EAGLE

EASILY APPLICABLE GRAPHICAL LAYOUT EDITOR

Manual

Version 4.1

2nd Edition

How to reach us

Office Hours are:

Mon - Thu:	9 am to 5 pm EST
Fri:	9 am to 4 pm EST

Phone:	+1 561 274-8355
Fax:	+1 561 274-8218
E-mail :	Info@cadsoftusa.com
Web:	http://www.cadsoftusa.com

CadSoft Computer, Inc.
801 South Federal Highway, Suite 201
Delray Beach, Florida 33483-5185
U.S.A.

EAGLE LICENSE AGREEMENT

This is a legal agreement between you, the end user, and CadSoft Computer, Inc., which markets software products under the trademark EAGLE. CadSoft Computer, Inc. shall be referred to in this Agreement as CadSoft. If you do not agree to the terms of this Agreement, promptly return the diskette package and accompanying items (including written materials and containers) to the place you obtained them for a full refund. USE OF THIS PRODUCT CONSTITUTES YOUR ACCEPTANCE OF THESE TERMS AND CONDITIONS AND YOUR AGREEMENT TO ABIDE BY THEM.

Grant of License

CadSoft grants to you the right to use one copy of the accompanying EAGLE software program and any and all updates that you may receive (the Software) on a single computer or workstation. You may, however, install the Software on more than one computer or on a file server provided you do not operate the Software on more than one computer or workstation at a time.

Copyright

The Software is owned by CadSoft and is protected by United States copyright laws and international treaty provisions. Therefore, you must treat the Software like any other copyrighted material (e.g., a book or musical recording). You may not copy the written materials accompanying the Software.

Other Restrictions

You may not rent or lease the Software, but you may transfer your stand-alone copy of the Software and accompanying written materials on a permanent basis provided you retain no copies and the recipient agrees to the terms of this Agreement. Any such transfer must include all updates and prior versions of the Software and accompanying written materials, and notice must be given by you to CadSoft that such transfer has taken place. You may not reverse engineer, decompile, disassemble, or create derivative works based on the Software for any purpose other than creating an adaptation to the Software as an essential step in its utilization for your own use. You acknowledge Cadsoft’s claim that the Software embodies valuable trade secrets proprietary to CadSoft; you may not disclose any information regarding the internal operations of the Software to others.

LIMITED WARRANTY

CadSoft warrants the accompanying Software and documentation to be free of defects in materials and workmanship for a period of ninety (90) days from the purchase date. The entire and exclusive liability and remedy for breach of this Limited Warranty shall be, at Cadsoft's option, either (a) return of the price paid or (b) replacement of defective Software and/or documentation provided the Software and/or documentation is returned to CadSoft with a copy of your receipt. Cadsoft's liability shall not include or extend to any claim for or right to recover any other damages, including but not limited to, loss of profit, data or use of the Software, or special, incidental or consequential damages or other similar claims, even if CadSoft has been specifically advised of the possibility of such damages. In no event will Cadsoft's liability for any damages to you or any other person ever exceed the lower of suggested list price or actual price paid for the license to use the Software, regardless of any form of the claim.

TO THE EXTENT PERMITTED UNDER APPLICABLE LAW, CadSoft DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. SPECIFICALLY, CadSoft MAKES NO REPRESENTATION OR WARRANTY THAT THE SOFTWARE IS FIT FOR ANY PARTICULAR PURPOSE, AND ANY IMPLIED WARRANTY OF MERCHANTABILITY IS LIMITED TO THE NINETY-DAY DURATION OF THE LIMITED WARRANTY COVERING THE SOFTWARE AND PHYSICAL DOCUMENTATION ONLY, AND IS OTHERWISE EXPRESSLY AND SPECIFICALLY DISCLAIMED.

THIS LIMITED WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS; YOU MAY HAVE OTHERS WHICH MAY VARY FROM STATE TO STATE. SOME STATES DO NOT ALLOW THE EXCLUSION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, OR THE LIMITATION ON HOW LONG AN IMPLIED WARRANTY LASTS, SO SOME OF THE ABOVE MAY NOT APPLY TO YOU.

