool size field within the aperture list when performing annular ring checking.

**Well Behaved** The DRC command supports two separate modes; "well behaved" and normal. In "well behaved" mode, GerbTool assumes that legal pad/trace or trace/trace connections have common X-Y locations (see the Netlist command in Chapter 7 for a description of "well behaved" Gerber files). This means that *any* actual contact between items that don't share a common X-Y location, and are in different nets, is considered a violation. Conversely, in normal mode, any actual contact between items is not considered a violation. Only items that are not in contact but are within the minimum spacing rules are considered in violation. "Well behaved" mode is preferred if your Gerber files were produced accordingly as it provides much faster processing and more accurate results.



**Note** If a valid netlist does not already exist, GerbTool prompts you to generate one. While a netlist is not a prerequisite to DRC, a netlist increases the usefulness and accuracy of DRC.

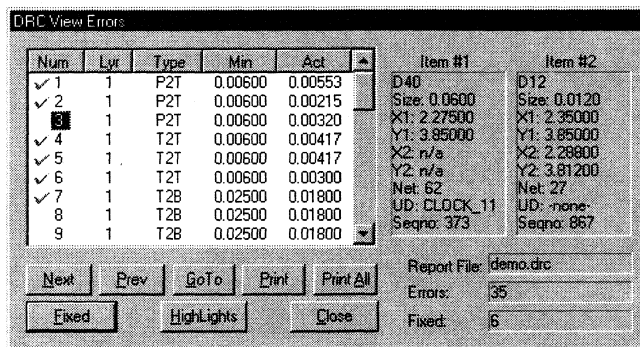
**Window** Specifies that GerbTool runs the DRC on a particular window of data, rather than on the complete layer.

**Find Stubs** Allows GerbTool to locate and highlight all trace stubs. A trace stub is any trace that touches a pad or trace on only one end.

**Max Errors** Sets a limit to the number of errors DRC produces. This prevents DRC from generating a huge report file if you enter the incorrect spacing rules for a given design.

**Don't check items in select group** Tells DRC to ignore all items that are in the current select group. For example, if you have blocks of text that you want to exclude from the check, you can use the Add command to create a select group with these blocks of text. DRC then ignores the text, thereby producing a much cleaner report.

After performing the DRC, GerbTool enables DRC error viewing and displays the DRC View Errors dialog box as show below:



*DRC view errors.*

You can use this dialog box to examine and document any or all DRC errors found.

**Next** Jump forward to the next non-fixed error.

**Prev** Jump back to the next non-fixed error.

**GoTo** Jump to the current error fixed or not.

**Print** Print the currently selected error on the default Windows printer. The error report includes a screen capture of the error and all related information about the error and items involved.

**Print All** Prints a separate error report for each error in the list.

**Fixed** Toggles the status of an error. By setting the status to fixed you can tell at a glance which errors have already been corrected.

**HighLights** Toggles the display of the current error highlights. Use this button to temporarily turn off highlights to allow easy correction of the error.

Use the Errors command to toggle the display of the DRC View Errors dialog box.



**Note** GerbTool updates the DRC report file generated by the last run of the DRC command with the fixed status of each error. Therefore, you should not remove or substantially alter this file if you intend to view DRC errors again using this file.

## Snoman

This menu selection starts the Snoman tool. The Snoman tool creates a *maximum material condition* at the point of trace entry into a pad. See *Appendix D: Snoman concepts* for a more technical description.

*Snoman editing dialog box.*

Enter a valid filename in the Rep File field; GerbTool logs any errors to this file. You must specify a layer to operate on (From Layer) as well as an output layer (To Layer) for the generated Snoman pads.

---

**Note** If you enter a zero in the From Layer field, GerbTool processes all viewed layers, with the resultant Snoman pads being added to their respective layers.

---

You can restrict the generation of Snoman pads to a particular D-code by entering its identifier in the D-Code field. A D-code of zero matches all. Edit the spacing parameters to specify the design rules to which Snoman must adhere. The Host Offset field contains the offset maintained between the host pad centroid and the edge of the generated Snoman pad. This value may be negative. If Snoman detects a spacing rule violation while placing a Snoman pad, it reduces the size of the Snoman pad to avoid such errors. You can control the percentage of the host pad size that Snoman can reduce the size of the Snoman pad. Use the Min Percent field to specify this value. Use the Max Percent field to control the maximum size of the generated Snoman pad as a percentage of the host pad size. You can also indicate whether Snoman should operate on a window of data versus a complete layer.

---

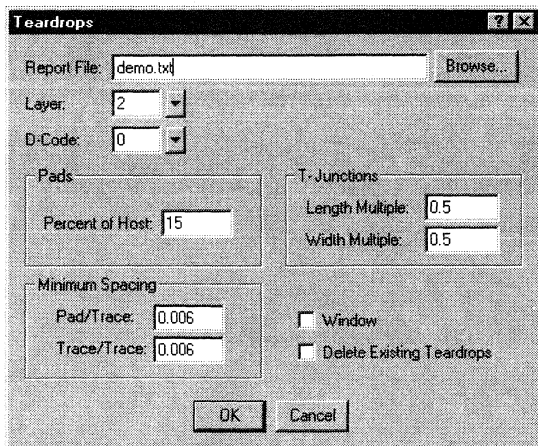
**Note** If a valid netlist does not already exist, GerbTool prompts you to generate one. A netlist is required for the Snoman tool to work properly.

---

Use the Errors command (View menu) to view potential rule violation errors, if any, after executing this command.

## Teardrops

The Teardrop tool creates a *maximum material condition* at the point of trace entry into a pad, or at T junctions for traces. That is, when you use the Teardrops command, GerbTool adds copper at the point in question to insure that the connection is maintained.



Teardrop dialog box.

**Report File** Specifies the file to which GerbTool writes any errors for the Teardrops command.

**Layer** Specifies the layer that receives the teardrops.



**Note** If you enter a '0' in the Layer field, GerbTool processes all viewed layers.

**D-Code** Specifies that GerbTool add teardrops only to those pads that use the specified D-Code. A D-Code of '0' matches all.

**Percent of host** Specifies the relative size of the teardrop "tail." GerbTool calculates this length as a percentage of the diameter of the host pad. The value you enter can be greater than 100%.

**Pad/Trace** Specifies the pad to trace spacing parameters for teardrops.

**Trace/Trace** Specifies the trace to trace spacing parameters for teardrops.

**Length Multiple** Specifies the "T" junction tail length. GerbTool calculates this value in multiples of the host trace diameter. The value you enter can be fractional.


**Width Multiple** Specifies the "T" junction tail width. GerbTool calculates this value in multiples of the host trace diameter. The value you enter can be fractional.

**Window** When you select this option, GerbTool creates teardrops for pads and "T" junctions only within a particular window, not for the entire design.



**Delete Existing Teardrops** When you select this option, GerbTool removes existing teardrops from the design (or window within the design) before creating the new teardrops.


---

 **Note** If a valid netlist does not already exist, you will be prompted to generate one now. A netlist is required for the Teardrop tool to work properly.

---

All pad locations for which GerbTool could not generate a Teardrop are highlighted and their locations are specified in the generated report file.

---

 **Note** You can remove teardrops using the Undo command.

---

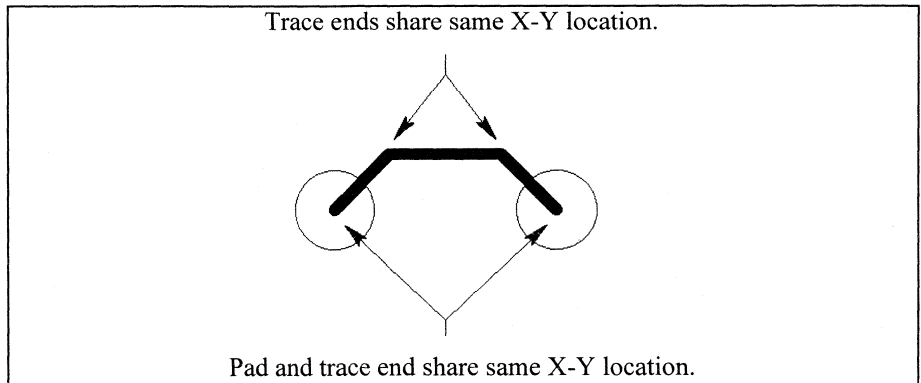
## Netlist

The Netlist item displays the Generate and Save commands.

### Generate

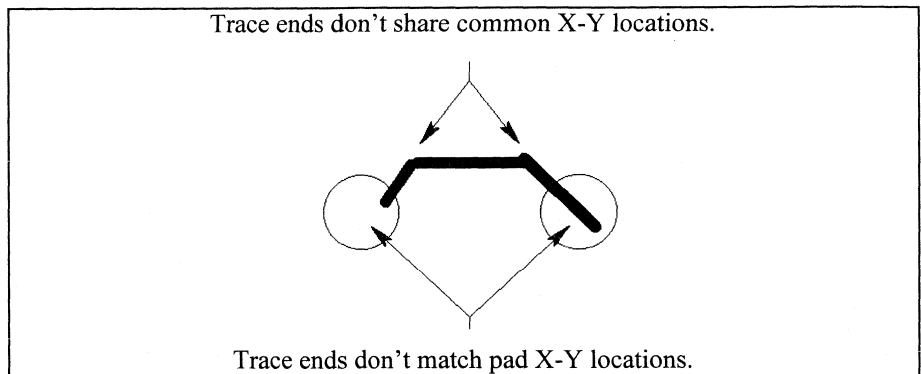
The Generate command processes all viewed layers and creates a single multi-layer netlist that becomes part of the internal database. The netlist may then be used by other GerbTool Test Points commands that require a netlist such as DRC and Snoman (Tools menu).

You can indicate whether your database is “Well Behaved” or not. A well behaved Gerber file is defined as a file where all items that are to be considered connected share a common X-Y location as shown below:



*Example of a “well-behaved” Gerber file.*

The following is an example of a Gerber file that is *not* well-behaved:



*Example of a Gerber file that is not “well-behaved.”*

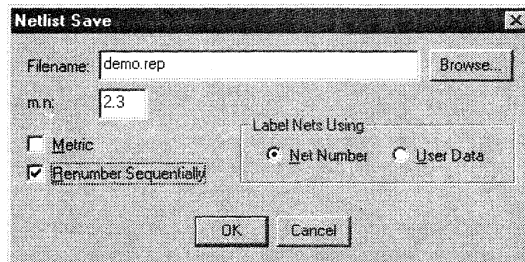
If you determine that your Gerber files are indeed “well-behaved,” choose this mode when generating a netlist, as there is a dramatic increase in processing speed due to the well-behaved nature of the Gerber files.

Since so many of GerbTool’s features require a netlist to perform properly, you can save the generated netlist within your Gerber files for later use. If netlist saving is enabled (see *Chapter 7: Command reference*), and a netlist is present, GerbTool saves it when the layer is saved. To remove a netlist from a Gerber file, load the layer (or layers), disable netlist saving using the Format command (Files menu) and then save the necessary layers.

**Note** GerbTool uses the G04 command to embed a netlist within a Gerber file. This causes the Gerber file to increase slightly in size. Remove netlists as described above before submitting your files to be photoplotted, due to their increased size and the possibility of the photoplot equipment not properly recognizing the G04 command.

## Save

This command generates an ASCII netlist file consisting of net numbers, or names using UserData, and pad X-Y coordinates. See *Appendix C, Sample Netlist File*. This command uses all viewed layers in generating the netlist file.



*Netlist editing dialog box.*

**Filename** Specifies the desired output filename. Use the Browse button to locate the desired output file.

**M.N** Specifies the coordinate format.

**Renumber Sequentially** Instructs GerbTool to renumber the net numbers if needed to make sure that they are output in order and with no gaps in the net numbers.

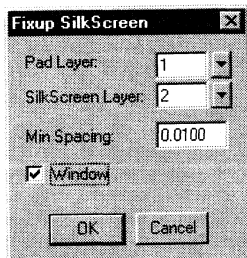
**Label Net Using** Provides a means of determining how GerbTool labels nets. If your nets have UserData assigned to them you may choose to have your netlist labeled with the UserData instead of net numbers.

**Note** If a valid netlist does not already exist, GerbTool prompts you to generate one. A netlist is required for this command to work properly.

## Fix SilkScreen

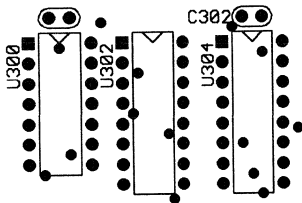


This command automatically moves silkscreen data away from pads.

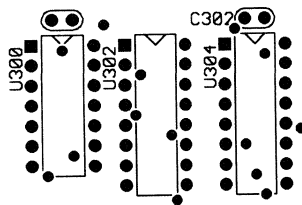


*Fix silkscreen editing dialog box.*

You specify the layer that contains the pads (Pad Layer) and the layer that contains the silkscreen data (SilkScreen Layer) as well as a minimum spacing to be maintained and finally whether you desire window mode. GerbTool then moves line segments that violate the minimum spacing requirement as shown in the following before and after sequence:



*Before running the Fix SilkScreen command.*




*After running the Fix SilkScreen command.*

## Pad removal

The Pad Removal command displays the Isolated and Stacked commands.

### Isolated

Selecting this command removes any unused pads (isolated/floating pads) from inner layers.

 **Note** GerbTool considers only Inner-type layers. Use the Edit command (Layers menu) to change this if necessary for a particular layer.

GerbTool does not remove targets or thermal pads. You specify the layer from which to remove the pads and whether GerbTool processes a window or the entire layer.

### Stacked

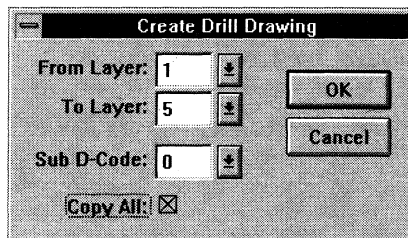
Selecting this command removes any unnecessary pads that are identical and stacked exactly one on top of another on the same layer. You specify the layer from which to remove the pads and whether GerbTool processes a window or the entire layer.

## Drill

The NC Drill command displays the Drawing and Save commands.

### Drawing

This command creates a drill drawing using the *Legend* field associated with each D-code in an aperture list.

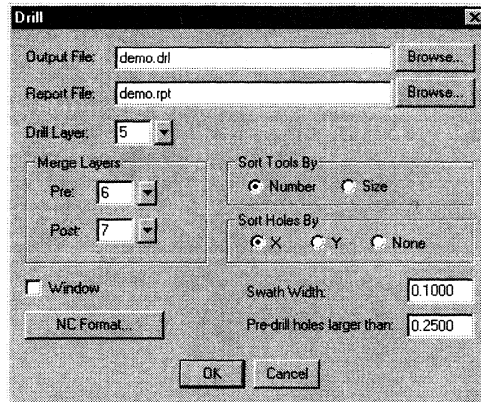


*Create Drill Drawing dialog box.*

For each D-code in the From Layer field, GerbTool adds the D-Code specified by the corresponding Legend field to the layer specified in the To Layer field. Use the Copy All option to copy D-Codes with invalid Legends. If you select the Copy All option, you can use the Sub D-Code field to specify a particular D-Code to use as a substitute for invalid Legend D-Codes. If Sub D-Code is '0', all D-Codes with an invalid Legend D-Code use the original D-Code value in the To Layer field.

## Save

The Drill command creates an ASCII output file containing X-Y pad locations in the selected NC format. GerbTool optimizes the output and removes duplicate hits within a single tool.



*Drill editing dialog box.*

This command relies on the Tool assignments within the aperture list for the selected input layer. Optimization is controlled by the Swath Width value and by whether an X or Y sort is performed. The report file contains an approximate distance that the drill head will travel. Therefore, by adjusting the swath width and examining the report file you can achieve the fastest drilling throughput.



**Note** Perform image panelization prior to executing this command. If you perform virtual panelization the output of this command contains step and repeat codes. Use these codes if your drilling equipment has limited memory capacity. Otherwise, a fully optimized non-virtual panel results in more efficient drilling.

Occasionally there may be items that you don't want optimized, but that you want to include in the same drill file, such as test coupons and mounting holes. Place these items on a layer in the exact order that you want them drilled. GerbTool enters these layers into the Merge Layer Pre and Post fields. GerbTool inserts these layers into the drill file without optimization before and after inserting the optimized information from the layer specified in the Drill Layer field. This is done on a tool by tool basis. Therefore, GerbTool outputs information for tool #1 on the Pre merge layer first, then optimizes and outputs the drill layer, followed by the tool #1 information from the post merge layer. GerbTool then repeats the process for tool #2, and so on. This also works for Virtual panels when you want to include non-panelized drill data.

**Sort Tools By - Number or Size** Indicates how GerbTool orders the tools in the output file.

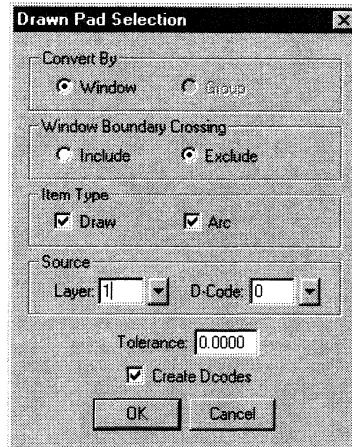
**Pre-drill holes larger than** Indicates the maximum size for holes that will not be pre-drilled.

## Convert

The Convert menu item presents a sub-menu with the Drawn Pads and Arcs commands.

### Drawn Pads

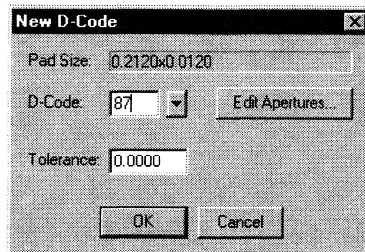
Use this command to convert pads created with Gerber draws into flashes. Use this command prior to attempting any other editing or data extraction such as NC Drill. This command can significantly decrease the size of your database if it contains drawn pads.



*Drawn Pad Conversion dialog box.*

This command prompts you to enter a window around the drawn pad you want to convert.

If you select the Create D-Codes check button, GerbTool creates new D-Codes as necessary to match the dimensions of the drawn pads selected for conversion. If you don't select the Create D-Codes check button, GerbTool informs you of the calculated size of the pad as shown below:



*Drawn Pad Replacement D-Code dialog box.*

Find or create a corresponding flash in the aperture list for this layer. Enter the appropriate D-Code in the D-Code field and a tolerance value, if needed, in the Tolerance field.



---

**Note** The tolerance value allows GerbTool to increase its match frequency when the CAD system that generated the drawn pads exhibits round off errors. Usually a value of 0.002 (inches) will suffice.

---

GerbTool locates and highlights all occurrences of any matching drawn pads and prompts you on whether or not to continue.

### Arcs

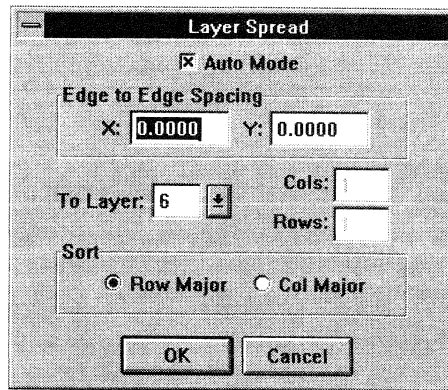


This command converts circular interpolated circles into segmented circles, individually or by window. Use this command if your photoplotter can't handle circular interpolated arcs.



## Layer Spread

Use the Layer Spread command to reduce your film costs by automatically copying and spreading all viewed layers onto one layer (and thus one sheet of film).



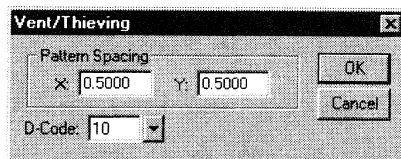
*Layer Spread editing dialog box.*

You can select automatic or manual mode using the Layer Spread editing dialog box as shown above. If you select Auto Mode, GerbTool automatically calculates the number of images that will fit in the film box as well as the position of each image. In auto mode the X and Y spacing fields specify the opposing border-to-border minimum spacing requirements. In manual mode, you must specify the number of rows and columns and the center to center spacing in the X and Y spacing fields. In either case, you can select either row major or column major placement. While the To Layer field may specify one of the layers to be spread, it usually is an empty layer created to accept the properly spread out images.

When you choose the OK button, GerbTool prompts you to select the order in which the layers are spread. You must click on each layer to define the proper order. After doing so, GerbTool shows the placement of all layers for your approval. If you respond affirmatively, GerbTool copies and spreads the layers as shown.

## Vent

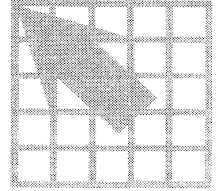
Use this command to manually add Venting/Thieving patterns to your database. GerbTool displays the Vent Parameters edit dialog box where you can edit the venting parameters such as pattern spacing and aperture selection.



*Vent Parameters Edit dialog box.*

You can then define a rectangular area by entering two coordinate points. After confirmation, GerbTool fills the specified area with a pattern of flashes as specified.






# Macros

GerbTool provides a powerful macro command language that you can use to create new commands to accomplish everything from simplifying repetitive tasks to implementing entirely new functions. The macro language provides the ability to make decisions, repetitively execute a group of commands, scan the database, prompt the user for data and more. This chapter details the command language used in creating a GerbTool macro.

## Creating a macro

You create macros using any text editor that supports plain ASCII text.

---

 **Note** If you use a word processor you may have to specifically save your macro file in ASCII text format.

---

A macro file can contain multiple macros, each with a name up to sixteen characters long. If a macro is defined more than once, only the last occurrence of the macro is visible to GerbTool. While any number of macro files can be loaded, GerbTool allows a maximum of 1024 total macros to be defined.

Each macro consists of a **MACRO** statement followed by one or more macro language commands or statements and terminated by an **END** statement as shown below:

```
MACRO testmacro
.
.
.
ENDMACRO
```

## Using variables

You can define macro variables with names up to 32 characters. You can use a variable anywhere a number or text string is expected within a macro. To use a variable, prefix a symbolic name with a \$, as shown in the following example:

```
GETPOINT "Enter New Coord", $XCOORD, $YCOORD
CALC      $XCOORD = $YCOORD * 2.0
ADDFLASH $XCOORD, $YCOORD + 0.5
```

GerbTool automatically defines many system level variables that provide basic program information, such as the number of layers configured, and variables that contain the specific results of certain macro commands after they have been executed. All system level variables are defined with an additional leading dollar sign (\$) character, such as \$\$STATUS.



**Note** The system variable \$\$STATUS is set by most commands, indicating the success or failure of the command. Positive values indicate success; negative failure. In some cases the value may indicate a count, such as in the Copy command (Edit menu).

A list of system level variables is shown below.

\$\$ACTIVE_LAYER	\$\$GROUPMODE	\$\$PLATFORM
\$\$CALLDEPTH	\$\$ITEMMODE	\$\$RECTANGLE
\$\$COMPLEX	\$\$LAYERMODE	\$\$ROUND
\$\$CURRENTDCODE	\$\$MAXLAYERS	\$\$SELGRPCNT
\$\$CUSTOM	\$\$MODEL	\$\$SQUARE
\$\$DIAMOND	\$\$NO	\$\$STATUS
\$\$DONUT	\$\$OBLONG	\$\$TARGET
\$\$DRAW	\$\$OCTAGON	\$\$THERMAL
\$\$DRILLCOUNT	\$\$PANELCOLS	\$\$THERM45
\$\$DRILLEFTRAVEL	\$\$PANELROWS	\$\$TRUE
\$\$DRILLTRAVEL	\$\$PANELXOFF	\$\$VERSION
\$\$DSNNAME	\$\$PANELXSPACING	\$\$WINDOWMODE
\$\$FALSE	\$\$PANELYOFF	\$\$YES
\$\$FLASH	\$\$PANELYSPACING	

You can also create arrays of variables using the `ARRAY` and `STRARRAY` functions. An array is a list of variables referenced through a single variable name and an index. Array indexes can be any expression including another array variable. For example:

```
STRARRAY $names (3)
STRCPY $names (1), "Bob"
STRCPY $names (2), "Betty"
STRCPY $names (3), "Jessie"
```

## Coordinate lists

For added ease of use, you can specify a range of array indexes when using numeric arrays in coordinate lists submitted to a macro command. For example, `ADDFLASH $xy (1, 4)` would be equivalent to `ADDFLASH $xy (1), $xy (2), $xy (3), $xy (4)`. This is especially useful when using variables as indexes.

Coordinate lists also support both *absolute* and *relative* coordinate modes. The default coordinate mode is absolute. To change to relative mode, prefix a coordinate with an `R`. Once activated this way, all coordinates following are relative to the previous coordinate. You can turn relative mode off by prefixing a coordinate with an `A`. The specified mode only applies to the command in which it was used.

For example:

```
ADDDRAW $xs, $ys, R0.1, 0, 0, 0.1, -0.1, 0, A$xs, $ys
```

In the above example the `ADDDRAW` command is provided an initial absolute coordinate followed by three relative coordinates and finally a absolute coordinate.

## Repeating blocks of commands

Macros allow you to repeat a series of commands until an event occurs that terminates the loop. An example of the `REPEAT` statement is shown below.

```
REPEAT $cnt > 0
    ...will be executed as long as $cnt > 0...
END
```

The above example repeatedly executes the commands between the `REPEAT` and `END` statements until the variable `$cnt` is less than or equal to zero. You can nest `REPEAT` blocks.

## *Making decisions*

Macros also have conditional capabilities, using the IF statement:

```
IF $shape == $$RECTANGLE
    ...will be executed if above test is TRUE...
END
IF $shape == $$RECTANGLE
    ...will be executed if above test is TRUE...
ELSE
    ...will be executed if above test is FALSE...
END
```

Combining IF statements within REPEAT blocks provides virtually unlimited macro programming possibilities.

## **Loading macros**

You can load a macro file using the Load command (from the Macro menu item on the Tools menu). Use this command to load a macro file after GerbTool is up and running.

## **Running macros**

There are several methods of running a macro:

- The first method is to use the normal menu and select the Run command (Macro menu). Use this command to choose a previously loaded macro for execution.
- The second method is to use the M shortcut key.
- Another method is to assign a macro to a function key using the KeyCmds command from the Options menu. This allows you to run a macro with a single keystroke.

## Macro language reference

This section describes each macro command and statement, and the parameters expected for each.

### *Conventions used*

...	Three dots (an ellipsis) indicate that additional similar parameters are allowed.
[x, y]	Items inside square brackets indicate optional parameters.
<a   b>	Angle brackets and vertical bars indicate a choice among two or more items.
literal	A literal numeric or string value, such as 14.125 or Yes.
variable	A numeric or string value stored in a variable, such as \$XCOORD or \$REFDES.
numvar	A numeric variable.
strvar	A string variable.
operator	One of the following mathematical or comparison operators: <ul style="list-style-type: none"> <li>+ addition</li> <li>- subtraction</li> <li>/ division</li> <li>* multiplication</li> <li>== equal</li> <li>!= not equal</li> <li>&lt; less than</li> <li>&gt; greater than</li> <li>&lt;= less than or equal</li> <li>&gt;= greater than or equal</li> </ul>
exp	Numeric expression of the form: <literal   numvar> [operator <literal   numvar>]
yesno	<"Yes"   "No"   \$\$YES   \$\$NO>
\	Use the back slash as the last character on a line to indicate that a long command is to continue on the next line.
#	This character initiates a comment.

*Add functions***ADD3PTARC**

Purpose	Allows the user to enter arcs by specifying two end points and a point on its circumference.
Menu command	Arc 3 Pt (Add item on the Edit menu)
Syntax	ADD3PTARC [x1, y1, x2, y2, x3, y3] ...
Parameters	
x1	The x coordinate of the first end point of the arc.
y1	The y coordinate of the first end point of the arc.
x2	The x coordinate of the second end point of the arc.
y2	The y coordinate of the second end point of the arc.
x3	The x coordinate of a point on the circumference of the arc.
y3	The y coordinate of a point on the circumference of the arc.
Description	This command adds three-point arcs into your Gerber layer. GerbTool draws the arc on the active layer using the current D-code in a counter-clockwise direction. The arc is created either as a 360° interpolated circle or with multiple line segments, depending on the style that is currently active. Note that you can use this command to draw multiple arcs with a single call by passing all of the coordinates for all of the arcs to the command.
Example	The following example adds an arc to layer 1 using D-Code 10, whose end points are at (0, 0) and (1, 0) and passes through (0.5, 0.5).  <pre> ACTIVE LAYER      1 CURRENT D CODE    10 ADD3PTARC         0, 0, 1, 0, 0.5, 0.5 </pre>
See also	ADDARC, ADDCIRCLE



**ADDARC**

Purpose	Allows you to enter arcs by specifying the center, and two points defining the starting and ending angles.
Menu command	Arc Ctr (Add item on the Edit menu)
Syntax	ADDARC [x1, y1, x2, y2, x3, y3]...
Parameters	
x1	The x coordinate of the center of the arc.
y1	The y coordinate of the center of the arc.
x2	The x coordinate of the starting point of the arc.
y2	The y coordinate of the starting point of the arc.
x3	The x coordinate of the ending point of the arc.
y3	The y coordinate of the ending point of the arc.
Description	Use this command to add an arc to your Gerber layer. GerbTool draws this arc on the active layer using the current D-code in a counter-clockwise direction. The arc is created either as a 360° interpolated circle or with multiple line segments, depending on the style that is currently active. Note that you can use this command to draw multiple arcs with a single call by passing all of the coordinates for all of the arcs to the command.
Example	The following example adds an arc to layer 1 using D-code 10, with a center at 0,0 and end points at -1,0 and 1,0.  <pre> ACTIVELAYER      1 CURRENTDCODE     10 ADDARC           0,0, -1,0, 1,0 </pre>
See also	ADD3PTARC, ADDCIRCLE

**ADDCIRCLE**

Purpose	Allows you to draw a circle by specifying its center, and a point defining its radius.
Menu command	Circle (Add item on the Edit menu)
Syntax	ADDCIRCLE [x1, y1, x2, y2] . . .
Parameters	
x1	The x coordinate of the center of the circle.
y1	The y coordinate of the center of the circle.
x2	The x coordinate of a point on the radius of the circle.
y2	The y coordinate of a point on the radius of the circle.
Description	Use this command to add a circle to your Gerber layer. GerbTool draws this circle on the active layer using the current D-code in a counter-clockwise direction. The arc is created either as a 360° interpolated circle or with multiple line segments, depending on the style that is currently active. Note that you can use this command to draw multiple circles with a single call by passing all of the coordinates for all of the circles to the command.
Example	The following example adds a circle to layer 1 using D-code 10, with a center at 0,0 and a radius of 3 inches.  <pre> ACTIVELAYER      1 CURRENTDCODE     10 ADDCIRCLE        0, 0,  3, 0 </pre>
See also	ADD3PTARC, ADDARC

**ADDDRAW**

Purpose	Allows you to draw a line by specifying its end points.
Menu command	Draw (Add item on the Edit menu)
Syntax	ADDDRAW [x1, y1, x2, y2] ...
Parameters	
x1	The x coordinate of the starting point of the line.
y1	The y coordinate of the starting point of the line
x2	The x coordinate of the ending point of the line.
y2	The y coordinate of the ending point of the line.
Description	Use this command to add a line or trace into your Gerber layer. GerbTool draws this line on the active layer using the current D-code. Note that you can use this command to draw multiple lines with a single call by passing all of the coordinates for all of the lines to the command. When drawing more than one line, you only need to pass the ending points of subsequent lines. GerbTool automatically starts each line at the endpoint of the previous line.
Example	<p>The following example creates a 1 inch box out of 4 lines, whose lower left corner is at the 0,0 coordinate. Note how only the changed coordinates need to be entered.</p> <pre> ACTIVELAYER      1 CURRENTDCODE     10 ADDDRAW          0,0, 0,1, 1,1, 1,0, 0,0 </pre>

**ADDFILL**

Purpose	Allows you to perform a polyfill.
Menu command	Fill (Add item on the Edit menu)/
Syntax	<b>ADDFILL</b> [x, y] . . .
Parameters	
x	The x coordinate of a point on the fill border.
y	The y coordinate of a point on the fill border.
Description	Use this command to enter a closed polygon and GerbTool fills the interior of the polygon. This command is commonly used to create ground plane areas. To use this command, enter at least 3 x, y coordinate pairs to specify the outline to fill. If you enter fewer, GerbTool stops and prompts you for the missing coordinates. GerbTool then outlines the polygon with the current D-code, and begins filling the interior of the polygon with increasing aperture sizes as long as it can continue to find an aperture that is twice the size of the current aperture being used. As it fills toward the center of the polygon the aperture sizes become larger and larger. This allows the polygon to be filled with the least amount of data thereby keeping the database size as small as possible. If an aperture twice the size of the last aperture used cannot be located GerbTool fills the remainder of the polygon with the last aperture found.
Example	The following example creates a 1-inch filled box, with a lower left corner at the 0,0 coordinate.  <pre> ACTIVE LAYER      1 CURRENT D CODE    10 ADDFILL           0,0, 0,1, 1,1, 1,0, 0,0 </pre>

**ADDFLASH**

Purpose	Allows you to add a flash at a specified location.
Menu command	Flash (Add item on the Edit menu)
Syntax	ADDFLASH [x, y] . . .
Parameters	
x	The x coordinate of the location to add a flash.
y	The y coordinate of the location to add a flash.
Description	Use this command to add a flash to your Gerber data. The flash is created on the active layer using the current D-code. If you want, you can use multiple coordinates with this command: GerbTool places a flash at each location.
Example	<p>The following example adds four flashes forming the corners of a 1 inch box, with a lower left corner at the 0,0 coordinate.</p> <pre> ACTIVELAYER      1 CURRENTDCODE     10 ADDFLASH         0,0, 0,1, 1,1, 1,0, 0,0 </pre>

**ADDPUR**

Purpose	Allows you to perform a polypour.
Menu command	Pour (Add item on the Edit menu)
Syntax	ADDPUR [x, y] . . .
Parameters	
x	The x coordinate of a point on the pour border.
y	The y coordinate of a point on the pour border.
Description	Use this command to enter a closed polygon and GerbTool fills the interior of the polygon using an intelligent polypour. This command is commonly used to create ground plane areas. To use this command, enter at least 3 x, y coordinate pairs to specify the outline you want filled. If you enter fewer, GerbTool stops and prompts you to enter more. This command uses the pour settings set up in the menu command or from the POUR command.
Example	<p>The following example creates a 1-inch filled box, with a lower left corner at the 0,0 coordinate.</p> <pre>ACTIVELAYER      1 CURRENTDCODE     10 ADDPUR           0,0, 0,1, 1,1, 1,0, 0,0</pre>
See also	POUR

**ADDTTEXT**

Purpose	Allows you to add a string of text to the database.
Menu command	Text (Add item on the Edit menu) <i>t</i>
Syntax	ADDTTEXT string [x, y]...
Parameters	
string	A string containing the text to be added.
x	The x coordinate of the location to add the text.
y	The y coordinate of the location to add the text.
Description	Use this command to add text to a Gerber layer. The text is drawn using the current D-code on the active layer using the settings that were set up in the menu command or from the TEXT command.
Example	The following example adds the text string "Hello World" starting at coordinate 0,0. <pre>ACTIVELAYER      1 CURRENTDCODE     10 ADDTTEXT         "Hello World" 0,0</pre>
See also	TEXT

**ADDVERTEX**

Purpose	Allows you to add a vertex to a line.
Menu command	Vertex (Add item on the Edit menu)
Syntax	ADDVERTEX [x, y] . . .
Parameters	
x	The x coordinate of the vertex to add.
y	The y coordinate of the vertex to add.
Description	Use this command to add a vertex to an existing trace, creating two traces from the original. The coordinate specified is the location of the new vertex and must be located on an existing trace.
Example	<p>The following example adds a vertex to the center of an existing line, whose end points are located in the variables \$x1, \$y1 and \$ux, \$uy.</p> <pre>CALC \$ptx = \$ux - \$lx CALC \$ptx = \$ptx / 2.0 CALC \$ptx = \$ptx + \$lx CALC \$pty = \$uy - \$ly CALC \$pty = \$pty / 2.0 CALC \$pty = \$pty + \$ly  ADDVERTEX \$ptx, \$pty</pre>



## Aperture functions

### APREPORT

Purpose	Produces an aperture usage report.
Menu command	Report (Apertures menu)
Syntax	<pre> APREPORT     REPFILExstring     LAYERexp     SORTBYstring     GO END </pre>
Parameters	
REPFILEx	The name of the report file the command produces.
LAYERexp	Specifies the layer on which to report. A value of '0' indicates that all layers are included in the report.
SORTBYstring	Must be either D-Code or List. You this to specify the ordering of the report for reports that target multiple layers.
Description	Use this command to produce an aperture usage report. The format of this report is the same as that produced by the Report command from the Aperturues menu. This report details the D-Codes (and their their definitions) that are being used on a per layer basis. Included in the report are use counts for both flashes and draws.
Example	<p>The following example produces an aperture report for all visible layers sorted by aperture list and places the result in the file OUTPUT.RPT.</p> <pre> APREPORT     REPFILEx "output.rpt"     LAYER 0 #output rep based on all lyrs     SORTBY "List"     GO END </pre>

**GETAPINFO**

Purpose	Use this command to obtain information about an aperture in an aperture list.
Menu command	Edit (Apertures menu)
Syntax	GETAPINFO [layer, dcode, shape, xsize, ysize, type, tool, toolsize, legend]
Parameters	
layer	An expression specifying the layer with the aperture list to be used.
dcode	An expression specifying the D-Code to get information on.
shape	A numeric variable that returns the shape of the aperture.
xsize	A numeric variable that returns the width of the aperture.
ysize	A numeric variable that returns the height of the aperture.
type	A numeric variable that returns the type of the aperture.
tool	A numeric variable that returns the tool number of the aperture.
toolsize	A numeric variable that returns the toolsize for the aperture.
legend	A numeric variable that returns the legend value for the aperture.
Description	Specify the number of a layer that uses the aperture list and the aperture D-Code. GerbTool places the appropriate information into the other variables passed into the command. You must include all of the variables listed with this command at least once within a macro, even if they are not going to be used by the macro. Subsequent calls can specify the first two parameters (layer, dcode) or can omit all parameters. Those variables used in the last complete call are remembered and used again.
Example	The example gets the aperture information used by layer 1 of a design, and writes its toolsize and D-code out to a file. <pre>IF \$dcode != 0     GETAPINFO 1, \$dcode, \$shp, \$xs, \$ys, \         \$type, \$tool, \$toolsize, \$legend     FILEWRITE \$fd, "%n %n", \$toolsize, \$dcode END</pre>
See also	PUTAPINFO

**PUTAPINFO**

Purpose	To update the information about a specific aperture in an aperture list.
Menu command	Edit (Apertures menu)
Syntax	PUTAPINFO [layer, dcode, shape, xsize, ysize, type, tool, toolsize, legend]
Parameters	See the parameter list for the GETAPINFO command. The PUTAPINFO and GETAPINFO parameter lists are identical and are shared between the two commands. For this reason, you need only specify the parameter list for one command, and the other command automatically uses the same variables.
Description	Use this command to update information about an aperture. You specify the number of a layer that uses the aperture list and the aperture D-code. GerbTool then places the appropriate information from the other variables passed into the command into the actual aperture list. All of the variables listed with this command must be included at least once within a macro, even if they are not going to be used by the macro. Subsequent calls can specify the first two parameters (layer, dcode) or can omit all parameters. Those variables used in the last complete call are remembered and used again.
Example	<p>The following example gets the aperture information for an aperture used by layer 1 of a design, changes the size and updates the aperture list.</p> <pre> IF \$dcode != 0     GETAPINFO \$lyr,\$dcode,\$shp,\$xs,\$ys, \         \$type,\$tool,\$toolsize,\$legend     CALC \$xs = \$xs * 1.2     CALC \$ys = \$ys * 1.2     PUTAPINFO END </pre>
See also	GETAPINFO

## Control statements

**CALLMACRO**

Purpose	Use this command to execute a macro within another macro.
Menu command	None
Syntax	CALLMACRO macroname, parameter0-9
Parameters	
macroname	A string indicating which macro to execute.
parameter0-9	From 0 to 9 variables, literals, or expressions.
Description	Allows a macro to “call” another macro. A system variable \$\$CALLDEPTH detects if a macro is being called: the parameters passed are visible to the called macro as parameters \$1-\$9. Any changes to these variables are reflected in the “caller” macro. The system variable \$\$STATUS is passed back to the calling macro. Note: If you pass in a literal value such as four, it is not updated when the called macro returns.
Example	<p>Within the called macro “maxes,” \$stop_layer and \$stop_dcode are automatically defined as \$1 and \$2. When maxes ends, \$stop_layer and \$stop_dcode are updated with values of \$1 and \$2.</p> <pre> MACRO test1     CALLMACRO "maxes", \$stop_layer,     \$stop_dcode     STRWRITE \$msg, "Max:lyr=%n\ndcode=%n", \     \$stop_layer, \$stop_dcode     MESSAGEBOX "MAX Info", \$msg, 0 END MACRO maxes     # \$1 is synonym for \$stop_layer     SET \$1 = \$\$MAXLAYERS + 1     REPEAT \$\$STATUS == \$\$FALSE         CALC \$1 = \$1 - 1         ACTIVE LAYER \$1     END     # \$2 is synonym for \$stop_dcode     CALLMACRO "maxuseddcode", \$2 END </pre>

**DEBUG**

Purpose	To turn macro debug mode on/off.
Menu command	None
Syntax	<code>DEBUG exp   strlit</code>
Parameters	
exp	An expression where 0=OFF and 1=ON, or a string literal of "Y" or "N".
Description	Use this function to toggle debug mode on or off. When debug mode is on, debug information is output to a file named <code>_MACRO_DEB</code> .
Example	The following example turns debug mode on and off. <pre>DEBUG 1      # on CALLMACRO   "NewMac" DEBUG 0      # off</pre>

**IF**

Purpose	To conditionally execute blocks of macro commands.
Menu command	None
Syntax	IF [numvar = ] exp
Parameters	
numvar	An optional numeric variable that receives the result of the expression on the right side of the equal sign.
exp	An expression indicating whether the macro statements between the IF and the corresponding ELSE or END statement will be executed.
Description	This function is used when you need to execute a block of macro statements only when a certain condition exists.
Example	<p>The following example shows two IF statements, one IF/ELSE/END trio and a nested IF/END pair.</p> <pre>GETSTRING "Enter your name", \$name STRLEN \$name IF \$\$STATUS == 0     #this block only executed if the     #variable \$\$STATUS is equal to 0     STOP "Bad Username" ELSE     #this block only executed if the     #variable \$\$STATUS is NOT equal to 0     GETSTRING "Enter your passwd", \$passwd     STRCMP \$passwd, "dilbert"     IF \$\$STATUS != 0         #this block only executed if the         #variable \$\$STATUS NOT equal to 0         STOP "Bad Passwd"     END END</pre>

**REPEAT**

Purpose	Repeats a series of macro commands until the test condition becomes false.
Menu command	None
Syntax	REPEAT [numvar = ] exp
Parameters	
numvar	An optional numeric variable that receives the result of the expression on the right hand side of the equal sign.
exp	An expression indicating whether the macro statements between the REPEAT and the corresponding END statement will be executed.
Description	This function executes a block of macro statements repeatedly while a certain condition exists. As long as the test condition is not zero, the commands between the REPEAT and END are executed repeatedly.
Example	<p>The following macro continues to prompt you for a text string as long as a blank string isn't entered.</p> <pre> SET \$cnt = 1 GETSTRING "Enter a component", \$comp STRLEN \$name REPEAT \$\$STATUS != 0     FILEWRITE \$fid, \         "COMP%.0n: %s", \         \$cnt, \$comp     CALC \$cnt = \$cnt + 1     GETSTRING "Enter a component", \$comp     STRLEN \$name END </pre>

## STOP

Purpose	Causes the macro to stop.
Menu command	None
Syntax	STOP [message]
Parameters	
message	An optional string variable or string literal giving a message that is presented to you when the macro is stopped.
Description	This command causes the macro to stop.
Example	The following example stops a macro if you enter “no.” <pre>GETYESNO "Quit? ", \$value IF \$value == \$\$YES     STOP END</pre>



## Database functions

### COPYITEM

Purpose	Use this command to copy an item in a Gerber database.
Menu command	Copy (Edit menu)
Syntax	COPYITEM layer, seqno, dx, dy
Parameters	
layer	An expression representing the layer containing the item to copy.
seqno	An expression representing the sequence number of the item to copy.
dx	An expression representing the offset to apply to the x coordinate of the object.
dy	An expression representing the offset to apply to the y coordinate of the object.
Description	Use this function to copy Gerber items in a layer. It takes the layer and sequence number of the object you want to copy. The sequence number used is the same as that returned by the menu command Item Info or the macro command GETFIRSTITEM. This function returns \$\$TRUE in the \$\$STATUS variable if the command was successful and \$\$FALSE if it was not.
Example	<p>The following example scans a layer and copies all flashes 1 inch to the left.</p> <pre> GETFIRSTITEM \$layer, \$seqno, \$net, \                 \$dcode,\$type, \$x, \$y, \                 \$x2, \$y2,\$dia,\$cw REPEAT \$\$STATUS != \$\$FALSE     IF \$type == \$\$FLASH         COPYITEM \$layer, \$seqno, -1.0, 0     GETNEXTITEM END END </pre>
See also	GETFIRSTITEM, GETNEXTITEM

**DELETEITEM**

Purpose	Use this command to delete an item from a Gerber database.
Menu command	Erase (Edit menu)
Syntax	<code>DELETEITEM layer, seqno</code>
Parameters	
layer	An expression representing the layer containing the item to delete.
seqno	An expression representing the sequence number of the item to delete.
Description	Use this function to delete Gerber items from a layer. It takes the layer and sequence number of the object you want to delete. The sequence number used is the same as that returned by the menu command Item Info or the macro command GETFIRSTITEM. This function returns \$\$TRUE in the \$\$STATUS variable if the deletion was successful and \$\$FALSE if it was not.
Example	The following example scans a layer and removes all flashes. <pre>GETFIRSTITEM \$layer,\$seqno,\$net,\$dcode, \                   \$type, \$x, \$y, \$x2, \$y2, \                   \$dia, \$cw REPEAT \$\$STATUS != \$\$FALSE     IF \$type == \$\$FLASH         DELETEITEM \$layer, \$seqno     END     GETNEXTITEM END</pre>
See also	GETFIRSTITEM, GETNEXTITEM

**GETEXTENTS**

Purpose	Use this command to calculate and return the extents of the Gerber data on any one or all loaded layers.
Menu command	Extents (Query menu)
Syntax	GETEXTENTS <i>layer</i> , <i>lx</i> , <i>ly</i> , <i>ux</i> , <i>uy</i>
Parameters	
<i>layer</i>	An expression indicating the layer or layers you want the extents of.
<i>lx</i>	A numeric variable that returns the x coordinate of the lower left extent of the data on the requested layer(s).
<i>ly</i>	A numeric variable that returns the y coordinate of the lower left extent of the data on the requested layer(s).
<i>ux</i>	A numeric variable that returns the x coordinate of the upper right extent of the data on the requested layer(s).
<i>uy</i>	A numeric variable that returns the y coordinate of the upper right extent of the data on the requested layer(s).
Description	This calculates the extents of the requested layer or layers. If <i>layer</i> contains a value greater than zero, the command determines the extent for that layer. If the value is zero, the extent for all visible layers is calculated, and if -1 is used, the extents of all layers regardless of their visibility is returned. The coordinates calculated by this command are returned in the other four variables.
Example	The following example gets the extents of all the layers in a design and draws a box around it.  <pre>GETEXTENTS    -1, \$lx, \$ly, \$ux, \$uy ADDDRAW      \$lx,\$ly,\$lx,\$uy,\$ux,\$uy, \              \$ux,\$ly, \$lx, \$ly</pre>

**GETFILMBOX**

Purpose	Use this command to return the size of the film box.
Menu command	Filmbox (Options menu)
Syntax	<code>GETFILMBOX xsize, ysize</code>
Parameters	
xsize	A numeric variable that returns the horizontal size of the film box.
ysize	A numeric variable that returns the vertical size of the film box.
Description	This function returns the size of the current film box. Since the lower left corner of the film box is always at coordinate 0,0, These values can be used to determine the proper positioning of any Gerber information.
Example	<p>The following example gets the size of the film box and draws a box on the active layer in the same location.</p> <pre>GETFILMBOX    \$xs, \$ys ADDDRAW      0, 0, 0, \$ys, xs, \$ys, \$xs, 0, 0, 0</pre>

**GETFIRSTITEM**

Purpose	Use this command to retrieve information about an object in a Gerber layer. This function also sets up the variables needed for the GETNEXTITEM function.
Menu command	Item Info (Query menu)
Syntax	GETFIRSTITEM layer, seqno, net, dcode, type, x, y, x2, y2, dia, cw, flags
Parameters	
layer	A variable containing the layer to scan for information. If the layer specified by this variable is not a valid layer, all layers are scanned, and the original contents of this variable is replaced with the layer of the item being returned.
seqno	A variable that returns the sequence number of the item being scanned. An item's sequence number is its relative location in the Gerber file. This is the same number as that displayed in the command Item Info.
net	A variable that returns the net id number associated with this item. If the item does not have a net associated with it, GerbTool returns -1.
dcode	A variable that returns the D-code of the scanned item.
type	A variable that returns the type of the item being scanned. Valid return values are \$\$FLASH and \$\$DRAW.
x	A variable that returns an x coordinate for the item being scanned. If the object in question is of type \$\$FLASH, this value represents the coordinate of the center of the flash. If the variable is of type \$\$DRAW, it represents the x coordinate of one of its end points.
y	A variable that returns an y coordinate for the item being scanned. If the object in question is of type \$\$FLASH, this value represents the coordinate of the center of the flash. If the variable is of type \$\$DRAW, it represents the y coordinate of one of its end points.
x2	A variable that returns the x coordinate of the second end point for lines. These are Gerber items that return a type of \$\$DRAW. If the type of an item is \$\$FLASH, this variable is not used.

y2	A variable that returns the y coordinate of the second end point for lines. These are Gerber items that return a type of \$\$DRAW. If the type of an item is \$\$FLASH, this variable is not used.
dia	A variable that returns the diameter of a Gerber arc. If the object being scanned is not an arc, the value 0.0 is returned.
cw	A variable that returns \$\$TRUE if the item being scanned is drawn in a clockwise direction, and \$\$FALSE if it is drawn in a counter-clockwise direction. If the item being scanned is not an arc, this variable is not used.
flags	A variable that returns the flags value for the current item. This parameter is currently used to indicate TOP (2048) or BOTTOM (4096) of the test point layer items. This parameter may have more uses in the future.
Description	Use this function to scan loaded Gerber files for information. When called, it examines the layer and seqno parameters for valid data. If they are valid, the specified items are loaded into the variables described above. If not, the first item in the first loaded layer is returned. To move to the next item, use the GETNEXTITEM command, which uses the variables initialized by this command. If this command is unable to find a Gerber item on this layer, a status of \$\$FALSE is returned, otherwise \$\$TRUE is returned.
Example	<p>The following example uses the GETFIRSTITEM and GETNEXTITEM commands to loop through all of the items on the layer represented by \$layer. When there are no more items left on the layer, \$\$STATUS will return \$\$FALSE and the repeat loop will stop.</p> <pre> GETFIRSTITEM \$layer,\$seqno,\$net,\$dcode, \ \$type,\$x,\$y,\$x2,\$y2,\$dia,\$cw REPEAT \$\$STATUS != \$\$FALSE ...process data here... GETNEXTITEM END </pre>
See also	GETNEXTITEM

**GETLAYER**

Purpose	Use this command to obtain information about a GerbTool layer (Gerber file).
Menu command	Edit (Layers menu)
Syntax	GETLAYER layer, fn, ln, an, vis, fc, dc, type, polarity, key, ft, lx, ly, ux, uy, netid
Parameters	
layer	A numeric variable containing the layer to obtain the information for. Note: A variable must be used.
fn	A string variable that returns the Gerber filename associated with this layer.
ln	A string variable that returns the layer name associated with this layer. This variable is only used on extended Gerber files.
an	A string variable that returns the name of the aperture list associated with this layer.
vis	A numeric variable that returns the visibility of the layer. The possible values are: 0=OFF, 1=ON, 2=REF.
fc	A string variable that returns the color of the flashes on this layer.
dc	A string variable that returns the color of the draws on this layer.
type	A numeric variable that returns the layer type of this layer. The possible values are: 0=TOP, 1=INNER, 2=BOTTOM, 3=PLANE, 4=COMPOSITE, 5=OTHER.
polarity	A numeric variable returning the polarity of this layer. The possible values are 0=clear, 1=dark. This variable is only used on extended Gerber files.
key	A numeric variable that returns the key value associated with this layer. This variable is only used on extended Gerber files.
ft	A numeric variable that returns the type of file loaded on this layer. This value is currently not used.