GOVERNING LAW AND GENERAL PROVISIONS

This License and Limited Warranty shall be construed, interpreted and governed by the laws of the State of Florida, U.S.A. If any provision is found void, invalid or unenforceable, it will not affect the validity of the balance of this License and Limited Warranty which shall remain valid and enforceable according to its terms. If any remedy, hereunder, is determined to have failed of its essential purpose, all limitations of liability and exclusions of damages set forth herein shall remain in full force and effect. This License and Limited Warranty may only be modified in writing, signed by you and a specifically authorized representative of CadSoft. All rights not specifically granted in this License Agreement are reserved by CadSoft.

U.S. GOVERNMENT RESTRICTED RIGHTS

The Software and documentation are provided with RESTRICTED RIGHTS. Use, duplication or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights In Technical Data and Computer Software clause at 252.227-7013. Contractor/manufacturer is CadSoft Computer, Inc., 801 South Federal Highway, Suite 201, Delray Beach, Florida 33483-5185, U.S.A.

Table of contents

1 Introduction

2 Installation

3 EAGLE Modules and Editions

4 A First Look at EAGLE

7 The Autorouter

Appendix

Chapter 1
Introduction

This manual describes the use of the EAGLE software and its basic principles. The order of chapters follows the typical process from drawing a schematic to a ready-to-use layout.

1.1 What is in This Manual?

A chapter's main heading is intended to tell you briefly what the contents of that chapter are. Here in the first chapter we want to give a quick overview what you can expect from this manual.

Chapter 1 — Introduction
Contains a preview of the manual.

Chapter 2 — Installation
Deals with the program's installation.

Chapter 3 — EAGLE Modules and Editions
Explains the various program variants.

Chapter 4 — A First Look at EAGLE
Gives a preview of the program's structure and describes the editor windows and their commands.

Chapter 5 — Principles for Working with EAGLE
Examines the basic ways of using and configuring EAGLE.

Chapter 6 — From Schematic to Finished Layout
Follows the route from schematic to layout.

Chapter 7 — The Autorouter
Dedicated to the Autorouter module and its configuration.

Chapter 8 — Component Design Explained through Examples
Explains the definition of library components through examples and informs about library and component management.

Chapter 9 — Preparing the Manufacturing Data
Everything you need to know about generating manufacturing data.

Appendix
Lists useful additional information and explains some error messages EAGLE prompts in certain situations.

For a quick, hands-on introduction, refer to the EAGLE Tutorial. Please read the tutorial for a better understanding before working with the manual.

Anybody who has already been working with a prior version of EAGLE is advised to read the file ***UPDATE*** under Linux or ***UPDATE.TXT*** under Windows. It contains a description of all the differences from earlier versions. This file is located in the *eagle/doc* directory. Please read it before you start working with EAGLE 4.

Information that was not available or that has been changed since printing this manual is also described in ***UPDATE*** or ***UPDATE.TXT***, or, if it exists, in a *README* file.

Detailed information, especially about the EAGLE command language and the EAGLE User Language, is available on the help pages. You can reach a basic understanding very quickly by using this manual, and you can use the convenient search features of the help function to quickly locate the answers to particular questions.

1.2 Technical Terms

In this manual, in the help function, and in EAGLE itself we frequently use some technical terms that should be explained here in a few words.

Airwire:
Unrouted connection on a board, displayed in the unrouted layer (= rubber band).

Annulus Symbol:
An isolation ring that will be drawn in a supply layer or in a copper-filled area.

Blind Via:
A plated-through hole for changing the layer of a track which has not been drilled through all layers in the production process of a multilayer board.

Buried Via:
A plated-trough hole, which has been drilled through the current layer stack in the production process like a normal (through) via, but does not connect all layers of the whole board.

Core: Two copper layers applied to a solid substrate.

Design Rule Check (DRC):
EAGLE can identify the violation of certain Design Rules (e.g. if two different tracks overlap or are too close) with the DRC.