<code>lx</code>	A numeric variable that returns the x coordinate of the lower left most extent of the data on this layer.
<code>ly</code>	A numeric variable that returns the y coordinate of the lower left most extent of the data on this layer.
<code>ux</code>	A numeric variable that returns the x coordinate of the upper right most extent of the data on this layer.
<code>uy</code>	A numeric variable that returns the y coordinate of the upper right most extent of the data on this layer.
<code>netid</code>	A numeric variable that returns the netid value that is associated with this layer.
Description	Use this command to retrieve information about a given Gerber layer. This information is returned in the variables described above. Note that all of the variables must be included with this command even if they are not used.
Example	The following example retrieves information regarding the layer whose number is stored in <code>\$layer</code> . <pre>GETLAYER \$layer,\$fn,\$ln,\$an,\$vis, \ \$fc,\$dc,\$type,\$polarity, \ \$key,\$ft,\$lx,\$ly,\$ux, \$uy,\$netid</pre>



**GETNEXTITEM**

Purpose	Use this command to retrieve information about an object in a Gerber layer. This function is used in conjunction with the GETFIRSTITEM function.
Menu command	Item Info (Query menu)
Syntax	GETNEXTITEM
Parameters	None
Description	Use this function along with GETFIRSTITEM to scan loaded Gerber files for information. GETFIRSTITEM must be called first in order to setup the variables and retrieve the first item. When GETNEXTITEM is called, the information for the next Gerber object on the layer is placed into the same variables that were created and used by the GETFIRSTITEM command, and the \$\$STATUS variable is set to \$\$TRUE. When there are no more objects to process, a status of \$\$FALSE will be returned.
Example	<p>The following example uses the GETFIRSTITEM and GETNEXTITEM commands to loop through all of the items on the layer represented by \$layer. When there are no more items left on the layer, \$\$STATUS will return \$\$FALSE and the repeat loop will stop.</p> <pre> GETFIRSTITEM \$layer,\$seqno,\$net,\$dcode, \     \$type, \$x, \$y, \$x2, \$y2, \$dia, \$cw REPEAT \$\$STATUS != \$\$FALSE     ...process data here...     GETNEXTITEM END </pre>
See also	GETFIRSTITEM

**GETUSERDATA**

Purpose	Use this command to obtain the UserData field for a specific database item.
Menu command	Item Info (Query menu) or Item (Edit menu)
Syntax	GETUSERDATA [layer, seqno userdata]
Parameters	
layer	A numeric variable indicating the layer.
seqno	A numeric variable indicating the sequence number of the item.
userdata	A string variable that will receive the UserData from the specified database item.
Description	Use this function to locate the specified item and copy its UserData field to the specified string variable. Specify the parameters for this function on the first call to GETUSERDATA. You can omit them thereafter. The variables used for the layer and seqno parameters are usually those used in a GETFIRSTITEM/GETNEXTITEM loop.
Example	<p>The following example scans the active layer allowing you to edit the UserData field of each item.</p> <pre> MACRO getuserdata     SET \$lyr = \$\$ACTIVE_LAYER     SET \$seqno = 0     STRSET \$user, ""      GETUSERDATA \$lyr, \$seqno, \$user     GETFIRSTITEM \$lyr, \$seqno, \$net, \$dcode, \         \$type, \$x, \$y, \$x2, \$y2, \$d     REPEAT \$\$STATUS         GETUSERDATA         GETSTRING "Edit UserData:", \$user         GETNEXTITEM     END ENDMACRO </pre>
See also	PUTUSERDATA, GETFIRSTITEM, GETNEXTITEM

**GETVIEWEXTENTS**

Purpose	Use this command to obtain the extents of the current viewing window.
Menu command	None
Syntax	GETVIEWEXTENTS lowerx, lowery, upperx, uppery
Parameters	
lowerx	A numeric variable that receives the lower left X limit.
lowery	A numeric variable that receives the lower left Y limit.
upperx	A numeric variable that receives the upper right X limit.
uppery	A numeric variable that receives the upper right Y limit.
Description	This function returns the extents of the current viewing window in the four numeric variables specified.
Example	<p>The following example gets the size extents of the current view window and then zooms in at its center.</p> <pre> GETVIEWEXTENTS \$lx, \$ly, \$ux, \$uy CALC \$tx = \$ux - \$lx CALC \$ty = \$uy - \$ly CALC \$tx = \$tx / 2 CALC \$ty = \$ty / 2 CALC \$tx = \$lx + \$tx CALC \$ty = \$ly + \$ty ZOOMIN \$tx, \$ty </pre>

**MOVEITEM**

Purpose	Use this command to move a selected item in a Gerber database.
Menu command	Move (Edit menu)
Syntax	<code>MOVEITEM layer, seqno, dx, dy</code>
Parameters	
layer	An expression representing the layer containing the item to move.
seqno	An expression representing the sequence number of the item to move.
dx	An expression representing the amount of offset to apply to the x coordinate of the object.
dy	An expression representing the amount of offset to apply to the y coordinate of the object.
Description	Use this function to move Gerber items in a layer. It takes the layer and sequence number of the object you want to move. The sequence number used it the same as that returned by the Item Info command on the Query menu or the macro command <code>GETFIRSTITEM</code> . This function returns <code>\$\$TRUE</code> in the <code>\$\$STATUS</code> variable if the move was successful and <code>\$\$FALSE</code> if it was not.
Example	<p>The following example scans a layer and moves all flashes 1 inch to the right.</p> <pre> GETFIRSTITEM \$layer,\$seqno,\$net,\$dcode, \ \$type, \$x, \$y, \$x2, \$y2, \$dia, \$cw REPEAT \$\$STATUS != \$\$FALSE     IF \$type == \$\$FLASH         MOVEITEM \$layer, \$seqno, 1.0, 0     END     GETNEXTITEM END </pre>
See also	<code>GETFIRSTITEM</code>

**PUTUSERDATA**

Purpose	Use this command to update the UserData field for a specific database item.
Menu command	Item (Edit menu)
Syntax	PUTUSERDATA [layer, seqno, userdata]
Parameters	
layer	A numeric variable indicating the layer.
seqno	A numeric variable indicating the sequence number of the item.
userdata	A string variable that will be used to update the UserData in the specified database item.
Description	This function locates the specified item and copies the string from the userdata parameter into its UserData field. The above parameters should be specified on the first call to GETUSERDATA but can thereafter be omitted. The variables used for the layer and seqno parameters are usually those used in a GETFIRSTITEM/GETNEXTITEM loop.
Example	<p>The following example scans the active layer initializing the UserData field of each item.</p> <pre> MACRO putuserdata     SET \$lyr = \$\$ACTIVE_LAYER     SET \$seqno = 0     SET \$cnt = 0     STRSET \$user, ""      GETUSERDATA \$lyr, \$seqno, \$user     GETFIRSTITEM \$lyr, \$seqno, \$net, \$dcode, \         \$type, \$x, \$y, \$x2, \$y2, \$d      REPEAT \$\$STATUS          CALC \$cnt = \$cnt + 1         STRWRITE \$user, "U%.0n", \$cnt         PUTUSERDATA         GETNEXTITEM      END  ENDMACRO </pre>
See also	GETUSERDATA, GETFIRSTITEM, GETNEXTITEM

## Editing functions

### ALIGNLAYERS

Purpose	Use this command to align Gerber layers based on common items.
Menu command	Align (Edit menu)
Syntax	<code>ALIGNLAYERS [x, y...]</code>
Parameters	
x, y	A variable number of parameters specifying the coordinates of common objects to align.
Description	Use this function to align Gerber layers that no longer share a common origin. The first coordinate represents the location on a reference layer of an item to which you want to align all of the other layers. The remainder of the coordinates represent the locations of objects on other layers that you want to align to the first specified object.
Example	The following example performs a layer alignment of two layers. The location of the reference object is at 0,0. The location of the object on the second layer has been calculated, and is located in the variables \$x, \$y. <code>ALIGNLAYERS 0, 0, \$x, \$y</code>

**CLIP**

**Purpose** This command allows a macro to erase items, allowing clipping of lines.

**Menu command** Clip (Edit menu)

**Syntax** CLIP

```

BY          exp
BOUNDARY   yesno
FLASHES    yesno
DRAWS      yesno
ARCS       yesno
DCODE      exp
LAYER      exp
GO         [x1, y1, x2, y2 ...]

```

END

**Parameters**

**BY** An expression indicating how to perform the clipping. Valid values are: 1=window, 2=group.

**BOUNDARY** Controls erasure of flashes straddling the window boundary.

**DCODE** Indicates a D-Code filter to use for the command. If zero is used, all D-Codes may be affected by the command.

**LAYER** Indicates a layer filter to use for the command. If zero is used, all visible layers may be affected by the command.

**GO** A variable number of coordinates used by the system.

**Description** Use this function to erase a selection of items with automatic clipping of lines that cross the window boundary.

Example

The following example clips all arcs using window mode.

CLIP

```
By          $$WINDOWMODE
Boundary $$NO # no flashes
Flashes    $$NO
Draws      $$NO
Arcs       $$YES
Layer 0    #Erase from all visible lyrs
Dcode 0    #Erase all D-Codes
GetWindow "Enter Clip Window", \
```

```
          $lx,$ly,$ux,$uy
```

```
GO $lx,$ly,$ux,$uy
```

END

See also

ERASE



**COPY**

**Purpose** This command allows a macro to perform a copy.

**Menu command** Copy (Copy menu)

**Syntax**

```

COPY
    BY          exp
    BOUNDARY   yesno

    FLASHES   yesno
    DRAWS     yesno

    ARCS      yesno
    DCODE     exp

    LAYER     exp
    TOLAYER   exp
    GO        [x1, y1, x2, y2 ...]
END

```

**Parameters**

**BY** An expression indicating how to perform the copy. Valid values are: 0=item, 1=window, 2=group.

**DCODE** An expression indicating a D-Code filter to use for the copy. If zero is used, all D-Codes may be affected by the command.

**LAYER** An expression indicating a layer filter to use for the copy. If zero is used, all visible layers may be affected by the command.

**TOLAYER** An expression indicating the layer to copy all of the selected objects to. If zero is used, the objects will be kept on their original layers.

[x1, y1, x2, y2 ...] A variable number of coordinates used by the system to complete the command.

**Description** Use this function to perform a copy of a number of items. If window mode is selected the first 2 x, y coordinate pairs given are used to specify the window, if item mode is used, the first pair is used to select the item to copy, and in group mode the select group is used and none of the coordinate pairs are used to select the items. The next coordinate is used to specify the starting location for the copy, and all subsequent coordinate pairs are used to specify to locations where the copied data is to be placed.

## Example

The following example obtains a window, obtains from and to locations from the user, and performs a copy.

COPY

```
By          $$WINDOWMODE
Boundary    $$YES
Flashes     $$YES
Draws       $$YES
Arcs        $$YES
Layer       0 # Copy from all visible
Dcode       0 # Copy all D-Codes
GetWindow   "Enter Copy Window", \
            $lx,$ly,$ux,$uy
GetPoint    "Enter from location", \
            $fx,$fy
GetPoint    "Enter to location", \
            $tx,$ty
GO          $lx,$ly,$ux,$uy, \
            $fx,$fy, $tx, $ty
```

END

**DCEXPAND**

**Purpose** Use this command to expand a custom aperture into normal Gerber entities.

**Menu command** Expand (Dcode item on the Edit menu)

**Syntax**

```
DCEXPAND
        DCODE      exp
        LAYER      exp
        GO
END
```

**Parameters**

**DCODE** An expression indicating a D-Code filter to use when transcoding. If zero is used, all D-Codes may be affected by the command.

**LAYER** An expression indicating a layer filter to use for transcoding. If zero is used, all visible layers may be affected by the command.

**Description** This function is used to expand custom apertures into their basic Gerber constructs.

**Example** The following example expands all of the custom apertures on all visible layers.

```
DCEXPAND
        LAYER 0 # Change all visible layers
        DCODE 0 # Change all D-Codes
END
```

**DCODESCALE**

Purpose	This command allows a macro to scale D-Codes in a design.
Menu command	Scale (Dcode item on the Edit menu)
Syntax	<pre>DCODESCALE     FIXED      yesno     SCALE      exp, exp     BY          exp     BOUNDARY   yesno     FLASHES    yesno     DRAWS      yesno     ARCS       yesno     DCODE      exp     LAYER      exp     GO         [x1, y1, x2, y2 ...] END</pre>
Parameters	
SCALE	A pair of expressions indicating the X-Y scaling factors.
BY	An expression indicating how to select the items to change. Valid values are: 0=item, 1=window, 2=group.
DCODE	An expression indicating a D-Code filter to use for the command. If zero is used, all D-Codes may be affected by the command.
LAYER	An expression indicating a layer filter to use for the command. If zero is used, all visible layers may be affected by the command. [x1, y1, x2, y2 . . .]. A variable number of coordinates used by the system to complete the command.
Description	Use this function to scale the D-codes of a number of selected items. If window mode is selected the first 2 x, y coordinate pairs given are used to specify the window, if item mode is used, the first pair is used to select the item to copy, and in group mode the select group is used and no coordinates need to be specified. If fixed is chosen, the values specified for the scale are added to the sizes of the apertures in question. If fixed is given the value of \$\$\$NO, the sizes of the apertures are multiplied by the scale values.

## Example

The following example obtains a window's from and to locations from the user, and scales all of the flashes in this area up by 5 percent.

```
DCODESCALE
    FIXED    $$NO
    SCALE    1.05, 1.05
    BY       $$WINDOWMODE
    BOUNDARY $$YES
    FLASHES  $$YES
    DRAWS    $$NO
    ARCS     $$NO
    LAYER 0  #Copy from all visible layers
    DCODE 0  #Copy all D-Codes
    GETWINDOW"Enter Window", \
            $lx,$ly,$ux,$uy
    GO $lx,$ly,$ux,$uy
END
```

**ERASE**

Purpose	This command allows a macro to erase items.
Menu command	Erase (Edit menu)
Syntax	<pre> ERASE     BY          exp     BOUNDARY  yesno     FLASHES   yesno     DRAWS     yesno     ARCS      yesno     DCODE     exp     LAYER     exp     GO        [x1, y1, x2, y2 ...] END </pre>
Parameters	
BY	An expression indicating how to perform the erasure. Valid values are: 0=item, 1=window, 2=group.
DCODE	An expression indicating a D-Code filter to use for the command. If zero is used, all D-Codes may be affected by the command.
LAYER	An expression indicating a layer filter to use for the command. If zero is used, all visible layers may be affected by the command.
GO	A variable number of coordinates used by the system to complete the command.
Description	Use this function to erase a selection of items. If window mode is selected the first 2 x, y coordinate pairs given are used to specify the window, if item mode is used, the first pair is used to select the item to delete, and in group mode the select group is used and no coordinates need to be specified.

## Example

The following example obtains a window and erases all arcs and draws in it.

## ERASE

```
By          $$WINDOWMODE
Boundary    $$YES
Flashes     $$NO
Draws       $$YES
Arcs        $$YES
Layer 0     #Erase from all visible lyrs
Dcode 0     #Erase all D-Codes
GetWindow "Enter Erase Window", \
           $lx,$ly,$ux,$uy
GO $lx,$ly,$ux,$uy
```

END

## See also

CLIP

**MIRROR**

Purpose This command allows a macro to mirror objects.

Menu command Mirror (Edit menu)

Syntax

**MIRROR**

```

DIRECTION      <exp | "H" | "V">
CENTERED       yesno
BY             byexp
BOUNDARY       yesno
FLASHES        yesno
DRAWS          yesno
ARCS           yesno
DCODE          dcrexp
LAYER          lyrexp
GO             [x1, y1, x2, y2 ...]

```

**END**

Parameters

**DIRECTION** An expression indicating the direction of the mirroring. Valid values are: 0=horizontal and 1=vertical. Also acceptable are the text characters H and V.

**BY** An expression indicating how to perform the mirror. Valid values are 0=item, 1=window, 2=group.

**DCODE** An expression indicating a D-Code filter to use mirroring. If zero is used, all D-Codes may be affected by the command.

**LAYER** An expression indicating a layer filter to use for mirroring. If zero is used, all visible layers may be affected by the command.

**GO** A variable number of coordinates used by the system to complete the command.

Description

Use this function to mirror a selection of items. If window mode is selected the first 2 x, y coordinate pairs given are used to specify the window, if item mode is used, the first pair is used to select the item to copy, and in group mode the select group is used and none of the coordinate pairs are used to select the items. If centered is set to `$$NO`, the next coordinate specifies the center of rotation, otherwise GerbTool uses the center of the selected items.



## Example

The following example obtains a window from the user and mirrors all of the items in it.

## MIRROR

```
DIRECTION 0 # horz
CENTERED  $$YES
By        $$WINDOWMODE
Boundary  $$YES
Flashes   $$YES
Draws     $$YES
Arcs      $$YES
Layer 0   # Mirror all visible layers
Dcode 0   # Mirror all D-Codes
GetWindow "Enter Mirror Window", \
          $lx,$ly,$ux,$uy
GO        $lx,$ly,$ux,$uy
```

END

**MOVE**

Purpose	This command allows a macro to perform a move.
Menu command	Move (Edit menu)
Syntax	<pre> MOVE     BY          exp     BOUNDARY   yesno     FLASHES    yesno     DRAWS      yesno     ARCS       yesno     DCODE      exp     LAYER      exp     TOLAYER    exp     GO         [x1, y1, x2, y2 ...] END </pre>
Parameters	
BY	An expression indicating how to perform the move. Valid values are: 0=item, 1=window, 2=group.
DCODE	An expression indicating a D-Code filter to use for the move. If zero is used, all D-Codes may be affected by the command.
LAYER	An expression indicating a layer filter to use for the move. If zero is used, all visible layers may be affected by the command.
TOLAYER	An expression indicating the layer to move all of the selected objects to. If zero is used, the objects will be kept on their original layers.
GO	A variable number of coordinates used by the system to complete the command.
Description	Use this function to perform a move on a selection of items. If you select window mode, the first 2 x, y coordinate pairs given specify the window. If you select item mode, the first pair indicates the item to copy. In group mode, the select group is used and none of the coordinate pairs are used to select the items. The next coordinate is used to specify the starting location for the move, and all subsequent coordinate pairs are used to specify to locations where the moved data is to be placed.

Example      The following example obtains a window, from and to locations from the user and performs a move.

MOVE

```
By            $$WINDOWMODE
Boundary     $$YES
Flashes     $$YES
Draws       $$YES
Arcs        $$YES
Layer 0     #Move from all visible layers
Dcode 0     #Move all D-Codes
GetWindow "Enter Move Window", \
            $lx,$ly,$ux,$uy
GetPoint "Enter from location", $fx,$fy
GetPoint "Enter to location", $tx,$ty
GO          $lx,$ly,$ux,$uy, $fx,$fy,$tx,$ty
```

END

**ORIGIN**

Purpose	Use this command to enable a macro to change the origin used by GerbTool.
Menu command	Origin (Edit menu)
Syntax	ORIGIN [x, y...]
Parameters	
x, y	A variable number of coordinates representing new origins.
Description	Use this function to change the origin used by GerbTool. The origin is the 0,0 coordinate of the Gerber files and is always located at the lower left corner of the film box.
Example	The following example calculates the center of the extents of all visible Gerber files and moves the origin to that point.

```
GETEXTENTS -1, $lx, $ly, $ux, $uy
CALC $x = $ux - $lx
CALC $x = $x / 2
CALC $x = $x + $lx
CALC $y = $uy - $ly
CALC $y = $y / 2
CALC $y = $y + $ly
ORIGIN $x, $y
```

**POUR**

Purpose	This command allows a macro to perform a polypour.
Menu command	Pour (Add item on the Edit menu)
Syntax	<pre>POUR         DRAWCLR    exp         FLASHCLR   exp         MINAREA    exp         TYPE       exp         LAYER      exp         HATCHLINE  line, dcode, step, angle         DCODE     exp         GO         [x1, y1, x2, y2 ...] END</pre>
Parameters	
DRAWCLR	An expression indicating the clearance that the pour must maintain from drawn objects.
FLASHCLR	An expression indicating the clearance the pour is to maintain from flashes.
MINAREA	An expression that indicates the minimum area to be filled by a pour.
TYPE	An expression giving the type of pour to perform. Valid values are: 1=outline, 2=solid, 3=hatch.
LAYER	An expression indicating the layer to place the resulting polypour onto.
DCODE	An expression that represents D-Code to be used for this particular hatch line.
HATCHLINE	A quad of expressions indicating describing one of the three possible hatch lines. The first expression is a line number of 1-3. The remaining parameters describe the dcode, step size and angle of the selected hatch line. This parameter can be used multiple times in a single pour command for more complicated pour patterns.
GO	A variable number of coordinates the outline of the area to perform the polypour. The starting coordinate should also be specified at the end of the coordinate list in order to close the polygon and complete the pour.
Description	Use this function to perform a polypour.

**Example**

The following example draws a doubly-nested rectangle in such a way that it can be filled with the cross hatch pattern set up for this pour.

POUR

```
DRAWCLR 0.02
FLASHCLR 0.02
TYPE 2 # 0 == OUTLINE,
      # 1 == SOLID,
      # 2 == HATCHED
HATCHLINE 1, $$CURRENTDCODE, 0.35, 45
HATCHLINE 2, $$CURRENTDCODE, 0.35, 135
HATCHLINE 3,0,0.0,0 #only use 2 lines
GO $olx,$oly, $olx,$ouy, $oux,$ouy, \
   $oux,$oly, $iux,$ily, $iux,$iuy, \
   $ilx,$iuy, $ilx,$ily, $iux,$ily, \
   $oux,$oly, $olx,$oly
```

END

**PURGE**

Purpose	Use this command to compact the Gerber databases and purge the undo queue.
Menu command	Purge (Edit menu)
Syntax	PURGE
Parameters	None
Description	This function compacts the Gerber files that are loaded into GerbTool. GerbTool doesn't actually erase data from memory during edits, and therefore memory may become fragmented and less efficient. The purge command deleted these items from the Gerber database and removes the contents of the undo queue.
Example	The following is an example of using the undo command. <pre>GETYESNO "Purge?", \$value IF \$value == \$\$YES     PURGE END</pre>

**ROTATE**

Purpose	Use this function to rotate items.
Menu command	Rotate (Edit menu)
Syntax	<pre> ROTATE     DEGREES  exp     CENTERED yesno     BY       exp     BOUNDARY yesno     FLASHES  yesno     DRAWS    yesno     ARCS     yesno     DCODE    exp     LAYER    exp     GO       [x1, y1, x2, y2 ...] END </pre>
Parameters	
DEGREES	An expression indicating the number of degrees to rotate the selected objects in a counter clockwise direction.
BY	An expression indicating how to perform the rotation. Valid values are: 0=item, 2=group.
DCODE	An expression indicating a D-Code filter to use for the command. If zero is used, all D-Codes may be affected by the command.
LAYER	An expression indicating a layer filter to use for the command. If zero is used, all visible layers may be affected by the command.
GO	A variable number of coordinates used by the system to complete the command.
Description	Use this function to rotate a selection of items. If window mode is selected the first 2 x, y coordinate pairs given are used to specify the window, and in group mode the select group is used. If <b>CENTERED</b> is set to <b>\$\$NO</b> , the next coordinate is used to specify a pivot point.



## Example

The following example obtains a window and rotates all arcs and draws but not the flashes.

```
ROTATE
  DEGREES  90.0
  CENTERED $$YES
  BY       $$WINDOWMODE
  BOUNDARY $$YES
  FLASHES  $$NO
  DRAWS    $$YES
  ARCS     $$YES
  LAYER    0
  DCODE    0
  GETWINDOW"Enter Window to rotate", \
          $lx,$ly,$ux,$uy
  GO $lx,$ly,$ux,$uy
END
```

**SELECTCRITERIA**

Purpose	This command allows a macro to modify the selection criteria that is used by many of the editing commands.
Menu command	All editing commands.
Syntax	<pre> SELECTCRITERIA     BY          exp     BOUNDARY   yesno     FLASHES    yesno     DRAWS      yesno     ARCS       yesno     DCODE      exp     LAYER      exp  END </pre>
Parameters	
BY	An expression that describes which items are to be selected. Valid values are: 0=item, 1=window, 2=group, 3=layer, 4=net.
DCODE	An expression that describes the D-Code you are interested in operating on. Use a value of zero to enable selection of all D-Codes.
LAYER	An expression that describes the layer you select objects from. Use a value of zero to select from all visible layers.
Description	Use this function to set the selection criteria that is shared by most GerbTool editing commands. Note that since this is a block command, only the variables that you want to change need to be given when using the command.
Example	<p>The following example sets the selection criteria to window mode, includes items that cross the window boundary, includes flashes, excludes draws and arcs, sets the layer to the value of the \$layer variable, and sets the D-code to zero.</p> <pre> SELECTCRITERIA     BY          \$\$WINDOWMODE     BOUNDARY   \$\$YES     FLASHES    \$\$TRUE     DRAWS      \$\$FALSE     ARCS       \$\$FALSE     LAYER      \$layer     DCODE      0  END </pre>

**SELECTGROUP**

Purpose	This command allows a macro to manipulate the select groups used in GerbTool.
Menu command	Select (Edit menu)
Syntax	<pre> SELECTGROUP         BY          exp         BOUNDARY  yesno         FLASHES   yesno         DRAWS     yesno         ARCS      yesno         DCODE     exp         LAYER     exp         MODE      exp         GO        [lx, ly, ux, uy...] END </pre>
Parameters	
BY	An expression that describes how to perform the selection. Valid values are: 0=item, 1=window, 4=net.
DCODE	An expression that describes the D-Code you are interested in operating on. Use a value of zero to enable selection of all D-Codes.
LAYER	An expression that describes the layer you select objects from. Use a value of zero to select from all visible layers.
MODE	An expression that describes which action to perform on the select set. Valid values are: 0=reset, 1=add, 2=remove, 3=invert.
GO	A variable number of variables that are used to pass coordinates to this command. If you are selecting by item or net, each pair of x, y coordinates will be used to select an item. If you are selecting by window, each two pair of coordinates will be used to determine the window the command will use. If you are resetting or inverting the select group, no coordinates need to be passed into this command.
Description	This function is the macro interface into the Select Group command of GerbTool. Note that since this is a block command, only the variables that you want to change need to be given when using the command. The system variable \$\$SELGRPCNT is used to return the current number of items in the select group.

## Example

The following example performs a selection by window of all the flashes on a given layer. Note how this example also shows both, how only the variables to be changed need to be mentioned, but also how they can be called multiple times in the same block.

```
SELECTGROUP
  By      $$WINDOWMODE
  Flashes $$TRUE
  Draws   $$FALSE
  Arcs     $$FALSE
  Layer   $layer
END

REPEAT      $$TRUE
  GetWindow "Enter Component Window", \
            $lx,$ly, $ux,$uy
  SELECTGROUP
    MODE 0      #reset the select set
    GO
    MODE 1      #select by window
    GO $lx,$ly,$ux,$uy
  END
END
```

**TEXT**

Purpose This command allows a macro insert text.

Menu command Text (Add item on the Edit menu)

Syntax TEXT

```

HEIGHT      exp
WIDTH      exp
ROTATE     yesno
SLANT      exp
MIRROR     yesno
FILE       string
LINESPACE  exp
CHARSPACE  exp
GO         [x, y, ...]

```

END

## Parameters

**HEIGHT** An expression that describes the height of the characters in the text to be added.

**WIDTH** An expression that describes the width of the characters in the text to be added.

**SLANT** An expression that describes the amount of slant, in degrees, you want to apply to each individual character.

**FILE** A string that gives the name of the file that contains the text to be added.

**LINESPACE** An expression that gives the spacing between lines of text, where 1.0 represents single-spaced text, 2.0 represents double-spaced, and so on.

**CHARSPACE** An expression that represents the spacing between characters in the added text, where 1.0 gives normal character spacing, 2.0 gives double spaces between characters, and so on.

**GO** A variable number of coordinates specifying where to add the text. If more than one set of coordinates are given, the same text will be added to each location.

Description This function adds text to a Gerber file. The text to be added is in read in from a file whose name is specified in the FILE variable.

Example

The following example places a logo one-half inch in from the lower left corner of the board.

```
GETTEXTENTS -1, $lx, $ly, $ux, $uy
CALC $lx = $lx + 0.5
CALC $ly= $ly+ 0.5
TEXT
    HEIGHT 0.25
    WIDTH 0.25
    FILE "logo.txt"
    GO $lx, $ly
END
```

**TRANSCODE**

**Purpose** This command allows a macro to change the D-Codes of selected items.

**Menu command** Transcode (Dcode item on the Edit menu)

**Syntax**

```
TRANSCODE
    NEWDCODE exp
    BY      exp
    BOUNDARY yesno
    FLASHES yesno
    DRAWS   yesno
    ARCS    yesno
    DCODE   exp
    LAYER   exp
    GO      [x1, y1, x2, y2 ...]
```

END

**Parameters**

**NEWDCODE** An expression indicating the new D-Code that you want all selected items changed to.

**BY** An expression indicating how to perform the transcode. Valid values are: 0=item, 1=window, 2=group.

**DCODE** An expression indicating a D-Code filter to use when transcoding. If zero is used, all D-Codes may be affected by the command.

**LAYER** An expression indicating a layer filter to use for transcoding. If zero is used, all visible layers may be affected by the command.

**GO** A variable number of coordinates used by the system to complete the command.

**Description** This function changes the apertures or transcodes a selection of items. If you select window mode, the first two x, y coordinate pairs given specify the window. If you select item mode, the first pair selects the item to copy. In group mode, the select group is used and none of the coordinate pairs are used to select the items.

Example

The following example obtains a window and changes all of the D10 flashes in the window to D15.

```
TRANSCODE
  NEWDCODE 15
  By      $$WINDOWMODE
  BOUNDARY $$YES
  FLASHES $$YES
  DRAWS   $$NO
  ARCS    $$NO
  LAYER 0 # Change all visible layers
  DCODE 0 # Change all D-Codes

  GETWINDOW"Enter Window to change", \
        $lx,$ly,$ux,$uy
  GO $lx,$ly,$ux,$uy
END
```



## Environment functions

### ACTIVELAYER

Purpose	This command allows you to set the active layer.
Menu command	Active (Layers menu)
Syntax	ACTIVELAYER layer
Parameters	
layer	An expression indicating the layer to make active.
Description	Use this function to change the active layer inside a macro. The active layer is the layer where any newly created objects are placed. If the macro attempts to set the active layer to a layer that doesn't exist, the function returns \$\$FALSE, otherwise it returns \$\$TRUE.
Example	The following example loops through a number of layers, and adds a flash to each layer by changing the active layer to the value of the loop.

```

SET $layer = 0
REPEAT $layer <= $$MAXLAYERS
    ACTIVELAYER $layer
    IF $$STATUS
        ADDFLASH $lx,$ly
    END
    CALC $layer = $layer + 1
END

```

## BKCOLOR

Purpose	This command allows you to set the background color on the screen.
Menu command	Bg Color (Options menu)
Syntax	BKCOLOR color
Parameters	
color	A text string representing the color to set the background to.
Description	Use this function to change the color of the background screen. The color string can be any of the colors given in the color file COLOR.RGB, even if these colors are not available in GerbTool's color chooser.
Example	The following example sets the background color in GerbTool to a rather unusable color.  MACRO TESTCOLOR BKCOLOR "PapayaWhip" ENDMACRO

**CURRENTDCODE**

Purpose	This command allows you to set the current D-code.
Menu command	Change (Apertures menu)
Syntax	CURRENTDCODE dcode
Parameters	
dcode	An expression indicating the D-code to make current.
Description	Use this function to change the current D-code inside a macro. The current D-code is the aperture with which any newly created objects are created. If the macro attempts to set the current D-code to an aperture that doesn't exist, the function returns \$\$FALSE, otherwise it returns \$\$TRUE.
Example	The following example loops through a number of layers, and adds a different flash to each layer by changing the active layer to the value of the loop, and incrementing the current D-code.

```

REPEAT      $layer <= $$MAXLAYERS
    ACTIVELAYER    $layer
    CURRENTDCODE$dcode
    IF      $$STATUS
        ADDFLASH    $lx,$ly
    END
    CALC      $layer = $layer + 1
    CALC      $dcode = $dcode + 1
END

```

## EXTENSIONS

Purpose	Use this command to change the default file extensions associated with the different file types supported by GerbTool.
Menu command	Defaults (Options menu)
Syntax	<pre>EXTENSIONS     GERBER      string     APLISTS     string     DESIGNS     string     DRILL       string     HPGL        string     POSTSCRIPT  string     TOOL        string     LASERJET    string END</pre>
Description	Use this command to control the default file extensions that GerbTool will use when searching for files of a particular type.
Example	<p>The following example modifies the default extensions.</p> <pre>EXTENSIONS     GERBER      "lgr"     APLISTS     "apr"     DESIGNS     "job"     HPGL        "plt" END</pre>

**FILESPATH**

Purpose	Use this command to change the location where GerbTool looks for files.
Menu command	Defaults (Options menu)
Syntax	<code>FILESPATH string</code>
Parameters	
string	A text string defining the new path.
Description	This function accepts a character string and uses it to change the path GerbTool uses to find the Gerber files it is working on. This can be used to change the location where files are to be saved or loaded from.
Example	The following example changes the current directory to a temporary one so the files can be saved there and not overwrite the original ones. <code>FILESPATH "C:\temp\gerbers"</code>

**FILMBOX**

Purpose	Use this command to adjust the size and color of the film box.
Menu command	Film Box (Options menu)
Syntax	<code>FILMBOX xsize, ysize, color</code>
Parameters	
xsize	An expression giving the width of the film box.
ysize	An expression giving the height of the film box.
color	A string giving the color that the film box is to be drawn with.
Description	Use this command to change the size and color of the film box that GerbTool displays. Since the lower left corner of the film box is always at the 0,0 coordinate, only the width and height need to be given. The color string can be any of the colors given in the color file COLOR.RGB, even if these colors are not available in GerbTool's color chooser.
Example	The following example sets the film box to 8.5 by 11 and the color of the film box to a particular color. <code>FILMBOX 8.5, 11.0, "SeaGreen"</code>

## FLAGS

Purpose	Use this command to modify the flags variable associated with a design.
Menu command	There are no menu commands to affect this setting: the CTRL+F shortcut key can be used, however.
Syntax	FLAGS exp
Parameters	
exp	An expression describing the value to assign to the flags field.
Description	This function modifies the system's environment flags. The flags field controls GerbTool settings that are either too infrequently used or too recently added to the system to have a more conventional menu access. This command is intended for use by software developers. The values used by the flags may change at any time, and improper usage may have unpredictable results.
Example	No example is provided for this function.

## FORMAT

Purpose	Use this command to change the file format parameters for a specific file type.
Menu command	Format (File menu) or Edit (Layers menu)
Syntax	<pre> FORMAT          string                 DIALECT          string                 M.N              exp, exp                 MODE              string                 ZEROSUPPRESSION  string                 TERMINATOR       string                 CHARSET          string                 METRIC           yesno                 MODEL            yesno                 NETS             yesno                 USERDATA        yesno                 GCMDS           yesno                 ARCS360         yesno                 ARCSMODAL       yesno                 HONORCRLF       yesno                 COMMENTS         yesno                 END </pre>
Parameters	
FORMAT	The selected file type: Gerber, or Drill.
DIALECT	A supported dialect such as Excellon or RS247X.
M.N	A pair of expressions specifying the m.n of the specified format.
MODE	A (Absolute) or I (Incremental).
ZEROSUPPRESSION	L (Leading), T (Trailing), or N (None).
TERMINATOR	A string indicating which characters should be output at the end of every line of output when writing a file of this type.
CHARSET	ASCII, EBCDIC, or EIA.
Description	Use this command to control the format of the specified file type. Normally, this command operates on a global format. To operate on a local format you can use this command from within a LAYERN block.



## Example

The following example modifies the format associated with the layer specified in the \$lyrno variable regardless if it is a local or global format.

```
LAYERN $lyrno
    FORMAT "Gerber"
        NETS $$YES
        METRIC $$NO
        MODAL $$YES
        MODE "A"
        TERMINATOR "*\r\n"
    END
END
```

## GRIDSIZE

Purpose	Using this function, you can change the size of the display grid.
Menu command	<i>Options/Grid</i>
Syntax	GRIDSIZE <i>xsize, ysize</i>
Parameters	
<i>xsize</i>	An expression specifying the horizontal spacing of the grid.
<i>ysize</i>	An expression specifying the vertical spacing of the grid.
Description	You can change the grid that is displayed in GerbTool. Note that the macros themselves do not make use of the grid.
Example	The following macro changes the grid 0.1 inches horizontally and 0.15 inches vertically.  GRIDSIZE 0.1, 0.15

**GRIDSNAP**

Purpose	Use this command to change the grid snap settings.
Menu command	Grid (Options menu)
Syntax	GRIDSNAP <i>yesno</i>
Parameters	
<i>yesno</i>	Your choice of the values <code>\$\$YES</code> and <code>\$\$NO</code> .
Description	You can change the grid snap settings used in GerbTool. Note that the macros themselves do not make use of the grid.
Example	The following macro turns on grid snap. <code>GRIDSNAP \$\$YES</code>

## GRIDVISIBLE

Purpose	Use this command to to change the visibility of the GerbTool grid.
Menu command	Grid (Options menu)
Syntax	GRIDVISIBLE <i>yesno</i>
Parameters	
<i>yesno</i>	Your choice of the values \$\$YES and \$\$NO.
Description	You can change the grid visibility settings used in GerbTool. Note that the macros themselves do not make use of the grid.
Example	The following macro turns on grid visibility. GRIDVISIBLE \$\$YES

**HILICOLORS**

Purpose	Use this command to change the highlight colors used by the system.
Menu command	Defaults (Options menu)
Syntax	<pre>HILICOLORS     QUERY    color     SELECT   color     DRC      color END</pre>
Parameters	<p><b>QUERY</b> A text string indicating the color to use when highlighting items while using the Item Info command (Query menu).</p> <p><b>SELECT</b> A text string indicating the color to use when highlighting items while using the Select command (Edit menu).</p> <p><b>DRC</b> A text string indicating the color to use when performing a DRC.</p>
Description	Use this function to change the highlight colors that are used by the system. The color string can be any of the colors given in the color file COLOR.RGB, even if these colors are not available in GerbTool's color chooser. Since this is a block command, only the variables that you want to change need to be used.
Example	<p>The following macro sets the color used to display DRC errors and select groups.</p> <pre>HILICOLORS     SELECT  \$selectColor     DRC     "vga16magenta" END</pre>

**LAYERN**

**Purpose** Use this command to change the settings of a particular Gerber layer.

**Menu command** Edit (Layers menu)

**Syntax**

```
LAYERN      layer
           FILE      string
           LYRNAME   string
           APLIST    string
           VISIBILITY exp
           FLASHCOLOR string
           DRAWCOLOR string
           TYPE      string
           POLARITY  string
           KEY       exp
           FTYPE     string
           EXTENTS   lx, ly, ux, uy
           NETID     exp
           VIRTUAL   exp
END
```

**Parameters**

**LAYERN** An expression indicating the layer to process.

**FILE** A text string indicating the filename to associate with this layer.

**LYRNAME** A text string indicating the layer name to associate with this layer.

**APLIST** A text string indicating the aperture list name to associate with this layer.

**VISIBILITY** An expression used to control the visibility of this layer. Valid values are: 0 (off), 1 (on), and 2(ref).

**FLASHCOLOR** A text string indicating the color to use when displaying flashes from this layer.

**DRAWCOLOR** A text string indicating the color to use when displaying draws from this layer.

**TYPE** A text string indicating the type for this layer. Valid values are Top, Bottom, Inner, Plane, Composite.

**POLARITY** A text string indicating the polarity of this layer in composite formats such as extended Gerber. Valid settings are dark and clear.

KEY	An expression used to indicate the key field used for this layer in composite formats such as extended Gerber.
FTYPE	A text string used to indicate the file type for this layer. This variable is for future expansion and as such the only valid value currently is Gerber.
EXTENTS	A quad of expressions used to set the extents of this layer. Note that these settings are only temporary, and can be changed when another function is called.
NETID	An expression used to indicate the net id for this layer. This value is used to determine layer to layer netlist accuracy, and should not generally be modified.
VIRTUAL	An expression used to indicate whether the layer should be included in a virtual panelization and step and repeat patterns.
Description	Use this function to set many of the parameters concerning individual layers in GerbTool. Since this is a block command, only the variables that you want to change need to be used.
Example	<p>The following example obtains the visibility of a layer, stores it in the layers net id and then turns the visibility of the layer off. The stored information can then be used later to restore the layers visibility to its original condition. Note how only the variables for the information we want to change are included in the command.</p> <pre> GetLayer \$Layer,\$fn,\$ln,\$an,\$vis,\$fc, \         \$dc,\$type,\$pol,\$key,\$ft, \         \$lx,\$ly,\$ux,\$uy, \$netid LAYERN \$Layer    #Set this layers info         Netid \$vis #save true visibility         Visibility 0 #turn the layers vis off END </pre>

## MAPPATH

Purpose	Use this command to change the location that GerbTool uses to find aperture lists.
Menu command	None
Syntax	<code>MAPPATH path</code>
Parameters	
<code>path</code>	A string defining the new path.
Description	This function accepts a character string and uses it to change the path GerbTool uses to find its aperture lists. This can be used to change the location where files are to be saved or loaded from.
Example	The following example changes directory where GerbTool looks for aperture lists to a temporary one. <code>MAPPATH "C:\temp\aperturs"</code>



**NETID**

Purpose	Use this command to change the net id number for a layer.
Menu command	None
Syntax	NETID exp
Parameters	
exp	An expression representing the new value to set the netid to.
Description	Use this function to set the net ID value for the currently active layer. Since these values are only used internally to keep the layer to layer net lists in sync, there are few instances where you should attempt to modify this value. One thing this function can be used for in a macro is for storing information with a layer, that can be retrieved and used later. Doing this destroys the layer to layer netlist information however, and the macro writer should be sure to warn you of this fact.
Example	<p>The following example obtains the visibility of a layer, stores it in the layers net id and then turns the visibility of the layer off. The stored information can then be used later to restore the layer's visibility to its original condition.</p> <pre> GetLayer \$Layer, \$fn, \$ln, \$an, \$vis, \$fc, \$dc, \         \$type, \$pol, \$key, \$ft, \$lx, \$ly, \$ux, \         \$uy \$netid LAYERN \$Layer          # Chg settings for this lyr         Netid \$vis      # save true visibility         Visibility 0    # turn lyr visibility off END </pre>

## OFFSETS

Purpose	Use this command to change the offsets applied to files loaded and merged into the system.
Menu command	Offsets (File menu)
Syntax	<code>OFFSETS xoff, yoff</code>
Parameters	
xoff	An expression describing the offset to be applied to the x coordinates of objects.
yoff	An expression describing the offset to be applied to the y coordinates of objects.
Description	Use this function to apply an offset to any files loaded or merged into the system.
Example	<p>The following example merges in a Gerber file 2 inches to the left and 3 inches up from where it would normally be located.</p> <pre>OFFSETS -2.0, 3.0 MERGEGERBER "infile.gbr"     # reset the offsets so they will     # not affect future merging. OFFSETS 0,0</pre>

**OVERLAYMODE**

Purpose	Use this command to change the state of the overlay mode setting.
Menu command	Overlay (Options menu)
Syntax	OVERLAYMODE <i>yesno</i>
Parameters	
<i>yesno</i>	Your choice of the values <code>\$\$YES</code> and <code>\$\$NO</code> .
Description	Use this function to change the state of the overlay mode that GerbTool is currently using.
Example	The following example turns overlay mode off. OVERLAYMODE <code>\$\$YES</code>

**PREVIOUSVIEW**

Purpose	This command allows a macro to redefine the previous view.
Menu command	Previous (View menu)
Syntax	<code>PREVIOUSVIEW lx, ly, ux, uy</code>
Parameters	
lx	An expression defining the x coordinate of the lower left corner of the view you are defining.
ly	An expression defining the y coordinate of the lower left corner of the view you are defining.
ux	An expression defining the x coordinate of the upper right corner of the view you are defining.
uy	An expression defining the y coordinate of the upper right corner of the view you are defining.
Description	This function allows a macro to redefine the previous view that is held internally by GerbTool. Normally this value is updated automatically whenever the user performs any of the View commands. One use of this command is to restore a view that was saved when first starting a macro. This way, once the macro has finished, the users viewing ability will not be changed.
Example	The following example sets the previous view to pre-calculated values.  <code>PREVIOUSVIEW \$lx, \$ly, \$ux, \$uy</code>

**SCALE**

Purpose	Use this command to allow a macro to change the scale of a file that is loaded or merged into GerbTool.
Menu command	Offsets (File menu)
Syntax	<code>SCALE xscale, yscale</code>
Parameters	
<b>xscale</b>	An expression giving the scale factor to be applied in the x dimension.
<b>yscale</b>	An expression giving the scale factor to be applied in the y dimension.
Description	Use this command to change the scale of files that are loaded or merged into GerbTool. Note that this only effects the coordinates of the Gerber file. If you want to modify the sizes of apertures, you should use the <code>DCODESCALE</code> command.
Example	<p>The following example merges in a Gerber file and increases the scale of the file by 0.5 percent in the x dimension. Small changes like this are often used to make allowances for shrinkage.</p> <pre>SCALE          1.005, 0.0 MERGEGERBER  "infile.gbr" OFFSETS 0,0  #reset the offsets so               # they will not affect               # future merging.</pre>

## SHOWERRORS

Purpose	This command controls the state of the show errors option in GerbTool.
Menu command	Show Errs (Options menu)
Syntax	SHOWERRORS <i>yesno</i>
Parameters	
<i>yesno</i>	Your choice of the values \$\$YES and \$\$NO.
Description	This function controls the state of the Show errors setting in GerbTool.
Example	The following example turns on the show errors setting. SHOWERRORS \$\$YES

**SKETCHMODE**

Purpose	Use this command to change the state of the sketch mode setting.
Menu command	Sketch (Options menu)
Syntax	<code>SKETCHMODE yesno</code>
Parameters	
<code>yesno</code>	Your choice of the values <code>\$\$YES</code> and <code>\$\$NO</code> .
Description	Use this function to change the state of the sketch mode that GerbTool is currently using.
Example	The following example turns sketch mode off. <code>SKETCHMODE \$\$NO</code>

## UNDO

Purpose	This function controls the state of the system undo.
Menu command	Undo (Options menu)
Syntax	UNDO <i>yesno</i>
Parameters	
<i>yesno</i>	Your choice of the values <code>\$\$YES</code> and <code>\$\$NO</code> .
Description	Use this function to toggle the state of the undo queue inside a GerbTool macro. Note that turning off undo destroys all of the current undo information.
Example	The following example disables the undo queue and destroys any existing undo information.  UNDO <code>\$\$NO</code>



**VIEWCOMPOSITES**

Purpose	Use this command to change the setting that determines how Composite extended Gerber files are viewed.
Menu command	Edit (Layers menu, View composites button)
Syntax	<code>VIEWCOMPOSITES yesno</code>
Parameters	
<code>yesno</code>	Your choice of the values <code>\$\$YES</code> and <code>\$\$NO</code> .
Description	Use this function to change the state of the View Composites setting that GerbTool is currently using. This setting only affects the viewing of composited extended Gerber files.
Example	The following example turns composite viewing on. <code>VIEWCOMPOSITES \$\$YES</code>

## VIEWMETRIC

Purpose	Use this command to control whether files are viewed in Metric or Inch units.
Menu command	Metric (Options menu)
Syntax	VIEWMETRIC yesno
Parameters	
yesno	Your choice of the values \$\$YES and \$\$NO.
Description	This command controls whether the coordinates used to display the Gerber files are presented in Metric or Inch units.
Example	The following macro turns off metric mode so the files will be displayed in inches.  VIEWMETRIC \$\$NO

*File handling functions***FILECLOSE**

Purpose	Use this command to close a previously opened disk file.
Menu command	None
Syntax	FILECLOSE fid
Parameters	
fid	A file id returned by a previous call to FILEOPEN.
Description	This function closes a file that was previously opened by a call to FILEOPEN.
Example	See FILEOPEN for an example of closing a file.
See also	FILEOPEN, FILEREAD, FILEWRITE

**FILEOPEN**

Purpose	Use this command to open a disk file for reading or writing.
Menu command	None
Syntax	<code>FILEOPEN fid, filename, mode</code>
Parameters	
fid	A numeric variable that will receive a file identification number.
filename	A string containing the filename of a disk file to open.
mode	A string containing a mode string of <code>r</code> for reading, <code>w</code> for writing, and <code>a</code> for append.
Description	<p>This function attempts to open the specified disk file in the mode indicated by the mode parameter. The value returned in the <code>fid</code> parameter can be used in subsequent calls to <code>FILEREAD</code>, <code>FILEWRITE</code> and <code>FILECLOSE</code>.</p> <p>WARNING: Opening an existing file in write mode will destroy any data that previously existed in the file.</p>
Example	<p>The following example opens a file for writing, processes the file, and then closes it.</p> <pre>FILEOPEN \$fid, "MYCMD.LOG", "w" ...file processing... FILECLOSE \$fid</pre>
See also	<code>FILECLOSE</code> , <code>FILEREAD</code> , <code>FILEWRITE</code>

**FILEREAD**

Purpose	Use this command to read a line of data from a disk file.
Menu command	None
Syntax	<code>FILEREAD fid, format, output_variables</code>
Parameters	
<i>fid</i>	A file id returned by a previous call to <code>FILEOPEN</code> .
<i>format</i>	A string describing the fields in the input line.
<i>output_variables</i>	A list of variables that will receive the data from the input line.
Description	This function reads a line from the input file and converts the data into individual string and numeric variables. The first parameter is the file id created by a previous call to <code>FILEOPEN</code> . The remaining parameters are exactly like those in the <code>STRREAD</code> command.
Example	The following example reads a line from a file and parses it into a string variable and two numeric variables. <code>FILEREAD \$fid, "%s %n %n", \$ref, \$x, \$y</code>
See also	<code>FILEOPEN</code> , <code>FILECLOSE</code> , <code>FILEWRITE</code>

**FILEWRITE**

Purpose	Use this command to output a formatted string to a disk file.
Menu command	None
Syntax	<code>FILEWRITE fid, format, input_variables</code>
Parameters	
fid	A file id returned by a previous call to <code>FILEOPEN</code> .
format	A string describing the variables that will used to construct the output line.
input_variables	A list of variables that will provide the data for the output line.
Description	This function writes to the output file converting the input data. The first parameter is a file id created by a previous call to the <code>FILEOPEN</code> command. The remaining parameters are exactly like those of the <code>STRWRITE</code> command.
Example	The following example outputs a line to a file consisting of a string and two numbers. <pre>FILEWRITE    \$fid, "%s %n %n", \$ref, \             \$x + 10.5, \$y</pre>
See also	<code>FILEOPEN</code> , <code>FILECLOSE</code> , <code>FILEREAD</code>

*File merging functions***MERGEDRILL**

Purpose	Use this command to merge (import) a drill file into the active layer.
Menu command	Drill (Import item, File menu)
Syntax	<code>MERGEDRILL filename</code>
Parameters	
filename	A string defining the disk file to import.
Description	This function imports a NC Drill file into the currently active layer using the current Drill format setup. All drill hits become Gerber flashes appended to the active layer.
Example	The following example changes the active layer and then imports a NC Drill file.  <code>ACTIVELAYER \$drilllayer</code> <code>MERGEDRILL \$drillfile</code>
See also	<code>FORMAT</code> , <code>MERGEGERBER</code> , <code>MERGEHPGL</code>

## MERGEGERBER

Purpose	Use this command to merge a Gerber file into the active layer.
Menu command	Gerber (Merge item, File menu)
Syntax	<code>MERGEGERBER filename</code>
Parameters	
filename	A string defining the disk file to merge.
Description	This function merges a Gerber file into the currently active layer using the current Gerber format associated with the active layer.
Example	The following example changes the active layer and then merges in a NC Gerber file.  <code>ACTIVELAYER \$mergelayer</code> <code>MERGEGERBER \$newfile</code>
See also	<code>FORMAT</code> , <code>MERGEDRILL</code> , <code>MERGEHPGL</code>



**MERGEHPGL**

Purpose	To merge (import) a HPGL plot file into the active layer.
Menu command	HPGL (Import item, File menu)
Syntax	<pre>MERGEHPGL    filename               PLOTSIZE    string               ROTATE      yesno               PEN          exp, exp               GO           string               END</pre>
Parameters	
filename	A string defining the disk file to import.
PLOTSIZE	A string defining the expected size of the plotter that the input file was generated for. Use S (small) for A/B plotters and L (large) for C/D/E plotters.
PEN	A pair of expressions matching a pen number to a D-Code.
GO	A string indicating the filename of the file to import.
Description	This function imports an HPGL file into the currently active layer using the current HPGL format setup. All data becomes Gerber flashes/draws on the active layer.
Example	<p>The following example changes the active layer and then imports a HPGL file.</p> <pre>ACTIVELAYER \$hpgllayer MERGEHPGL \$hpglfile</pre>
See also	FORMAT, MERGEDRILL, MERGEGERBER

## Mathematical functions

### ABS

Purpose	This function calculates the absolute value of a given value.
Menu command	None
Syntax	<code>ABS numvar = exp</code>
Parameters	
numvar	A numeric variable that is assigned the absolute value of the expression on the right hand side of the equal sign.
exp	The expression to take the absolute value of.
Description	You can calculate the absolute value of a number. This is a number with the negative sign removed if one exists. The resulting value is assigned to numvar. While the object on the right hand side of the equal sign can be any numeric expression, the value on the left of the equal sign must be a numeric variable.
Example	The following example takes the absolute value of -7.0 and assigns its value (7.0) to the variable \$answer.  <code>ABS \$answer = -7.0 * \$zaxis</code>

**ARRAY**

Purpose	Use this function to create an array of numeric variables.
Menu command	None
Syntax	ARRAY \$name (size)
Parameters	
name	The name of the variable.
size	An expression indicating the size of the array.
Description	This function creates an array of numeric variables that can be accessed with one name and a index expression.
Example	The following example shows how to fill an array of items. <pre>ARRAY \$v(50) CALC \$index = 1 REPEAT \$index &lt;= 50     GETVALUE "Enter Next Value: ", \$val     CALC \$v(\$index) = \$val END</pre>
See also	STRARRAY

**ASIN**

Purpose	This function calculates the principal value of the arcsin function of a given value.
Menu command	None
Syntax	<code>ASIN numvar = exp</code>
Parameters	
numvar	A numeric variable that is assigned the arcsin of exp.
exp	An expression to take the arcsin of.
Description	Use this function to calculate the arcsin of a number. The resulting value will be assigned to numvar. While the object on the right side of the equal sign can be either a numeric variable or numeric literal, the value on the left of the equal sign must be a numeric variable. Note that an error occurs if the argument is less than -1.0 or greater than 1.0.
Example	The following example takes the arcsin of 0.5984271 and assigns its value (2.5 radians) to the variable \$answer.  <code>ASIN \$answer = 0.5984271</code>

**ATAN**

Purpose	This function calculates the principal value of the arctangent function of a given value.
Menu command	None
Syntax	<code>ATAN numvar = exp</code>
Parameters	
numvar	A numeric variable that is assigned the arctangent of exp.
exp	The expression to take the arctangent of.
Description	You can calculate the arctangent of a number. The resulting value will be assigned to numvar. While the object on the right side of the equal sign can be either a numeric variable or numeric literal, the value on the left of the equal sign must be a numeric variable.
Example	<p>The following example takes the arctangent of -0.7470223 and assigns its value (2.5 radians) to the variable \$answer.</p> <pre>ATAN \$answer = -0.7470223</pre>

**CALC**

Purpose	Use this function to perform mathematical calculation.
Menu command	None
Syntax	<code>CALC numvar = exp1 operator exp1</code>
Parameters	
numvar	A numeric variable that is assigned the results of the mathematical operation.
exp1	The first variable or literal used in the calculation.
operator	The mathematical operation to perform. Valid operations are + - / *.
exp2	The second variable or literal used in the calculation.
Description	Use this function to perform mathematical calculations inside a macro. While the objects on the right side of the equal sign can be either numeric variables or numeric literals, the value on the left of the equal sign must be a numeric variable. Like any other programming language, it is possible for a variable to be on both sides of the equal sign at the same time. The result of doing this is that the value of the variable is first used in the calculation, and the answer is placed back into the variable.
Example	<p>The following example takes the contents of the variable \$size, multiplies it by 2.5 and assigns the result to the variable \$result.</p> <pre>CALC \$result = \$size * 2.5</pre>

## CEIL

Purpose	This function calculates the smallest integer not less than the input value.
Menu command	None
Syntax	CEIL numvar = exp
Parameters	
numvar	A numeric variable that is assigned the ceiling value of the expression on the right hand side of the equal sign.
exp	The expression to take the absolute value of.
Description	Use this function to find the largest integer value of a number.
Example	The following example takes the ceiling value of -7.658 and assigns its value (-7.0) to the variable \$answer.  CEIL \$answer = -7.658

## CSIN

Purpose	This function calculates the trigonometric cosin value of a given value.
Menu command	None
Syntax	<code>CSIN numvar = exp</code>
Parameters	
numvar	A numeric variable that is assigned the cosin of the expression.
exp	The expression from which the cosin is derived.
Description	Use this function to calculate the cosin of a number. The resulting value will be assigned to numvar While the object on the right side of the equal sign can be either a numeric variable or numeric literal, the value on the left of the equal sign must be a numeric variable. Note that this functions assumes the value given it is represented in radians and not degrees.
Example	The following example takes the cosin of 2.5 radians and assigns its value (-0.8011436) to the variable \$answer.  <code>CSIN \$answer = 2.5</code>



**FLOOR**

Purpose	This function calculates the largest integer not greater than the input value.
Menu command	None
Syntax	FLOOR numvar = exp
Parameters	
numvar	A numeric variable that is assigned the floor value of the expression on the right hand side of the equal sign.
exp	The expression to take the floor value of.
Description	Use this function to find the smallest integer value of a floating point number.
Example	The following example takes the value of -7.658 and assigns the integer value (-8.0) to the variable \$answer.  FLOOR \$answer = -7.658

## ROUND

Purpose	This function calculates the nearest integer of the input value.
Menu command	None
Syntax	ROUND numvar = exp
Parameters	
numvar	A numeric variable that is assigned the nearest integer value of the expression on the right hand side of the equal sign.
exp	The expression to take the absolute value of.
Description	Use this function to round a floating point number to its nearest integer value.
Example	The following example rounds the input of 7.658 and assigns the result of 8.0 to the variable \$answer.  ABS \$answer = 7.658

**SET**

Purpose	This function creates and assigns a value to a numeric variable.
Menu command	None
Syntax	<code>SET numvar = exp</code>
Parameters	
numvar	A numeric variable that is assigned the value of exp.
exp	An expression whose value is assigned to numvar.
Description	Use this function to assign a value to a variable. While the object on the right side of the equal sign can be either a numeric variable or numeric literal, the value on the left of the equal sign must be a numeric variable.
Example	<p>The following example assigns the contents of the variable \$value to the variable \$var1 and the number 4.12 to the variable \$var2.</p> <pre>SET \$var1 = \$value SET \$var2 = 4.12</pre>

## SETGLOBAL

Purpose	This function creates and assigns a value to a global numeric variable.
Menu command	None
Syntax	<code>SETGLOBAL numvar = exp</code>
Parameters	
numvar	A numeric variable that is assigned the value of the expression.
exp	An expression whose value is assigned to numvar.
Description	Use this function to assign a value to a global variable. Global variables must be defined before any other variables are defined in a macro. Once set, global variables last the duration of your GerbTool session and are available to all macros. Global variables are persistent from one macro invocation to another.
Example	The following example shows some typical global variable assignments.  <code>SETGLOBAL \$gvar1 = 1.75</code> <code>SETGLOBAL \$gvar2 = \$gvar1 * 4.12</code>

**SIN**

Purpose	This function calculates the trigonometric sin of a value.
Menu command	None
Syntax	<code>SIN numvar = exp</code>
Parameters	
numvar	A numeric variable that is assigned the sin of exp.
exp	The variable or literal to take the sin of. This value is assumed to be in radians.
Description	Use this function to take the sin of a number. The resulting value will be assigned to numvar. While the object on the right side of the equal sign can be either a numeric variable or numeric literal, the value on the left of the equal sign must be a numeric variable. Note that this functions assumes the value given it is represented in radians and not degrees.
Example	The following example takes the sin of 2.5 radians and assigns its value (0.5984721) to the variable \$answer.  <code>SIN \$answer = 2.5</code>

## SQRT

Purpose	This function calculates the square root of a value.
Menu command	None
Syntax	<code>SQRT numvar = exp</code>
Parameters	
numvar	A numeric variable that is assigned the square root of exp.
exp	The variable or literal to take the square root of.
Description	Use this function to take the square root of a number. The resulting value will be assigned to numvar. While the object on the right side of the equal sign can be either a numeric variable or numeric literal, the value on the left of the equal sign must be a numeric variable. Attempting to take the square root of a negative number will result in an error.
Example	The following example takes the square root of 36 and assigns its value (6) to the variable \$answer.  <code>SQRT \$answer = 36</code>

**TAN**

Purpose	This function calculates the trigonometric tangent of a given value.
Menu command	None
Syntax	TAN numvar = exp
Parameters	
numvar	A numeric variable that is assigned the tangent of exp.
exp	The variable or literal to take the tangent of.
Description	Use this function to calculate the tangent of a number. The resulting value will be assigned to numvar. While the object on the right side of the equal sign can be either a numeric variable or numeric literal, the value on the left of the equal sign must be a numeric variable. Note that this functions assumes the value given it is represented in radians and not degrees. Also, attempting to produce the tangent of odd multiples of $\pi/2$ will produce an error.
Example	The following example takes the tangent of 2.5 radians and assigns its value (-0.7470223) to the variable \$answer.  TAN \$answer = 2.5

## Plotting functions

### PLOTHPGL

**Purpose** Use this function to create an HPGL plot.

**Menu command** HPGL (Plot item, Files menu)

**Syntax** PLOTHPGL

```

        OUTFILE          string
        MEDIASIZE       exp, exp
        SCALE           exp
        MODE            string
        ROTATE          yesno
        BORDER          yesno
        BORDERTEXT     string
        BORDERPEN      exp
        PENWIDTH       exp
        PENSPEED       exp
        PADSONLY       yesno
        GO
    
```

END

#### Parameters

**OUTFILE** A text string specifying the name of the file to place the plot.

**MEDIASIZE** A pair of expressions specifying the horizontal and vertical size of the printable area of the output media.

**OFFSETS** A pair of expressions specifying the horizontal and vertical offsets. This can be used to position the plot at a specific location on the media.

**SCALE** An expression specifying the scale factor to apply to the plot.

**MODE** A string indicating the plotting mode. Choose from “S” (Sketch), “O” (Outline), or “F” (Fill).

**BORDERTEXT** This parameter allows you to specify the text that will appear in the border, if borders are enabled. GerbTool looks for the key words \$DATE, \$TIME, \$DESIGN and \$PROG. NOTE: These are not variable names. If GerbTool finds any of these keywords they will be replaced with the appropriate text. All other text specified will be included in the border verbatim.



BORDERPEN	An expression specifying the plotter pen to use when drawing the border if enabled.
PENWIDTH	An expression specifying the size of the pens used.
PENSPEED	An expression specifying the speed of the plotter pens.
GO	Executes the command using the current parameters.
Description	This function plots the currently visible layer to a HPGL plotter.
Example	<p>The following example plots all loaded layers, one layer per output file, using a predefined plotting scale and media size.</p> <pre> CALLMACRO "MaxLoadedLayer", \$maxlyr SET \$lyr = 1 REPEAT \$lyr &lt;= \$maxlyr     ACTIVELAYER \$lyr     IF \$\$STATUS         CALLMACRO "GetLayerFileName", \$lfn         SPLITPATH \$lfn, \$dir, \$fn, \$ext         STRWRITE \$fn, "%s.%0n", \$fn, \$lyr         PLOTHPGL             OUTFILE      \$fn             MEDIASIZE \$mediax, \$mediay             SCALE \$fscale             GO     END END     CALC \$lyr = \$lyr + 1 END </pre>

**PLOTPS**

Purpose	Use this function to create an PostScript plot.
Menu command	PostScript (Plot item, Files menu)
Syntax	<pre> PLOTPS     OUTFILE          string     MEDIASIZE        exp, exp     OFFSET            exp, exp     SCALE             exp     MODE              string     ROTATE            yesno     WINDOWMODE        yesno     PADSONLY          yesno     BORDER            yesno     BORDERTEXT        string     GREYSCALE         yesno     FITTOPAGE         yesno     GO END </pre>
Parameters	
OUTFILE	A text string specifying the name of the output file.
MEDIASIZE	A pair of expressions specifying the horizontal and vertical size of the printable area of the output media.
OFFSETS	A pair of expressions specifying the horizontal and vertical offsets. This can be used to position the plot at a specific location on the media.
SCALE	An expression specifying the scale factor for the plot.
MODE	A string indicating the plotting mode. Choose from S (sketch) or F (fill).
BORDERTEXT	Use this parameter to specify the text to appear in the border, if borders are enabled. GerbTool looks for the key words \$DATE, \$TIME, \$DESIGN and \$PROG. NOTE: These are not variable names. If GerbTool finds any of these keywords they will be replaced with the appropriate text. All other text specified will be included in the border verbatim.
GO	Executes the command using the current parameters.
Description	This function plots the currently visible layer to a PostScript plotter.

## Example

The following example plots all of the visible layers using the FITTOPAGE option to automatically scale the image to the media size.

```
PLOTPS
```

```
    OUTFILE          "output.ps"  
    MEDIASIZE        $mediax, $mediay  
    OFFSETS          0.0, 0.0  
    FITTOPAGE        $$YES  
    GO
```

```
END
```

## Query functions

### HILIDCODE

Purpose	You can highlight items in their database based on specified criteria.
Menu command	Dcode (Highlight item, Query menu)
Syntax	<pre>HILIDCODE         FLASHES          yesno         DRAWS            yesno         ARCS             yesno         DCODE            exp         LAYER            exp END</pre>
Parameters	
DCODE	An expression that describes the D-Code you are interested in highlighting. Use a value of zero to select all D-Codes.
LAYER	An expression that describes the layer you are interested in highlighting objects from. Use a value of zero to select from all visible layers.
Description	Use this function to highlight objects on the screen that match the given criteria. Note that since this is a block command, only the variables that you want to change need to be given when using the command.
Example	The following example highlights all of the flashes on layer 10 that use D-Code 45.

```
HILIDCODE
        FLASHES          $$YES
        DRAWS            $$NO
        ARCS             $$NO
        DCODE            45
        LAYER            10
END
```

**HILIITEM**

Purpose	Use this command to highlight an item.
Menu command	None
Syntax	<code>HILIITEM layer, seqno</code>
Parameters	
layer	A numeric variable containing the layer of the item.
seqno	A numeric variable containing the sequence number of the item.
Description	This function highlights an item using the current regular highlight color. The <code>layer</code> and <code>seqno</code> parameters would normally be returned by <code>GETFIRSTITEM/GETNEXTITEM</code> calls.
Example	<p>The following example scans the database highlighting all flashes.</p> <pre> REPEAT \$\$STATUS   IF \$type == \$\$FLASH     HILIITEM \$layer, \$seqno   END GETNEXTITEM END </pre>

**MEASUREE2E**

Purpose	Use this function to measure the minimum distance between two database items.
Menu command	Edge 2 Edge (Measure item, Query menu)
Syntax	MEASUREE2E layer, seqno1, seqno2
Parameters	
layer	A numeric variable containing the layer of the item.
seqno1	A numeric variable containing the sequence number of a item.
seqno2	A numeric variable containing the sequence number of a item.
Description	This function measures the minimum distance between two database items. The calculated distance is returned in the \$\$STATUS variable. The layer and seqno1/seqno2 parameters would normally be returned by GETFIRSTITEM/GETNEXTITEM calls.
Example	<p>The following example scans the database measuring the distance between flashes.</p> <pre>REPEAT \$\$STATUS   IF \$type == \$\$FLASH     MEASUREE2E \$layer,\$seqno,\$lastSeqno     IF \$\$STATUS &lt; \$minDist       STOP "Too Close!"     END     Calc \$lastSeqno = \$seqno   END GETNEXTITEM END</pre>

## String handling functions

### STRARRAY

Purpose	Use this function to create an array of string variables.
Menu command	None
Syntax	<code>STRARRAY \$name(size)</code>
Parameters	
name	The name of the variable.
size	An expression indicating the size of the array.
Description	This function creates an array of string variables that can be accessed with one name and a index expression. Each element of the array can hold up to 256 characters.
Example	The following example shows how to fill an array of items. <pre>STRARRAY \$s(50) CALC \$index = 1 REPEAT \$index &lt;= 50     GETSTRING "Enter Next Line:", \$str     STRCPY \$s(\$index), \$str END</pre>
See also	ARRAY

## STRCAT

Purpose	Use this function to concatenate two strings.
Menu command	None
Syntax	<code>STRCAT destination, source</code>
Parameters	
destination	A string variable to which the source string will be appended.
source	A string that will be appended to the destination string variable.
Description	This function copies the source string to the end of the destination string variable.
Example	<p>The following example appends the string literal "400" to the \$refdesg string variable. The value of \$refdesg following the STRCAT function would be "U400".</p> <pre>STRSET \$refdesg, "U" STRCAT \$refdesg, "400"</pre>



**STRCMP**

Purpose	Use this function to determine if two strings are equal.
Menu command	None
Syntax	STRCMP <i>string1</i> , <i>string2</i>
Parameters	
<i>string1</i>	A string.
<i>string2</i>	A string.
Description	This function compares two strings, without regard to case, and determines whether <i>string1</i> is less than, equal or greater than <i>string2</i> . A string is less than another when it would come first in the ASCII collating sequence. A value of zero indicates that the two strings are equal.
Example	In the following example <code>\$\$STATUS</code> would contain a value less than zero, indicating that ONE is less than TWO.  STRCMP "ONE", "TWO"

## STRCPY

Purpose	Use this function to make a copy of a text string.
Menu command	None
Syntax	<code>STRCPY destination, source</code>
Parameters	
destination	A string variable that will receive a copy of the source string.
source	A string that will be copied into the destination string variable.
Description	This function copies a string variable or literal into another string variable.
Example	The following example sets the value of <code>\$refdes</code> to "U400".  <code>STRCPY \$refdes, "U400"</code>

**STRLEN**

Purpose	Use this function to calculate the number of characters in a string.
Menu command	None
Syntax	STRLEN string
Parameters	
string	A string.
Description	This function counts the length of a string in characters. The calculated length is returned in the \$\$STATUS variable.
Example	In the following example \$\$STATUS would contain the value of 12 after executing the STRLEN function.  STRLEN "Short string"

**STRLOC**

Purpose	Use this function to find an occurrence of a string within another string.
Menu command	None
Syntax	<code>STRLOC source, search</code>
Parameters	
source	The string to search in.
search	The string to search for.
Description	You can locate the search string anywhere within the source string. If found, the <code>\$\$STATUS</code> variable contains the index of the first matching character within the source string.
Example	In the following example <code>\$\$STATUS</code> contains the value 3 after the <code>STRLOC</code> function is executed.  <pre>STRSET \$line, "This is a test" STRLOC \$line, "is"</pre>

**STRREAD**

Purpose	Use this function to parse a line of text into a series of variables.
Menu command	None
Syntax	<code>STRREAD source, format, output_variables</code>
Parameters	
source	A text string.
format	A string describing the format of the data fields in the input line.
output_variables	A list of variables that will receive the data from the input line.
Description	<p>This function reads the <code>source</code> string, converts the data according to the <code>format</code> string and places the converted data into individual string and numeric variables. The format string describes the position and type of each data field within the <code>source</code> string. White space characters in the source string are not converted and serve only to delimited the data fields. Within the format string <code>%s</code> matches a text string and <code>%n</code> matches a numeric value.</p> <p>Note: For those familiar with the C programming language, the format string is similar to the <code>scanf</code> format string, with <code>%n</code> being mapped to <code>%f</code>.</p>
Example	<p>The following example results in <code>\$ref</code> containing "U1" <code>\$x</code> containing 5.0, and <code>\$y</code> containing 4.25.</p> <pre>STRSET \$line = "U1 5.000 4.250" STRSET \$ref = "" SET \$x = 0 SET \$y = 0 STRREAD \$line, "%s %n %n", \$ref, \$x, \$y</pre>
See also	STRWRITE

**STRSET**

Purpose	This function creates and assigns a value to a string variable.
Menu command	None
Syntax	<code>STRSET strvar, string</code>
Parameters	
<code>strvar</code>	A string variable that is assigned the value of string.
<code>string</code>	A string whose value is assigned to strvar.
Description	Use this function to assign a value to a string variable. While the object on the right side of the comma can be either a string variable or literal, the value on the left of the comma must be a string variable.
Example	<p>The following example assigns the contents of the variable \$value to the variable \$var1 and the number 4.12 to the variable \$var2.</p> <pre>SETSTR \$var1, "This is a string" SETSTR \$var2, \$var1</pre>

**STRSETGLOBAL**

Purpose	This function creates and assigns a value to a global numeric variable.
Menu command	None
Syntax	<code>STRSETGLOBAL strvar, string</code>
Parameters	
<code>strvar</code>	A string variable that is assigned the value of string.
<code>string</code>	A string whose value is assigned to strvar.
Description	Use this function to assign a value to a global variable. Global variables must be defined before any other variables are defined in a macro. Once set, global variables last the duration of your GerbTool session and are available to all macros. Global variables are persistent from one macro invocation to another.
Example	The following example shows some typical global variable assignments.  <code>STRSETGLOBAL \$gvar1, "A Global Str Var"</code> <code>STRSETGLOBAL \$gvar2, \$gvar1</code>

## STRSUB

Purpose	Use this function to copy a portion of one string to another.
Menu command	None
Syntax	STRSUB destination, start, count, source
Parameters	
destination	A string variable that will receive the sub-string.
start	An expression indicating the index of the first character in the sub-string.
count	An expression indicating the number of characters in the sub-string.
source	A string that contains the sub-string.
Description	This function copies a specified portion of a string into another.
Example	<p>In the following example \$subline would contain the string "sub-string" after the STRSUB function executes.</p> <pre>STRSET \$line, "A small substring example" STRLOC \$line, "sub" STRSUB \$subline, \$\$STATUS, 10, \$line</pre>



**STRTOK**

Purpose	Use this function to parse a text string into individual tokens.
Menu command	None
Syntax	STRTOK destination, delimiters, source
Parameters	
destination	A string variable that will receive the string token.
delimiters	A string of characters that are used to separate tokens in the source string.
source	A string that contains the list of tokens or the numeric literal 0.
Description	This function parses the source string for sub-strings that are separated by any character present in the delimiters parameter. This function is meant to be initialized by a call with a valid source parameter. Subsequent calls are then made with the source parameter set to the numeric literal 0. As long as there are more tokens in the source parameter, this function continues to return the next string token. The \$\$STATUS variable is set to the length of the returned token. A \$\$STATUS value of zero, therefore, indicates that there are no more tokens.
Example	<p>In the following example, \$loops will contain the value 5 when the REPEAT block completes.</p> <pre> STRSET \$line, "This is a token test" SET \$loops = 0 STRTOK \$token, " ", \$line REPEAT \$\$STATUS &gt; 0     Calc \$loops = \$loops + 1     STRTOK \$token, " ", 0 END </pre>

**STRWRITE**

Purpose	Use this function to output a formatted string to a string variable.
Menu command	None
Syntax	<code>STRWRITE destination, format, input_variables</code>
Parameters	
destination	A string variable that will receive the formatted output string.
format	A string describing the variables that will used to construct the output line.
input_variables	A list of variables that will provide the data for the output line.
Description	<p>This function writes to the <code>destination</code> string converting the input data according to the <code>format</code> string. The <code>format</code> string contains a combination of conversion specifiers and characters that will be output verbatim. The <code>%</code> character is used to indicate a conversion specifier. Currently only two types are supported: <code>%s</code> and <code>%n</code>. When a <code>%s</code> is found, the next string input variable is output in its place. Similarly, if a <code>%n</code> is found, it is replaced with the value of the next numeric input variable. Each conversion specifier can be further enhanced with a precision specifier such as <code>%6.3n</code>. This indicates that the output data should be 6 characters wide with 3 places after the decimal point. For string conversions, the precision specifier allows you to specify a output field width such as <code>%20s</code>. The output is padded with spaces to achieve the proper width if the input variable is not very wide.</p> <p>Note: For those familiar with the C programming language, the format string is similar to the <code>printf</code> format string, with <code>%n</code> being mapped to <code>%f</code>.</p>
Example	<p>In the following example the variable <code>\$line</code> would contain the value <code>U10 10.500 5.500</code>.</p> <pre>STRWRITE \$line, "%10s %6.3n %6.3n", \     "U10", 10.5, 5.5</pre>
See also	STRREAD

## Tool functions

**DRAWNPADS**

**Purpose** Use this function to setup and execute the Convert Pads command.

**Menu command** Pads (Convert item, Tools menu)

**Syntax**

```
DRAWNPADS
        NEWDCODE      exp
        TOLERANCE     exp
        BY             exp
        BOUNDARY      yesno
        DRAWS         exp
        ARCS          exp
        DCODE         exp
        LAYER         exp
        GO            [lx, ly, ux, uy] . . .
END
```

**Parameters**

**NEWDCODE** Specifies the D-Code to replace found drawn pads.

**TOLERANCE** An expression specifying the allowable tolerance to use when finding matching drawn pads.

**BY** An expression indicating how to select items to convert. Valid values are: 1=window, 2=group.

**DCODE** An expression specifying the D-Code filter. Use zero to operate on all D-Codes.

**LAYER** An expression specifying the layer filter. Use zero to operate on all visible layers.

**GO** An optional list of expressions describing windows that enclose drawn pads that are to be converted.

**Description** This function converts drawn pads into flashes.

Example

This example converts all drawn pads that match the example drawn pad contained in the current select group.

```
DRAWNPAD
```

```
NEWDCODE          $newDcode  
TOLERANCE 0.004  
BY                $$GROUPMODE  
GO
```

```
END
```

**DRC**

**Purpose** Use this function to setup and execute the DRC command.

**Menu command** DRC (Tools menu)

**Syntax** DRC

```

    REPPFILE      string
    PAD2PAD       exp
    PAD2TRACE     exp
    TRACE2TRACE  exp
    MINFLASH     exp
    MINTRACE     exp
    MINRING      exp
    LAYER        exp
    DRILLLAYER   exp
    USETOOLSIZE  yesno
    WELLBEHAVED  yesno
    WINDOWMODE   yesno
    GO           [lx, ly, ux, uy]
  
```

END

**Parameters**

<b>REPPFILE</b>	A string containing the filename to which the report file will be written.
<b>PAD2PAD</b>	An expression indicating the minimum pad-to-pad spacing.
<b>PAD2TRACE</b>	An expression indicating the minimum pad-to-trace spacing.
<b>TRACE2TRACE</b>	An expression indicating the minimum trace-to-trace spacing.
<b>MINFLASH</b>	An expression indicating the minimum flash size allowed.
<b>MINTRACE</b>	An expression indicating the minimum trace size allowed.
<b>MINRING</b>	An expression that indicates the minimum annular ring allowed between drilled hole and pad size.
<b>LAYER</b>	An expression that indicates the layer on which to perform the DRC. A value of zero indicates that all signal layers are to be processed.

DRILLLAYER	An expression that indicates the drill layer to be used when performing the annular ring check.
GO	Executes the command using the current parameters, using an optional quad of expressions indicating a window to operate on.
Description	This function allows the setup and execution of the DRC command.
Example	The following example executes the DRC command.

```
DRC
  REPFIL  "drc.rep"
  PAD2PAD      0.006
  PAD2TRACE    0.006
  TRACE2TRACE  0.005
  MINFLASH     0.020
  MINTRACE     0.004
  MINRING      0.010
  LAYER        0
  DRILLLAYER   $drillLayer
  USETOOLSIZE  $$NO
  WELLBEHAVED $$YES
  WINDOWMODE   $$NO
GO
END
```

**DRILL**

**Purpose** Use this function to setup and execute the Drill command.

**Menu command** Drill (Tools menu)

**Syntax** DRILL

```

        OUTFILE          string
        REPPFILE        string
        SWATH           exp
        SORT            exp
        LAYER           exp
        MERGELAYER      exp
        WINDOWMODE     yesno
        GO              [lx, ly, ux, uy]

```

END

**Parameters**

**OUTFILE** A string containing the filename to which the drill file will be written.

**REPPFILE** A string containing the filename to which the report file will be written.

**SWATH** An expression indicating the swath size.

**SORT** A string indicating the sort method: None, X, or Y.

**LAYER** Indicates the layer to be used in generating the drill data.

**MERGELAYER** Indicates the layer to be merged into the normal drill data.

**GO** Uses the current parameters, using optional quad of expressions, to indicate a window to operate on.

**Description** Sets up and executes the Drill command. This command sets the \$\$DRILLHITS and \$\$DRILLTRAVEL system variables, as well as returning the hit count in \$\$STATUS.

Example

The following example executes the Drill command.

```
DRILL
    OUTFILE "final.drl"
    REPPFILE "final.rep"
    SWATH 0.100
    SORT "Y"
    LAYER $drillLayer
    WINDOWMODE $$NO
GO
END
```



**FIXSS**

**Purpose** Use this function to setup and execute the Fix SS command.

**Menu command** Fix SS (Tools menu)

**Syntax** FIXSS

```

        PADLAYER      exp
        SILKLAYER     exp
        SPACING       exp
        WINDOWMODE    yesno
        GO             [lx,ly,ux,uy] , . . .

```

END

**Parameters**

**PADLAYER** An expression indicating the pad master layer.

**SILKLAYER** An expression indicating the silkscreen layer.

**SPACING** An expression indicating the minimum spacing allowed between a pad and any silkscreen data.

**GO** An optional quad of expressions describing a window to operate on.

**Description** This function allows the setup and execution of the Fix SS command.

**Example** The following example executes the Fix SS command.

FIXSS

```

        PADLAYER      $padMaster
        SILKLAYER     $silkLayer
        SPACING       0.020
        WINDOWMODE    $$NO
        GO

```

END

**NETLIST**

Purpose	Use this function to generate a internal netlist and/or output such a netlist to a disk file.
Menu command	Netlist (Tools menu)
Syntax	<pre>NETLIST     OUTFILE          string     WELLBEHAVED     yesno     METRIC           yesno     M.N              exp, exp     GO               exp END</pre>
Parameters	
OUTFILE	A string indicating the output filename. This parameter is only required when writing a netlist to disk.
M.N	A pair of expressions specifying the m.n of the output file.
GO	An expression indicating whether to generate a netlist (0) or write an existing netlist to the output file (1) specified with the OUTFILE parameter.
Description	This function generates an internal netlist and optionally writes the netlist data to a disk file.
Example	<p>The following example executes the Netlist command.</p> <pre>NETLIST     WELLBEHAVED     \$\$YES     GO 0            #generate a netlist     OUTFILE         "final.net"     M.N             2, 4     METRIC          \$\$NO     GO 1            #write netlist to file END</pre>

**PADREMOVAL**

**Purpose** Use this function to set up and execute the Pad Removal command.

**Menu command** Pad Removal (Tools menu)

**Syntax**

```
PADREMOVAL
    TYPE          string
    LAYER         exp
    DCODE         exp
    WINDOWMODE   yesno
    GO            [lx, ly, ux, uy], ...
END
```

**Parameters**

**TYPE** A string indicating the type of pad removal to be performed: I (isolated) or S (stacked).

**LAYER** An expression that indicates the layer to perform the pad removal on. A value of zero indicates that all visible layers are to be processed.

**DCODE** An expression indicating a D-Code filter to be used when considering a pad for removal.

**GO** An optional quad of expressions describing a window to operate on.

**Description** You can set up and execute the Pad Removal command.

**Example** The following example executes the Pad Removal command.

```
PADREMOVAL
    TYPE          "I"    # isolated only
    LAYER         $$ACTIVE_LAYER
    DCODE         0
    WINDOWMODE   $$NO
    GO
END
```

**PANELIZE**

**Purpose** Use this function to setup and execute the Panelize command.

**Menu command** Panelize (Tools menu)

**Syntax** PANELIZE

```

ROWS          exp
COLS          exp
TABSIZe      exp, exp
VENTBORDER    exp
VENTSPACING  exp, exp
VENTDCODE    exp
VENTLAYER    exp
DOCUMENT     string
AUTOPANEL    yesno
AUTOVENT    yesno
VIRTUAL      yesno
GO           [lx,ly, ux,uy]

```

END

**Parameters**

**ROWS** An expression indicating the number of rows you want.

**COLS** An expression indicating the number of columns you want.

**TABSIZe** A pair of expressions indicating the X and Y spacing between images on the panel.

**VENTSPACING** A pair of expressions specifying the X and Y spacing between flashes in the vent pattern.

**VENTBORDER** An expression indicating the spacing to maintain between the vent pattern and each image.

**VENTDCODE** An expression that indicates the D-Code to be used in the vent pattern.

**VENTLAYER** An expression that indicates the layer to output the vent pattern into.

**DOCUMENT** A string containing the filename to which the report file will output.

**GO** Executes the command using the current parameters using optional quad of expressions indicating a window to operate on.

**Description** Use this function to setup and execute the GerbTool Panelize command. This command also sets the following system variables: \$PANELXOFF, \$PANELYOFF, \$PANELXSPACING, \$PANELYSPACING, \$PANELROWS, \$PANELCOLS.

**Example** The following example executes the Panelize command using auto panel and auto vent modes.

```
PANELIZE
    VENTBORDER      0.5
    VENTSPACING    0.25, 0.25
    VENTDCODE      250
    VENTLAYER      $$ACTIVE_LAYER
    DOCUMENT       "panel.rep"
    AUTOPANEL      $$YES
    AUTOVENT       $$YES
    VIRTUAL        $$YES
GO
END
```

**SEGMENTARCS**

**Purpose** Use this function to set up and execute the Convert Circles command.

**Menu command** Circles (Convert item, Tools menu)

**Syntax**

```

SEGMENTARCS
    CHORDANGLE    exp
    BY             exp
    BOUNDARY      yesno
    DCODE         exp
    LAYER         exp
    GO            [x1, y1, x2, y2 . . . ]
END

```

**Parameters**

**CHORDANGLE** An expression indicating the chord angle, in degrees, to be used when segmenting each arc.

**BY** An expression indicating how to select items to convert. Valid values are: 0=item, 1=window, 2=group.

**DCODE** An expression specifying the D-Code filter. Use zero to operate on all D-Codes.

**LAYER** An expression specifying the layer filter. Use zero to operate on all visible layers.

**GO** An optional list of expressions describing either locations of arcs or windows that enclose arcs depending on the setting of the BY parameter.

**Description** You can set up and execute the Panelize command. The converted arcs consist of a series of short line segments. The smoothness of the converted arcs depend on the setting of the CHORDANGLE parameter. The larger the chord angle the coarser the arc will be.

**Example** The following example converts all interpolated arcs that are in the current select group.

```

SEGMENTARCS
    CHORDANGLE    5
    BY             $$GROUPMODE
    GO
END

```

**SNOMAN**

**Purpose** Use this function to set up and execute the Snoman command.

**Menu command** Snoman (Tools menu)

**Syntax**

```
SNOMAN
    REPPFILE      string
    PAD2PAD       exp
    PAD2TRACE     exp
    OFFSET        exp
    MINPERCENT    exp
    MAXPERCENT    exp
    FROMLAYER     exp
    TOLAYER       exp
    DCODE         yesno
    GO            [lx, ly, ux, uy]
END
```

**Parameters**

**REPPFILE** A string containing the filename to which the report file will be written.

**PAD2PAD** An expression indicating the minimum pad-to-pad spacing.

**PAD2TRACE** Indicates the minimum pad-to-trace spacing.

**OFFSET** An expression indicating the host pad offset.

**MINPERCENT** An expression indicating the minimum percent of host pad size that a Snoman pad is allowed to be.

**MAXPERCENT** An expression indicating the maximum percent of host pad size that a Snoman pad is allowed to be.

**FROMLAYER** An expression that indicates the layer to perform the Snoman on. A value of zero indicates that all signal layers are to be processed.

**TOLAYER** An expression that indicates the output layer for the generated Snoman pads. A value of zero indicates the output layer is the same as the input layer.

**DCODE** An expression that indicates the D-Code filter used when determining which host pads to consider for generating Snoman pads. A value of zero indicates all D-Codes should be considered.

**GO** Executes the command using the current parameters with optional quad of expressions indicating a window to operate on.

**Description** This function allows the setup and execution of the Snoman command.

**Example** The following example executes the Snoman command.

```
SNOMAN
  REPPFILE "drc.rep"
  PAD2PAD 0.006
  PAD2TRACE 0.006
  OFFSET -0.005 # close hug
  MINPERCENT 40
  MAXPERCENT 80
  FROMLAYER 0
  TOLAYER 0
  DCODE 0
  WINDOWMODE $$NO
GO
END
```



**TEARDROPS**

Purpose	Use this command to create a maximum material condition at the point of trace entry into a pad, or at T junctions for traces.
Menu command	Teardrops (Tools menu)
Syntax	<pre>TEARDROPS     REPPFILE      string     LAYER         exp     DCODE         exp     WINDOWMODE    exp     DELETEXISTING yesno     HOSTPERCENT   exp     LENGTHMULTIPLE exp     WIDTHMULTIPLE exp     PAD2TRACE     exp     TRACE2TRACE   exp     GO END</pre>
Parameters	
REPPFILE	A string containing the filename to which the report file will be written.
LAYER	An expression indicating the layer to which GerbTool applies the teardrops.
DCODE	An expression that indicates the D-code filter used when determining which host pads to consider for generating teardrops. A value of zero indicates all D-codes should be considered.
WINDOWMODE	An expression that determines whether GerbTool applies teardrops to a viewing window or to the entire layer.
DELETEXISTING	Determines whether or not GerbTool deletes existing teardrops before performing the teardrop command.
HOSTPERCENT	An expression that determines the size of the teardrops as a percentage of the width of the host pad.
LENGTHMULTIPLE	An expression that specifies the “T” junction tail length. GerbTool calculates this value in multiples of the host trace diameter. The value you enter can be fractional.

WIDTHMULTIPLE	An expression that specifies the "T" junction tail width. GerbTool calculates this value in multiples of the host trace diameter. The value you enter can be fractional.
PAD2TRACE	An expression that specifies the pad to trace spacing parameters for teardrops.
TRACE2TRACE	An expression that specifies the trace to trace spacing parameters for teardrops.
Description	This function allows the setup and execution of the Teardrops command.
Example	The following example executes the Teardrops command.

TEARDROPS

```
REPFILE      "drc.rep"  
LAYER        $$ACTIVELAYER  
DCODE        0  
WINDOWMODE   $$NO  
DELETEXISTING $$YES  
HOSTPERCENT  40  
LENGTHMULTIPLE .5  
WIDTHMULTIPLE .5  
PAD2TRACE    0.006  
TRACE2TRACE  0.006  
GO
```

END

**SPREAD**

**Purpose** Use this command to set up and execute the Lyr Spread command.

**Menu command** Lyr Spread (Tools menu)

**Syntax**

```

SPREAD
    ROWS                exp
    COLS                exp
    TABSIZE             exp, exp
    LAYER               exp
    AUTOSPREAD          yesno
    SORTTYPE            exp
    GO
END

```

**Parameters**

**ROWS** An expression indicating the number of rows you want.

**COLS** An expression indicating the number of columns you want.

**TABSIZE** A pair of expressions indicating the X and Y spacing between images in the spread.

**LAYER** An expression that indicates the layer to output the spread pattern into.

**SORTTYPE** An expression where 0=ROW\_MAJOR and 1=COL\_MAJOR sorting.

**GO** Executes the command using the current parameters.

**Description** This function allows the setup and execution of the Lyr Spread command.

**Example** The following example executes the Lyr Spread command using autospread modes.

```

SPREAD
    SORTTYPE 0        #row major
    LAYER            $$ACTIVE_LAYER
    AUTOSPREAD       $$YES
    GO
END

```

## VENT

**Purpose** Use this function to set up and execute the Vent command.

**Menu command** Vent (Tools menu)

**Syntax**

```

VENT
    SPACING  exp, exp
    DCODE    exp
    GO       [lx, ly, ux, uy] . . .
END
    
```

### Parameters

**SPACING** A pair of expressions indicating the spacing between the flashes in the generated vent pattern.

**DCODE** An expression specifying the D-Code to be used in the generated vent pattern.

**GO** An optional quad of expressions describing a window to operate on.

**Description** You can set up and execute the Vent command.

**Example** The following example executes the Vent command.

```

VENT
    SPACING  0.100, 0.200
    DCODE    $$CURRENTDCODE
    GO       # go interactive
END
    
```

*User data entry functions***GETPOINT**

Purpose	Queries you for a point.
Menu command	None
Syntax	GETPOINT prompt, x, y
Parameters	
prompt	A string variable or string literal representing the prompt that will be shown the user when this command is run.
x	A numeric variable that returns the x coordinate of the point entered by the user.
y	A numeric variable that returns the y coordinate of the point entered by the user.
Description	You can display a message and enter a location. You can specify the coordinates in the same fashion as they would if one of the built-in GerbTool commands were querying them. The resulting coordinate is returned in the two provided variables.
Example	The following example queries you for a location and then adds a flash at that point.  <pre>GETPOINT "where do you want a flash?", \         \$x, \$y ADDFLASH    \$x, \$y</pre>

## GETSTRING

Purpose	To prompt a user to enter a text string.
Menu command	None
Syntax	GETSTRING prompt, destination
Parameters	
prompt	A string containing the prompt text.
destination	A string variable that will receive the user's response.
Description	You can use a macro to display a simple dialog box that contains a prompt message and a text field for the user to enter a response.
Example	The following example prompts you to enter your name. GETSTRING "Enter your name:", \$name

**GETWINDOW**

Purpose	Queries you for a window.
Menu command	None
Syntax	GETWINDOW prompt, x1, y1, x2, y2
Parameters	
prompt	A string variable or string literal representing the prompt that will be shown the user when this command is run.
x1	A numeric variable that returns the x coordinate of one corner of the user-specified window.
y1	A numeric variable that returns the y coordinate of one corner of the user-specified window.
x2	A numeric variable that returns the x coordinate of the second corner of the user-specified window.
y2	A numeric variable that returns the y coordinate of the second corner of the user-specified window.
Description	You can display a message and enter a window. You can specify the coordinates in the same fashion as they would if one of the built-in GerbTool commands were querying them. The resulting coordinates are returned in the four provided variables.
Example	The following example queries you for a window, and views that window.  <pre>GETWINDOW    "Enter window", \$x1, \$y1, \$x2, \$y2 VIEWWINDOW  \$x1, \$y1, \$x2, \$y2</pre>

## GETVALUE

Purpose	Queries you for a numerical value.
Menu command	None
Syntax	GETVALUE prompt, value
Parameters	
prompt	A string variable or string literal representing the prompt that will be shown the user when this command is run.
x	The numeric variable that returns the user-specified value.
Description	This command allows you to display a message box to the user, and allow them to enter a numerical value The resulting value is returned in the provided variable.
Example	The following example queries you for a layer number. GETVALUE "which layer", \$layer



**GETYESNO**

Purpose	Queries you for a Yes/No value.
Menu command	None
Syntax	GETYESNO prompt, yesno
Parameters	
prompt	A string variable or string literal representing the prompt that will be shown the user when this command is run.
yesno	A numeric variable that returns the user's choice. The possible values are \$\$YES and \$\$NO.
Description	You can display a message box and make a Yes/No decision. The resulting value is returned in the provided variable.
Example	<p>The following example queries you to see if it should terminate the macro.</p> <pre>GETYESNO "Quit? ", \$value IF \$value == \$\$YES     STOP END</pre>

**MESSAGEBOX**

Purpose	This function displays a dialog box with a title, message, and a choice of button labels.
Menu command	None
Syntax	<code>MESSAGEBOX title, message, button_style</code>
Parameters	
title	A string containing the dialog box title.
message	A string containing the dialog box message.
button_style	An expression indicating your choice of button labels.
Description	You can display a dialog box with both a title and a message. The body of the message can contain multiple lines of text by separating the lines with the two characters <code>\n</code> . The <code>button_style</code> parameter controls which buttons are displayed; 0=Okay, 1=Okay/Cancel, 2=Yes/No. The <code>\$\$STATUS</code> variable is set to 1 ( <code>\$\$YES</code> ) for Okay/Yes and 0 ( <code>\$\$NO</code> ) for Cancel/No.
Example	The following example prompts you for confirmation to continue.  <pre>MESSAGEBOX "MyMacro", \     "Found some errors\nContinue?", 2 IF \$\$STATUS == \$\$NO     STOP END</pre>

**SETPROMPT**

Purpose You can control the prompts shown while running a macro.

Menu command None

Syntax `SETPROMPT cmd_name, [cmd_prompt]`

## Parameters

`cmd_name` Any type of variable, literal or expression that will be displayed as the command name in the prompt area.

`cmd_prompt` Any type of variable, literal or expression that will be displayed is the command prompt.

Description This command updates the prompt area.

Example The following example sets the prompt bar to reflect the status of a variable used in a macro.

```
REPEAT $counter < $maxnets
    SETPROMPT "Processing net", $counter
    CALC $counter = $counter + 1
    .
    .
    .
END
```

## SHOWPROMPT

Purpose	Use this command to enable or disable the display of normal GerbTool prompts and messages in the prompt bar.
Menu command	None
Syntax	SHOWPROMPT <i>yesno</i>
Parameters	
<i>yesno</i>	An expression evaluating to zero for disable, otherwise enable.
Description	You can disable system prompts with this function. This allows some commands to run much faster, as there is less screen writing.
Example	The following example disables prompts, performs some time critical processing and finally turns prompts back on.  SHOWPROMPT \$\$NO ... <i>time critical processing</i> ... SHOWPROMPT \$\$YES

**PAUSE**

Purpose	This function causes the macro to pause.
Menu command	None
Syntax	<code>PAUSE [time] [cmd_name] [cmd_prompt]</code>
Parameters	
time	An optional expression specifying the amount of time to wait in tenths of a second.
cmd_name	An optional string specifying the command name portion of the prompt.
cmd_prompt	An optional string specifying the command prompt.
Description	This command causes the macro to pause, and presents message to the user to press any key to continue. Upon pressing any keyboard key, the macro continues. If the optional time value is included, the system waits for that period of time and then resumes, even if no key was pressed. A good use for this macro is when preparing demonstration macros showing how to perform some function.
Example	The following example pauses for 10 seconds or until the user presses a key.  <code>PAUSE 100, "My Cmd", "hit a key to cont"</code>

*Utilities and other functions***ABORTCHECKRATE**

Purpose	This function provides control on how often a macro checks for user aborts.
Menu command	None
Syntax	<code>ABORTCHECKRATE exp</code>
Parameters	
exp	An expression indicating the check rate.
Description	You can control how often a macro is interrupted to check for a user abort. The lower the checkrate number the more responsive a macro is to a user's request to abort and the slower the macro will run. Raising the checkrate has the opposite effect. The default checkrate is 200.
Example	<p>The following example sets the default abort check rate to a high value inside a database scan loop.</p> <pre># high speed, slow response ABORTCHECKRATE 5000 REPEAT \$\$STATUS ...database processing...     GETNEXTITEM END  # slow speed, fast response ABORTCHECKRATE 10</pre>

**CREATELAYER**

Purpose	Use this function to create and/or load a new layer.
Menu command	None
Syntax	CREATELAYER layer, filename, ap_filename
Parameters	
layer	An expression evaluating to a layer number.
filename	A string containing the filename of a Gerber file.
ap_filename	A string containing the filename an aperture list.
Description	This function creates a new layer in the currently loaded design. If the Gerber filename specified is found on disk, it will be loaded. Otherwise, an empty layer is created. If the layer parameter evaluates to zero, this function searches the layer table for the first unused layer. The actual layer created is returned in the \$\$STATUS variable. A return value less than or equal to zero indicates failure. This function operates similarly to normal design loading in that you do not need to specify an ap_filename if you intend to use a previously loaded aperture list.
Example	The following example creates a new unnamed layer in the first available empty layer and returns the new layer number in \$\$STATUS.  <pre>CREATELAYER 0, "", "" Calc \$tempLayer = \$\$STATUS</pre>

**SPLITPATH**

Purpose	Use this function to split a complete file path specification into its separate directory, filename, and extension components.
Menu command	None
Syntax	<code>SPLITPATH fullpath, dir, filename, ext</code>
Parameters	
fullpath	A string containing a complete file path specification.
dir	A string variable that will receive the directory portion of the fullpath string.
filename	A string variable that will receive the filename portion of the fullpath string.
ext	A string variable that will receive the extension portion of the fullpath string.
Description	This function separates the individual components of a complete file path specification.
Example	<p>In the following example, if <code>\$\$DSNNAME</code> contained the value <code>C:\PROJECTS\JOBS\GROMMIT.GTD</code>, then <code>\$filename</code> would end up containing the value <code>GROMMIT.ZIP</code>.</p> <pre>SPLITPATH \$\$DSNNAME, \$dir, \$file, \$ext STRWRITE \$filename, "%s.zip", \$file</pre>



**SYSCMD**

Purpose	Use this function to execute another program.
Menu command	None
Syntax	<code>SYSCMD command_line</code>
Parameters	
<code>command_line</code>	A string that will be passed to the host operating system for execution.
Description	<p>You can execute external programs while GerbTool waits. More than a simple convenience feature, this function allows you to have an external program perform a specialized task on a file created within a GerbTool macro. This modified file would then be read back into GerbTool, thereby extending GerbTool beyond its own limits.</p> <p>Note: This function CANNOT be used to execute a Windows executable even when issued from GerbTool.</p>
Example	<p>The following example shows how you can obtain a sorted copy data extracted from GerbTool.</p> <pre>STRWRITE \$cmd, "sort %s %s", \$infile, \$outfile SYSCMD \$cmd</pre>

## Viewing functions

### REDRAW

Purpose	Use this function to redraw the screen.
Menu command	Redraw (View menu)
Syntax	REDRAW
Parameters	None
Description	This command causes the screen to redraw. The scale or viewing positions are not changed.
Example	The following example changes the prompt to inform you that some calculations have been finished, and then performs a redraw so you can see the results.  <code>SETPROMPT "Finished", \$count REDRAW</code>

**VIEWALL**

Purpose	Use this function to view the extents of all visible layers.
Menu command	All (View menu)
Syntax	VIEWALL
Parameters	None
Description	This command changes the view scale so that all of the Gerber layers whose visibility is turned on are displayed on the screen.
Example	<p>The following example changes the prompt to inform you that some calculations have been finished, and then performs a VIEWALL so you can see the results.</p> <pre>SETPROMPT "Finished", \$count VIEWALL</pre>

## VIEWFILMBOX

Purpose	Use this function to view the film box and its contents.
Menu command	Filmbox (View menu)
Syntax	VIEWFILMBOX
Parameters	None
Description	This command changes the view scale so that the entire film box and its contents are displayed on the screen.
Example	<p>The following example changes the prompt to inform you that some calculations have been finished, and then performs a VIEWFILMBOX so you can see the results.</p> <pre>SETPROMPT "Finished", \$count VIEWFILMBOX</pre>

**VIEWPAN**

Purpose	Use this function to set the view window in GerbTool to a particular location.
Menu command	Pan (View menu)
Syntax	<code>VIEWPAN [x, y] ...</code>
Parameters	
x	An expression describing the x coordinate of the point to pan to.
y	An expression describing the y coordinate of the point to pan to.
Description	This function accepts a set of coordinates and changes the current view such that this coordinate is placed in the center of the screen. This function assumes that the point you want to pan to is currently on the screen. If this is not the case, you can use the <code>VIEWWINDOW</code> command to change the viewing location of the system.
Example	<p>This example takes a coordinate specified by \$x, \$y and pans to that location, changing the view window so that the entire screen is taken up by the 2 square inches surrounding the point.</p> <pre> CALC \$winLx = \$x - 1.0 CALC \$winLy = \$y - 1.0 CALC \$winUx = \$x + 1.0 CALC \$winUy = \$y + 1.0  VIEWWINDOW \$winLx, \$winLy, \$winUx, \$winUy VIEWPAN \$x, \$y </pre>

## VIEWPREVIOUS

Purpose	Use this function to cause the current view to be set to the state before it was last changed.
Menu command	Previous (View menu)
Syntax	VIEWPREVIOUS
Parameters	None
Description	This command recalls the last viewing window. This last viewing window is automatically saved after performing a view command such as View/Window.
Example	<p>The following example views a window whose size has been calculated previously, pauses 10 seconds to allow you to view the area, and then restores the view to its previous state.</p> <pre>VIEWWINDOW \$lx,\$lx, \$ux, \$uy PAUSE 100 VIEWPREVIOUS</pre>

**VIEWRECALL**

Purpose	Use this function to cause the current view to be set to that stored in the specified save locations.
Menu command	Recall (View menu)
Syntax	<code>VIEWSAVE location</code>
Parameters	
<code>location</code>	An expression giving the save location that contains the view. Valid locations are from 1 to 8.
Description	This command causes the current view to be changed to the value stored in one of the available save locations shown under the <i>View/Save</i> command, or set with the <code>VIEWSAVE</code> macro function.
Example	The following example saves the current view so that it can restore it after performing some calculations.  <code>VIEWSAVE 1</code> <code>...perform calculations here...</code> <code>VIEWRECALL 1</code>

## VIEWSAVE

Purpose	Use this function to cause the current view to be saved.
Menu command	Save (View menu)
Syntax	VIEWSAVE location
Parameters	
location	An expression giving the save location to place this view. Valid locations are from 1 to 8.
Description	This command causes the current view to be saved in one of the available save locations shown under the <i>View/Save</i> command.
Example	The following example saves the current view so that it can restore it after performing some calculations.  VIEWSAVE 1 ...perform calculations here... VIEWRECALL 1



**VIEWWINDOW**

Purpose	Use this function to set the view window in GerbTool to a particular location.
Menu command	Window (View menu)
Syntax	<code>VIEWWINDOW lx, ly, ux, uy</code>
Parameters	
lx	An expression describing the lower x coordinate of the view window.
ly	An expression describing the lower y coordinate of the view window.
ux	An expression describing the upper x coordinate of the view window.
uy	An expression describing the upper y coordinate of the view window.
Description	This function accepts four values representing the lower-left and upper-right coordinates of a rectangle. GerbTool then performs the equivalent of a Window command (View menu) on this area. Since there is no guarantee that the coordinates provided will coincide exactly with the view window, some of the design outside of the specified rectangle may be visible.
Example	<p>The following example gets the size of the film box and then views that area.</p> <pre>GETFILMBOX \$ux, \$uy VIEWWINDOW 0, 0, \$ux, \$uy</pre>

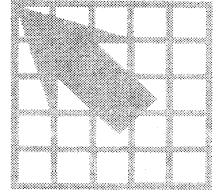
## ZOOMIN

Purpose	Use this function to set the view window in GerbTool to a particular location.
Menu command	Zoom In (View menu)
Syntax	ZOOMIN [x, y] . . .
Parameters	
x	An expression describing the x coordinate of the point to zoom in around.
y	An expression describing the y coordinate of the point to zoom in around.
Description	This function accepts a set of coordinates and changes the current view such that this coordinate is placed in the center of the screen and the magnification factor of the view is doubled.
Example	The following example gets the size of the film box and then zooms into its center.  GETFILMBOX    \$ux, \$uy CALC           \$ux = \$ux / 2 CALC           \$uy = \$uy / 2 ZOOMIN         \$ux, \$uy

**ZOOMOUT**

Purpose	Use this function to set the view window in GerbTool to a particular location.
Menu command	Zoom Out (View menu)
Syntax	ZOOMOUT [x, y] . . .
Parameters	
x	An expression describing the x coordinate of the point to zoom out around.
y	An expression describing the y coordinate of the point to zoom out around.
Description	This function accepts a set of coordinates and changes the current view such that this coordinate is placed in the center of the screen and the magnification factor of the view is decreased by half.
Example	<p>The following example gets the size of the film box and then zooms out from its center.</p> <pre> GETFILMBOX  \$ux, \$uy CALC        \$ux = \$ux / 2 CALC        \$uy = \$uy / 2 ZOOMOUT     \$ux, \$uy </pre>





## Aperture Conversion Rule files

In addition to providing the ability to convert most popular CAD and photoplotter aperture lists directly into the popular GerbTool format, you can create your own Aperture Conversion Rule (ACR) files for specialty, proprietary or otherwise unsupported aperture list formats.

### Definition of an ACR file

An Aperture Conversion Rule (ACR) file is an ASCII file that describes a particular aperture list format using conversion language statements. Using a text editor, you can create your own ACR file that describes the expected format of your aperture list. Once it is read in, GerbTool can convert your new aperture list format automatically, just as it converts the supported aperture list formats (see *Converting a CAD aperture list* in *Chapter 3: Quick start*).

### Creating an ACR file

An ACR file contains two types of statements. The first type describes the environment, such as the expected file extension, metric mode, number of header lines to skip, and so on. The second type is the actual rule statement, which is used to match incoming aperture list entries to corresponding GerbTool aperture shapes.

The following are descriptions of the environment-type of ACR statements and their expected parameters, if any.

### NAME

Syntax	<code>NAME converter_name</code>
Parameters	
<code>converter_name</code>	The name of the ACR file. Should be a single word.
Description	This statement will place the parameter in the header of the resulting aperture list.
Example	The following example sets the name of the converter to ALLEGRO.ACR. <code>NAME allegro.acr</code>

### VERSION

Syntax	<code>VERSION version_number</code>
Parameters	
<code>version_number</code>	The version number of the ACR file. The version number should be a single decimal number.
Description	This statement will place the parameter in the header of the resulting aperture list.
Example	The following example sets the version number of the converter to 6. <code>VERSION 6</code>

**HEADER**

Syntax	<code>HEADER lines_to_skip</code>
Parameters	
<code>lines_to_skip</code>	The number of lines to skip in the header of the aperture list.
Description	If this line is present, the number of lines specified will be skipped from the header of the aperture list file you are attempting to convert. This can be used to bypass information at the top of a file that you know does not contain any apertures.
Example	The following example instructs GerbTool to skip the first twenty lines of the aperture list.  <code>HEADER 20</code>

**SKIP**

Syntax	<code>SKIP skip_string</code>
Parameters	
<code>skip_string</code>	A text string to mark text to be skipped.
Description	If this line is present, all lines in the aperture list that start with the given character string will be ignored.
Example	The following example will allow GerbTool to skip over lines that begin with MOIRE.  <code>SKIP MOIRE</code>

## DEFAULT\_UNITS

Syntax	DEFAULT_UNITS mode
Parameters	
mode	One of \$\$INCH, \$\$MIL, or \$\$MM.
Description	If given, will cause the values read in to be interpreted as Inches, Mils, or Millimeters, depending on the value used.
Example	The following example sets the units mode to metric.  DEFAULT_UNITS \$\$MM

## CUSTOM

Syntax	CUSTOM yesno
Parameters	
yesno	Either \$\$YES or \$\$NO.
Description	If set to \$\$YES, GerbTool will attempt to create custom aperture names whenever possible. Otherwise a Diamond shape will be substituted. Note: GerbTool will not create the custom apertures themselves, only their names in the aperture list.
Example	The following example sets the creation of custom apertures to off.  CUSTOM \$\$NO



**EXTENSION**

Syntax	<code>EXTENSION extension</code>
Parameters	
<code>extension</code>	The default aperture list extension.
Description	The default extension of the aperture lists you will be converting with this rule file. If the value is entered here, you will not need to enter it when specifying the aperture list for conversion.
Example	The following example sets a default aperture list extension of MYA. <code>EXTENSION mya</code>

**DEBUG**

Syntax	<code>DEBUG mode</code>
Parameters	
<code>mode</code>	A value of 0, 1, or 2.
Description	Enables debugging information to be output into the aperture converter's log file. If zero is used, no debug information will be output. If 1 is used, GerbTool will output debug information while parsing the ACR file, and if the value is set to 2, debug information will be output while converting the aperture file itself. This function is for advanced users and should either not be included or be set to zero for normal converter operation.
Example	The following example sets the current debug mode to 2. <code>DEBUG 2</code>

## XTENSION

Syntax	<code>XTENSION dll_filename</code>
Parameters	
<code>dll_filename</code>	The name of a .DLL file that you supply.
Description	Causes the converter to look for the specified .DLL file to help in converting the aperture lists.
Example	The example specifies a user-supplied .DLL. <code>XTENSION myapfmt.dll</code>

## DCODE

Syntax	<code>DCODE mode</code>
Parameters	
<code>mode</code>	One of <code>\$\$ONLINE</code> , <code>\$\$SEQUENTIAL</code> , or <code>\$\$GERBER_ORDER</code> .
Description	Controls how D-Code values will be derived. If set to <code>\$\$ONLINE</code> (the default) the codes read on each line will be used. If <code>\$\$SEQUENTIAL</code> is used, lines that match the rules given will be assigned sequential numbers. Some aperture lists have their D-Codes arranged in a special non-sequential order used in certain Gerber photoplotters. Walcer will use this order if <code>\$\$GERBER_ORDER</code> is set.
Example	The example sets the D-Code mode to sequential. <code>DCODE \$\$SEQUENTIAL</code>

## #

Syntax	<code># any_text</code>
Parameters	
<code>any_text</code>	The body of a comment.
Description	This symbol leads comments in an ACR file.
Example	The example shows a typical comment. <code># Created By A. Designer</code>

The following is a description of each rule type of ACR statement and the expected parameters, if any.

### FORMAT\_shape

Syntax	FORMAT_shape rule
Parameters	
shape	The possible shapes are: ROUND, SQUARE, RECT, OBLONG, DONUT, DIAMOND, OCTAGON, THERMAL, THERM45, TARGET, and CUSTOM. Note that this parameter should be combined with the FORMAT_ statement to form a single word such as FORMAT_ROUND.
rule	A rule for matching apertures that are to be mapped to a GerbTool shape aperture.
Description	If the rule matches a line in the aperture list being converted, that line will be converted into a GerbTool shape aperture.
Example	The following example will match the line: JUNK D10 0.060 0.060 ROUND.  FORMAT_ROUND \$skip +D\$dcode \$xsize \$ysize ROUND

### FORMAT\_UNITS

Syntax	FORMAT_UNITS rule
Parameters	
rule	A rule for matching a line in the aperture list that specifies the format of the file.
Description	A line matching this is used to determine the format of the aperture list. This statement allows the aperture list itself to override a previous UNITS statement.
Example	The following example will match the line: FORMAT MM.  FORMAT_UNITS \$skip \$units

**FORMAT\_SPECIAL**

Syntax	<code>FORMAT_SPECIAL rule</code>
Parameters	
<code>rule</code>	A rule for matching lines for use by an XTENSION DLL.
Description	Does not produce a GerbTool D-Code line. It is used for special processing by an XTENSION-specified DLL.
Example	The following example will match the line: SQR D10 0.060 0.060.  <code>FORMAT_SPECIAL SQR +D\$dcode \$xsize \$ysize</code>

When constructing rules to match apertures, there are special keywords that you place in the rule that will cause GerbTool to assign the values contained in the fields to the corresponding GerbTool aperture list fields. These keywords are as follows:

<i>Keyword</i>	<i>Meaning</i>
<code>\$dcode</code>	Assigned to D-Code
<code>\$xsize</code>	Assigned to xsize
<code>\$od</code>	Assigned to xsize
<code>\$ysize</code>	Assigned to ysize
<code>\$id</code>	Assigned to ysize
<code>\$rot</code>	Assigned to rotation
<code>\$tool</code>	Assigned to tool num
<code>\$skip</code>	Skip this field
<code>\$custom</code>	Use this field to make a custom aperture
<code>\$units</code>	Used to determine the format of the aperture list

The following is a sample ACR file.

```
# Aperture converter for Mentor

NAME Mentor
VERSION 1.0
EXTENSION rpt

# handle swapped X/Y columns
XTENSION mentor.dll

DEBUG 0

CUSTOM $$NO

DEFAULT_UNITS $$INCH

HEADER 1

FORMAT_ROUND $skip +circle +$skip +$xsize +$ysize +$rot +false +false
+$dcode

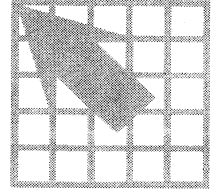
FORMAT_THERMAL $skip +circle +$skip +$xsize +$ysize +$rot +false +true
+$dcode

FORMAT_RECT $skip +rectangle +$skip +$xsize +$ysize +$rot +false +false
+$dcode

FORMAT_SPECIAL Position +Shape

# Mentor now has multiple formats
FORMAT_ROUND +$skip +$dcode +circle +$skip +$xsize +$ysize
FORMAT_THERMAL +$skip +$dcode +circle +$skip +power +$xsize +$ysize
FORMAT_RECT +$skip +$dcode +rectangle +$skip +$xsize +$ysize
FORMAT_SPECIAL Aperture Position
```






## Extended Gerber

GerbTool supports the extended Gerber data format, (sometimes referred to as 274X), developed by Gerber Systems, Inc. (GSI). This format provides for the inclusion of aperture data directly in the Gerber data files (embedded apertures), flexible aperture definitions and easy single file compositing.

### Embedded apertures

---

 **Note** While it not necessary to understand the syntax of extended Gerber to manipulate Extended Gerber files within GerbTool, several examples of this syntax are provided below. These examples are provided to acquaint you Extended Gerber only. See the instruction manuals provided with your photoplotter, or contact GSI directly, for more information on the extended Gerber syntax.

---

An extended Gerber file contains all aperture definitions necessary to plot the data, thereby eliminating the need for an external aperture list. An aperture is defined within an extended Gerber file with an AD command as follows:

```
%ADD<code><macro_name>,<parameter_list>*%
```

For example:

```
%ADD10C,0.06X0.020%
```

This example defines D10 as a simple 60-mils round flash using the GSI intrinsic aperture macro “C.”

GerbTool allows you to edit aperture definitions using the Edit AD button within the Edit dialog box. See *Chapter 7: Command reference* for more information.

## Aperture macros

Aperture macros describe the size and shape of special apertures. Using aperture macro primitives, you can design complex aperture shapes. Each primitive describes a basic shape such as a circle or a line. Each primitive also specifies its polarity (on/off) allowing data to be removed for such features as donuts or spokes in a thermal. Shown below are the different primitives available.

<i>Number</i>	<i>Type</i>	<i>Parameters</i>
1	Circle	on/off diameter xcenter ycenter
20	Line-Vector	on/off width xbeg ybeg xend yend rot
21	Line-Center	on/off width height xcenter ycenter rot
22	Line-Lower left	on/off width height xloc yloc rot
4	Outline	on/off count x y... rotation
5	Polygon	on/off sides xcenter ycenter diameter rot

### *Extended Gerber aperture macro primitives.*

Aperture macros are also programmable using *replaceable parameters*, which allow a macro to produce different results, depending on the aperture definition specified by the AD aperture definition command (explained in the preceding section). Replaceable parameters are indicated by a dollar sign (\$) followed by a numeric value. The numeric value indicates the parameter's position within the AD aperture definition. A typical donut macro and corresponding definitions are shown below.

```
%AMDONUT*
1,1,$1,0.0,0.0*
1,0,$2,0.0,0.0*
%

%ADD10DONUT,0.60X0.40%
%ADD20DONUT,0.08X0.70%
```

In the above example, D10 is defined as a 60-mils donut with a 40-mils hole, and D20 is defined as a 80-mils donut with a 70-mils hole. Note that both D10 and D20 refer to the same macro but have different sizes.

GerbTool allows you to edit aperture macros using the *Edit AM* button within the *Apertures/Edit* dialog box. See *Chapter 7: Command reference* for more information.



## Layer compositing

With extended Gerber, a single Gerber file defines a composite image of arbitrary complexity. Each "layer" of data within the Gerber file is prefixed with an appropriate polarity command. Ordering of the layers is critical as the data is processed sequentially. For assistance, check the example files provided and notice how each layer either adds or removes from the initial image.

GerbTool automatically creates separate layers for composite layers when reading an extended Gerber file and conversely creates a single file for all layers that form a composite when writing out data.

## Viewing composites

Composite layers can be displayed with the `v` shortcut key. This nested command toggles composite viewing on or off. When **enabled**, GerbTool displays composite layers as they are plotted. When **disabled**, GerbTool displays composite layers as if all layers were dark (positive). You can also control composite viewing using the Edit dialog box.

## Converting from RS-274-D to extended Gerber format

In order to convert a set of standard Gerber 274D files into a single extended Gerber composite file, load the 274D files as you normally do and then perform the following steps using the Edit command (Layers menu):

- Set the Layer Name field of each layer to a meaningful name.



**Tip** Setting the Layer Name field to the original filename of the same layer will label the extended Gerber "layers" in a fashion that will be familiar to the user.

- Decide on the filename you want to use for the new extended Gerber file and rename all of the Gerber filenames to this new name. It is important that each extended Gerber "layer" have the same filename.
- Set the Layer Type for each of these layers to Composite.
- Assign a polarity and a common number to the Key field for each of the extended Gerber "layers." For example, D1 for "Dark composite number 1" or C1 for "Clear composite number 1." A polarity of Dark means that the layer is to be displayed in the style a normal Gerber file is displayed. Clear tells GerbTool to display the layer using the current background color. This erases, or "clears," areas from an image that were previously drawn by a "dark" layer. Set negative layers to clear.

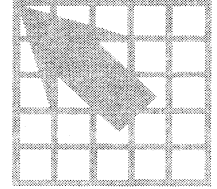


**Note** The common number portion of the Key field allows GerbTool to load multiple extended Gerber composite files at the same time. Each set of layers within an extended Gerber file should have a common number assign to the Key field.

---

- Click on the *Edit* button within the File Format group box. Change the Dialect field to extended Gerber (274X).
- Save the composite file using the Save command (File menu). All the layers will all be written into a single extended Gerber file with the name that you specified, along with an embedded aperture list.

To load this new extended Gerber composite file into another design, enter its filename into the Filename field as you would with any other Gerber file, making sure the file format has been set to extended Gerber (274X). There is no need to load in an aperture list as it is included in the file.




## Using custom apertures

Using GerbTool, you can create custom apertures. A custom aperture is nothing more than a Gerber file, and can therefore be of virtually any size or shape. This chapter details the steps for creating a custom aperture.

### Create a custom aperture

- Using the Format command (Files menu), set the Gerber format to Inch, absolute, 2.3 and no zero suppression.
- Select the Load command (Files menu).
- Enter the appropriate design filename.


---

 **Tip** Use one design file for all of your custom apertures.

---

- In the Layers dialog box, enter a descriptive name in the Filename field, such as FIDUCIAL.CUS (the .CUS extension is mandatory).
- Enter the filename of the aperture list that you will be using for this custom aperture in the Aperture List field.
- Choose the OK button. GerbTool will inform you that the specified Gerber file doesn't exist. Respond affirmatively to create the new layer.
- At this point you can create your custom aperture using any of the apertures defined in the aperture list assigned to the new layer.


---

 **Note** Before you save your custom aperture, ensure that the origin is where you want it. You can use the Origin command (Edit menu) to relocate the origin.


---

To use the new custom aperture, enter its filename (less extension) in the Shape field of an aperture list using the Edit command (Apertures menu).

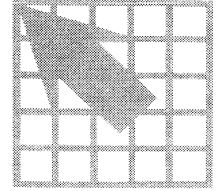
---

 **Caution** The aperture list used while designing your custom aperture must be specified in the Custom Ap List field within the Defaults dialog box. Use the Defaults command (Options menu) to change this field if necessary. A aperture list used for custom apertures should not itself contain any custom apertures.

---

 **Tip** It is recommended that you set aside one aperture list dedicated to all your custom apertures.

---



## Working with text fonts

GerbTool uses a font file containing a list of X-Y coordinate pairs that constitute the “strokes” required to display each character inserted by the Text command (Edit menu). You can have more than one font file but GerbTool will always read the STROKE.FNT file at startup. To use a different font file, rename STROKE.FNT to some other name, then rename your font file to STROKE.FNT. GerbTool allows you to edit existing fonts and create new fonts that are used for text insertion. This chapter details the steps for editing fonts.

### Editing a font

Before you edit a font you must convert it into individual Gerber files for each character. To do this, from the system prompt change to the GerbTool fonts directory and type the following command, then touch the ENTER key:

```
f2g ../stroke.fnt
```

This will create an individual Gerber file for each character in the font file. You can now start GerbTool and load one of the provided design files UPCASE.GTD, LWCASE.GTD, NUMBERS.GTD, PUNC1.GTD, or PUNC2.GTD, which cover uppercase, lowercase, numbers, and punctuation characters respectively. The Film Box is set to a 7-mils square, which each character must remain within. You can draw any shape you want as long as you stay in or on the film box and you don't try to add flashes.




---

**Note** It is important that the file format of the individual Gerber files for each character remain at Inch, absolute, 2.3, and no zero suppression.

---

Once you have finished editing the characters, you can use the following command at the system prompt to create a new font file, then touch the ENTER key.

```
g2f newfile.fnt
```

In the above example a new font file would be created with a filename of NEWFILE.FNT. Note that this program does not purge the individual Gerber character files. You may do this manually if you want. Remember that GerbTool will not recognize your new font file unless it is named STROKE.FNT and is in the GerbTool program directory.

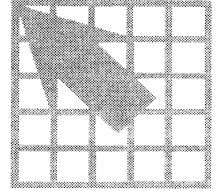
## Creating a new font

To create a completely new font you can follow the steps detailed in *Editing a font* above, but skip the font file to Gerber file conversion step.



**Note** It is usually easier (and faster) to modify an existing font than to create one from scratch.

---



## Command ID values

The tables in this appendix contain the command ID values associated with each GerbTool command. You can use these values to program your mouse and function keys.

<i>Command</i>	<i>ID</i>
Compact (Apertures menu)	AO
Convert (Apertures menu)	AV
Edit (Apertures menu)	AE
Load (Apertures menu)	AL
Merge (Apertures menu)	AM
Report (Apertures menu)	AR
Save (Apertures menu)	AS
Unload (Apertures menu)	AU
Arc 3 Pt (Add item, Edit menu)	EAA3
Arc Ctr (Add item, Edit menu)	EAAC
Circle (Add item, Edit menu)	EAC
Draw (Add item, Edit menu)	EAD
Flash (Add item, Edit menu)	EAF
Polygon (Add item, Edit menu)	EAP
Rectangle (Add item, Edit menu)	EAR
Text (Add item, Edit menu)	EAT
Vertex (Add item, Edit menu)	EAV
Align (Edit menu)	EA
Clip (Edit menu)	EK

*Command ID values (page 1 of 5).*

<i>Command</i>	<i>ID</i>
Copy (Edit menu)	EC
Expand (Dcode item, Edit menu)	EDE
Polarity (Dcode item, Edit menu)	EDP
Scale (Dcode item, Edit menu)	EDS
Transcode (Dcode item, Edit menu)	EDT
Erase (Edit menu)	EE
Item (Edit menu)	ET
Mirror (Edit menu)	EI
Move (Edit menu)	EM
Origin (Edit menu)	EO
Purge (Edit menu)	EP
Rotate (Edit menu)	ER
Add (Select item, Edit menu)	ESA
Invert (Select item, Edit menu)	EPI
New (Select item, Edit menu)	ESN
Off (Select item, Edit menu)	ESO
Remove (Select item, Edit menu)	ESR
Undo (Edit menu)	EU
Chgdir (File menu)	FD
Close (File menu)	FC
Exit (File menu)	FQ
BARCO DPF (Export item, File menu)	FEB
IPC-D-350 (Export item, File menu)	FE350
IPC-D-356 (Export item, File menu)	FE356
Drill (Format item, File menu)	FFD
Gerber (Format item, File menu)	FFG
Load (Format item, File menu)	FL
BARCO DPF (Import item, File menu)	FIB

---

*Command ID values (page 2 of 5).*



<i>Command</i>	<i>ID</i>
Drill (Import item, File menu)	FIN
HPGL (Import item, File menu)	FIH
IPC-D-356 (Import item, File menu)	FI356
Design (Merge item, File menu)	FMD
Gerber (Merge item, File menu)	FMG
Auto (New item, File menu)	FNA
Manual (New item, File menu)	FNM
Open (File menu)	FO
HPGL (Plot item, File menu)	FPH
PostScript (Plot item, File menu)	FPP
Print (File menu)	FP
Save (File menu)	FS
Colors (Layers menu)	LC
Edit (Layers menu)	LE
Arcs 360 (Options menu)	OA
Bg Color (Options menu)	OB
Defaults (Options menu)	OD
Filmbox (Options menu)	OF
Grid (Options menu)	OG
KeyCmds (Options menu)	OK
Metric (Options menu)	OM
Ortho (Options menu)	OR
Overlay (Options menu)	OO
Save (Options menu)	OV
Show Errs (Options menu)	OE
Sketch (Options menu)	OS
Undo (Options menu)	OU
Copper (Query menu)	QC

*Command ID values (page 3 of 5).*

<i>Command</i>	<i>ID</i>
Extents (Query menu)	QE
Dcode (Highlight item, Query menu)	QHD
Net (Highlight item, Query menu)	QHN
Off (Highlight item, Query menu)	QHO
Item (Query menu)	QI
Edge to Edge (Measure item, Query menu)	QME
Point to Point (Measure item, Query menu)	QMP
Circles (Convert item, Tools menu)	TCA
Pads (Convert item, Tools menu)	TCP
DRC (Tools menu)	TD
Fix SS (Tools menu)	TF
Lyr Spread (Tools menu)	TL
Load (Macro item, Tools menu)	TML
Run (Macro item, Tools menu)	TMR
Drawing (NC Drill item, Tools menu)	TNDD
Write (NC Drill item, Tools menu)	TNDW
Generate (Netlist item, Tools menu)	TNLG
Write (Netlist item, Tools menu)	TNLW
Isolated (Pad Removal item, Tools menu)	TPI
Stacked (Pad Removal item, Tools menu)	TPS
Panelize (Tools menu)	TP
Snoman (Tools menu)	TS
Vent (Tools menu)	TV
All (View menu)	VA
Errors (View menu)	VE
Filmbox (View menu)	VF
Pan (View menu)	VP
Previous (View menu)	VV

---

*Command ID values (page 4 of 5).*

---

<i>Command</i>	<i>ID</i>
Recall (View menu)	VC
Redraw (View menu)	VR
Save (View menu)	VS
Window (View menu)	VW
ZoomIn (View menu)	VI
ZoomOut (View menu)	VO

---

*Command ID values (page 5 of 5).*

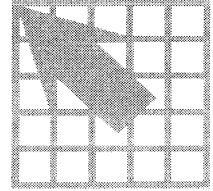
---

<i>Command</i>	<i>ID</i>
All (View menu)	VA
Film Box (View menu)	VF
Pan (View menu)	VP
Previous (View menu)	VV
Redraw (View menu)	VR
Window (View menu)	VW
Zoom In (View menu)	VI
Zoom Out (View menu)	VO

---

*Command ID values assignable to mouse buttons.*





## Aperture list file format

This appendix describes the format of a GerbTool aperture list and provides an example of an aperture list.

Aperture lists are stored as simple ASCII files. There are nine fields in each line of the file. Each line defines one D-code. The fields consist of the following:

<i>Field</i>	<i>Possible values</i>
D-Code	10 - 4095
Shape	Round, Square, Rectangle, Oblong, Donut, Diamond, Octagon, Thermal, Therm45, Target, Complex, or a filename prefixed by a “%”
Width	0.0 - 9.9999
Height	0.0 - 9.9999 When referring to Donuts or Thermals, this field represents the diameter of the inner hole. When referring to Targets, it refers to the diameter of the inner ring of the Target.
Type	SM (surface-mount) or TH (through-hole)
Tool	0 - 999 Specifies the Tool used to drill this D-Code.
Tool Size	0.0 - 9.9999 Specifies the size of the above Tool number.
Legend	10 - 4095 Specifies the D-Code to use in place of this D-Code when creating a Drill Drawing using the Drawing command from the Tools>Drill menu.
R90	10 - 4095 Specifies the D-Code to substitute for this D-Code when rotating 90 or 270 degrees. This field exists only for compatibility with older versions of GerbTool, as newer versions perform the D-Code substitutions automatically.

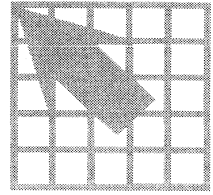
*Aperture list field definitions.*

All fields are separated by white space. Lines that begin with a “#” are treated as comments. Although the author and data comments are not required, they are generally included as an aid for other users. The header of a GerbTool aperture list may contain a format line proceeded by a “%.” This line contains either IMPERIAL or METRIC followed by a version number. If IMPERIAL is specified, all sizes are in inches. If METRIC is specified, they are in millimeters. If no format line is provided, IMPERIAL is assumed. The version number is for documentation purposes only. An excerpt from an aperture list showing the required format follows.

```
# Format, Version
%IMPERIAL, V3.0
#
# Author: GerbTool V1.0 (c) 1992 WISE Software
Solutions, Inc.
# Date:   Wed Oct  7 13:28:46 1992
#
#      Shape      Width  Height Type Tool   Size Legend R90
#
D12 Round      0.0100 0.0100 TH   0     0.0  0     0
D21 Square     0.0200 0.0200 TH   2     0.0  0     0
D22 Rectangle 0.0220 0.0180 SM   3     0.0 85    0
D23 Oblong     0.0220 0.0180 TH   3     0.0  0     0
D24 Diamond    0.0240 0.0240 TH   4     0.0  0     0
D25 Target     0.1800 0.1600 TH   0     0.0  0     0
D26 %FIDUCIAL 0.0000 0.0000 TH   0     0.0  0     0
D70 Octagon    0.0240 0.0240 TH   5     0.0  0     0
D71 Thermal    0.0240 0.0200 TH   0     0.0  0     0
```

*Sample aperture list file.*

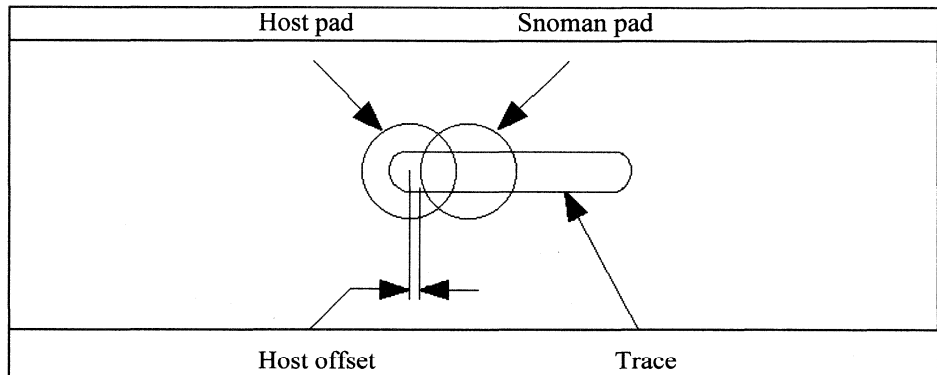
In the above example, D26 is specified as a Custom aperture with a filename of FIDUCIAL.CUS. The “%” is required, to notify GerbTool that what follows is a custom aperture filename.



## Snoman concepts

Snoman creates a *maximum material condition* at the point where a trace segment enters a pad, thereby eliminating the possibility of pad/trace separation (breakout). This is accomplished by examining a Gerber file (layer) and outputting pad flashes at the correct locations, and of the correct size, to provide the most material where a trace enters a pad. Automatic adjustments are made to the size and location of the generated Snoman pads to eliminate design rule spacing violations.

The following illustration shows the original pad and trace, as well as the resultant Snoman pad.

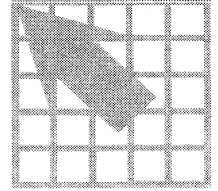


*Snoman concept.*

The distance maintained between the host pad center and the edge of the generated Snoman pad (see *Host offset* in illustration above) is adjustable. Negative values allow the Snoman pads to closely hug the host pads.







## A

**absolute mode** When all X-Y coordinates are referenced to a common origin (0,0).

**active layer** The layer that all items added to the database will go to.

**aperture list** A list of Gerber D-Code definitions.

## B-C

**block size** The size of a coordinate value in characters. Also known as *m.n* format.

**breakout** Pad and trace separation during fabrication.

## D-H

**design file** A file containing information about the layer structure of a single PCB design. This file also stores various information about the GerbTool operating environment.

**DRC** Acronym for Design Rules Check.

## I-M

**incremental mode** When each X-Y coordinate is a displacement from the previous coordinate.

**isolated pads** Pads that do not have a trace connected to them.

## N-O

**NC drill** Refers to files produced to drive Numerically Controlled drilling machines.

**netlist** A file containing groups of pad X-Y locations that are connected by traces.

## P-U

**pad removal** The act of removing isolated or stacked pads.

**pan** Moving the location of the viewing window without changing its size.

**panelize** Placing multiple copies of a PCB on one piece of film. The multiple copies are then manufactured on a single panel, thereby reducing manufacturing costs.

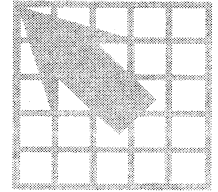
**point** A X-Y location within the drawing area.

# V

**virtual memory** A combination of hardware and software that allows an application to address all memory that the CPU is capable of addressing, even when there is less actual memory. The virtual memory manager swaps data back and forth to the disk and remaps memory addresses to provide applications with virtually unlimited memory. Available disk space becomes the limiting factor.

# W-Z

**wildcard specification** A method of specifying more than one filename at a time. Use the asterisk (\*) character to match any character or group of characters. Use a question mark (?) to match a single character. For example: \*.GTD represents all files that end with the .GTD extension.



## **A**

- active layer, *14*
- add
  - arc, *56*
  - arc 3 pt, *56*
  - circle, *55*
  - draw, *55*
  - flash, *55*
  - polygon, *56*
  - rectangle, *55*
  - text, *58*
  - vertex, *55*
- aligning layers, *21, 51*
- annular ring, *89*
- aperture conversion rules
  - #, *278*
  - CUSTOM, *276*
  - DCODE, *278*
  - DEBUG, *277*
  - DEFAULT\_UNITS, *276*
  - EXTENSION, *277*
  - FORMAT\_shape, *279*
  - FORMAT\_SPECIAL, *280*
  - FORMAT\_UNITS, *279*
  - HEADER, *275*
  - NAME, *274*
  - SKIP, *275*
  - VERSION, *274*
  - XTENSION, *278*
- aperture list
  - compacting, *68*
  - converting, *6, 68*
  - create, *68*
  - editing, *64*
  - embedded, *283*

- files, *15*
- format, *297*
- load, *68*
- merging, *68*
- saving, *68*
- unloading, *68*
- use report, *67*
- apertures
  - custom, *65*
  - definition, *66*
  - macro, *65, 66, 284*
  - maximum size, *75*
- arcs
  - chord angle, *75, 242*
  - conversion, *100*
  - interpolated, *13, 73*

## **B**

- background color, *77*
- BARCO DPF
  - exporting, *36*
  - importing, *34*
- birdseye view, *11*
- breakout, *299*

## **C**

- calculating
  - copper, *72*
  - data extents, *72*
- chord angle, *75*
- color list file, *3*
- command ID, *291*
- compacting
  - aperture list, *68*

## Index

---

- composite layers, 23
  - 274X, 61
- composites, 274X, 285
- composites, viewing 274X, 53, 285
- configuration, 3
- configure
  - aperture converters, 78
  - function keys, 77
  - macro files, 80
  - mouse, 77
  - paths, 79
  - user menu, 81
- conversion
  - aperture lists, 15
  - arcs, 100
  - drawn pads, 99
    - RS274D to 274X, 285
- converting aperture list, 68
- coordinate display, 14
- copper calculation, 72
- copying, 47
- creating a soldermask layer, 26
- creating NC Drill files, 22
- crosshair
  - size, 76
- crosshair cursor, 14

## D

- data extents calculation, 72
- D-code
  - expand, 50
  - highlight, 72
  - polarity, 51
  - scale, 50
  - transcode, 26, 50
- deleting, 48
- deleting with clipping, 48
- design
  - automatic, 8, 29
  - closing, 30
  - creating, 9, 29
  - merging, 24, 33
  - opening, 9, 30
  - save, 30
  - save all, 30

- save as, 30
- destination layer
  - copying, 47
  - moving, 48
- drawing
  - interrupting, 18, 19
- drawing area, 11, 14
- drawn pads, 25
  - conversion, 99
- DRC, 88
  - annular ring, 89
  - keepouts, 90
  - missing drill, 89
  - stubs, 89
  - view errors, 90
  - well behaved, 89

## E

- edit item, 47
- editing
  - aperture list, 64
- editing layers, 59
- ending a command, 18
- exiting GerbTool, 10, 42
- exporting
  - BARCO DPF, 36
  - HPGL, 37
  - IPC-D-350, 36
  - IPC-D-356, 36
  - PostScript, 39

## F

- file format
  - critical, 35
  - detection, 62
  - global, 31
  - local, 31, 62
  - metric, 32
- film box, 15
  - change, 76
- fonts
  - creating, 290
  - editing, 289
- function key assignments, 16

**G**

Gerber dialect  
 274X, 31  
 EIE, 31  
 FIRE9xxx, 31  
 Gerber files, 15  
 grid  
 change size, 76  
 display, 13, 53  
 snap, 13  
 group  
 selecting, 45  
 GTMAPDIR, 79

**H**

highlight  
 clear, 53  
 colors, 76  
 D-codes, 72  
 nets, 70  
 selections, 54  
 toggle, 53  
 userdata, 71

**I**

importing  
 BARCO DPF, 34  
 IPC-D-356 netlist, 35  
 NC Drill file, 35  
 interactive plot positioning, 38  
 IPC-D-350  
 exporting, 36  
 IPC-D-356  
 exporting, 36  
 IPC-D-356 netlist  
 importing, 35  
 isolated pad removal, 97  
 item  
 editing, 47  
 item information  
 displaying, 69

**J-K**

join, 49  
 joining lines, 49

**L**

layers  
 active, 14  
 aligning, 51  
 alignment, 21  
 color and visibility, 63  
 color, floating, 11  
 composite, 23  
 create, 63  
 editing, 59  
 maximum, 75  
 path, 60  
 rearranging, 61  
 ref mode, 75  
 save, 30  
 save all, 30  
 save as, 30  
 spread, 101  
 visibility, 61  
 lines  
 chamfer, 49  
 fillet, 49  
 joining, 49  
 loading macros, 106

**M**

macro  
 coordinate lists, 105  
 edit, 82  
 function keys, 77  
 language reference, 107  
 loading, 82, 106  
 running, 82, 106  
 system variables, 104  
 macros  
 ABORTCHECKRATE, 258  
 ABS, 198  
 ACTIVELAYER, 165  
 ADD3PTARC, 108  
 ADDARC, 109

ADDCIRCLE, 110  
ADDDRAW, 111  
ADDFILL, 112  
ADDFLASH, 113  
ADDPOUR, 114  
ADDTEXT, 115  
ADDVERTEX, 116  
ALIGNLAYERS, 138  
APREPORT, 117  
ARRAY, 199  
ASIN, 200  
ATAN, 201  
BKCOLOR, 166  
CALC, 202  
CALLMACRO, 120  
CEIL, 203  
CLIP, 139  
COPY, 141  
COPYITEM, 125  
CREATELAYER, 259  
CSIN, 204  
CURRENTDCODE, 167  
DCEXPAND, 143  
DCODESCALE, 144  
DEBUG, 121  
DELETEITEM, 126  
DRAWNPADS, 231  
DRC, 233  
DRILL, 235  
ERASE, 146  
EXTENSIONS, 168  
FILECLOSE, 191  
FILEOPEN, 192  
FILEREAD, 193  
FILESPATH, 169  
FILEWRITE, 194  
FILMBOX, 170  
FIXSS, 237  
FLAGS, 171  
FLOOR, 205  
FORMAT, 172  
GETAPINFO, 118  
GETEXTENTS, 127  
GETFILMBOX, 128  
GETFIRSTITEM, 129  
GETLAYER, 131  
GETNEXTITEM, 133  
GETPOINT, 249  
GETSTRING, 250  
GETUSERDATA, 134  
GETVALUE, 252  
GETVIEWEXTENTS, 135  
GETWINDOW, 251  
GETYESNO, 253  
GRIDSIZE, 174  
GRIDSNAP, 175  
GRIDVISIBLE, 176  
HILICOLORS, 177  
HILIDCODE, 216  
HILITEM, 217  
IF, 122  
LAYERN, 178  
MAPPATH, 180  
MEASUREE2E, 218  
MERGEDRILL, 195  
MERGEGERBER, 196  
MERGEHPGL, 197  
MESSAGEBOX, 254  
MIRROR, 148  
MOVE, 150  
MOVEITEM, 136  
NETID, 181  
NETLIST, 238  
OFFSETS, 182  
ORIGIN, 152  
OVERLAYMODE, 183  
PADREMOVAL, 239  
PANELIZE, 240  
PAUSE, 257  
PLOTHPGL, 212  
PLOTPTS, 214  
POUR, 153  
PREVIOUSVIEW, 184  
PURGE, 155  
PUTAPINFO, 119  
PUTUSERDATA, 137  
REDRAW, 262  
REPEAT, 123  
ROTATE, 156  
ROUND, 206  
SCALE, 185  
SEGMENTARCS, 242

- SELECTCRITERIA, 158
  - SELECTGROUP, 159
  - SET, 207
  - SETGLOBAL, 208
  - SETPROMPT, 255
  - SHOWERRORS, 186
  - SHOWPROMPT, 256
  - SIN, 209
  - SKETCHMODE, 187
  - SNOMAN, 243
  - SPLITPATH, 260
  - SPREAD, 247
  - SQRT, 210
  - STOP, 124
  - STRARRAY, 219
  - STRCAT, 220
  - STRCMP, 221
  - STRCPY, 222
  - STRLEN, 223
  - STRLOC, 224
  - STRREAD, 225
  - STRSET, 226
  - STRSETGLOBAL, 227
  - STRSUB, 228
  - STRTOK, 229
  - STRWRITE, 230
  - SYSCMD, 261
  - TAN, 211
  - TEARDROPS, 245
  - TEXT, 161
  - TRANSCODE, 163
  - UNDO, 188
  - VENT, 248
  - VIEWALL, 263
  - VIEWCOMPOSITES, 189
  - VIEWFILMBOX, 264
  - VIEWMETRIC, 190
  - VIEWPAN, 265
  - VIEWPREVIOUS, 266
  - VIEWRECALL, 267
  - VIEWSAVE, 268
  - VIEWWINDOW, 269
  - ZOOMIN, 270
  - ZOOMOUT, 271
- measuring
- edge-to-edge, 71
  - point-to-point, 71
- merging
- aperture list, 68
  - design file, 24, 33
  - Gerber file, 33
  - HPGL file, 34
- metric mode, 13, 73
- mirroring, 49
- mouse button assignments, 16
- moving, 48
- ## N
- NC Drill, 15
- creating, 22, 98
  - drawing, 97
  - importing, 35
  - tools, 65
- netlist
- generate, 94
  - highlight, 70
  - saving, 32, 95
  - well behaved, 94
- ## O
- offsets
- applying, 50
- operating environment, 15
- origin, 51
- ortho
- line angle, 75
- orthogonal mode, 13, 73
- overlay mode, 12
- overview, 5
- ## P
- pad removal
- isolated, 97
  - stacked, 97
- page setup, 41
- panelize, 23, 83
- virtual, 87, 98
- panning, 52
- plotting
- batch mode, 38, 40, 41

## Index

---

- borders, *37, 39, 212, 214*
- composite layers, *39*
- polygon
  - filling, *56*
  - pouring, *56*
- PostScript plotting, *39*
- print preview, *42*
- printer setup, *42*
- printing, *42*
  - borders, *41*
- programming
  - function keys, *20*
  - mouse buttons, *20*
- purging, *51*
- Q**
- query
  - nets, *70*
  - userdata, *71*
- querying
  - database information, *69*
- R**
- redraw
  - minimize, *77*
- rotating, *49*
- rule violation
  - displaying, *12, 54*
- running macros, *106*
- S**
- save, *30*
- save all, *30*
- save as, *30*
- scale
  - database, *50*
  - D-codes, *50*
- selecting groups, *45*
- selection criteria, *43*
- settings
  - current, *12*
- shortcut keys, *17, 19*
- silkscreen
  - automatic cleanup, *25*
  - auto fixup, *96*
  - sketch mode, *12*
  - Snoman, *27, 91*
  - soldermask layer
    - creating, *26*
  - split screen, *11*
  - starting GerbTool, *5*
  - status bar, *15*
  - step and repeat, *98*
  - surface-mount pads, *65*
- T**
- teardrops, *92*
- text, *58*
- thieving, *101*
- through-hole pads, *65*
- tool tips, *11*
- toolbar, *11*
- transcode, *26*
- U**
- undo, *19, 44*
  - toggle, *75*
- unused pad removal, *97*
- userdata
  - editing, *47*
  - highlight, *71*
- V**
- vent
  - automatic, *86*
  - manual, *101*
- view
  - all, *52*
  - birdseye, *14*
  - composites, *12, 53*
  - errors, *12, 90*
  - film box, *52*
  - grid, *53*
  - new window, *52*
  - overlay mode, *53*
  - panning, *52*
  - previous, *54*
  - recall, *54*



---

- redrawing, 53
- saving, 54
- sketch mode, 53
- split screen, 14, 54
- toolbars, 54
- virtual panel, 53
- zoom in, 52
- zoom out, 52

## W-Y

- well behaved
  - DRC, 89
  - netlist, 94
- window, 11
- wizard
  - new design, 29, 74

## Z

- zero suppression, 32
- zooming in, 52
- zooming out, 52