Device:
A fully defined element in a library. Consists of a package and a symbol.

Device Set:
Consists of devices that use the same symbols for the schematic but have different package variants or technologoies.

Drill:
Plated-through drilling in the layout (in pads and vias)

Electrical Rule Check (ERC):
EAGLE can identify the violation of certain electrical rules (e.g. if two outputs are connected) with the ERC. It also checks the consistency of the schematic and the layout.

Forward&Back Annotation:
Transforms all the actions one makes in a schematic online into the layout (and with limitations from layout into schematic). Both files are consistent all the time.

Gate:
The term *gate* is used in this manual for a part of a component which can be individually placed on a schematic. This can be one gate of a TTL component, one contact pair in a relay, or an individual resistor from a resistor array.

Hole:
Non plated-through drilling in the layout (e.g. a mounting hole).

Layer Stack:
Current number and order of copper and isolation layers which are used to build up a printed circuit board.

Micro Via:
A plated-through hole (like blind via) with a relatively small drill diameter which connects an outer layer with the next reachable inner layer.

Net:
Electrical connection in a schematic.

Package:
Component footprint stored in a library.

Pad:
Through-hole pad associated with a package.

Pin:
Connection point on a schematic symbol.

Prepreg:
Used in a compound of inner and outer layers for multilayer boards.

Rack:
Configuration table for a drilling machine. Needed for generating drill data.

Ratsnest:
Command for calculating the shortest airwires.

Restring:
Remaining width of the copper ring around a plated-through pad or via.

Signal:
Electrical connection in a board.

Supply Symbol:
Represents a supply signal in the schematic. Causes the ERC to run special checks.

Symbol:
Schematic representation of a component, stored in a library.

User Language:
Freely programmable, C-like language for data import and export.

Via:
Plated-through hole for changing the layer of a track. See also micro via, blind via, and buried via.

Wheel:
Aperture configuration file. Generated with Gerber data for board manufacturing.

Wire:
Electrical connection in a board, or a line (since lines are drawn with the WIRE command).

Chapter 2
Installation

2.1 What You Have Received

The EAGLE pack, with the license agreement on the outside, contains the EAGLE CD-ROM, a license disk, the User License Certificate with the personal installation code, a training manual and this reference manual.

It may be that not all of these items are included if you have asked for an upgrade to an existing Version 4.1 installation or for an extension of your license. In all cases, however, there is a new License Certificate and a new license disk.

If your computer is not equipped with a floppy disk drive, you may copy the file *license.key* from the license disk to another media or, if available, make it accessible through a network computer.

Keep the User License Certificate with your personal installation code in a safe place, inaccessible to unauthorized persons! Never give the license file or installation code to others! Never send your installation code by e-mail to CadSoft or to anyone else.
You need the license card as evidence of your entitlement to future upgrades or updates.

2.2 New Installations

The CD contains EAGLE for Windows (95, NT 4.0, and above) and Linux. Any particularly recent or additional information about the installation may be found in the *README* files in the relevant directories on the CD-ROM or on an included floppy disk.

Windows

Place the CD-ROM in the drive. After a short delay the CD-ROM startup window should open. If this does not happen, double-click on the CD-ROM symbol in the *My Computer* folder on the Windows desktop.

When the CD-ROM startup window has opened, the first thing to do is to select the language in which you want to work. The user interface, help texts, and additional documentation will be installed in that language. In the next window, click on the *Install program* item, and then simply follow the setup routine.

You will be asked for the license disk as the installation proceeds. Keep it to hand.

The program must be licensed the first time it is called. Enter the path to the license file (usually *A:\license.key*) and your personal installation code, as noted on the EAGLE User License Certificate, into the *EAGLE Product Registration* window.

The program will then start, and will show the license data in the Control Panel. You can display the license data at any time in the EAGLE Control Panel by means of the *Help/Product Information* menu.

Linux

You can either use the RPM or the TGZ archive to install EAGLE on your system. Depending on the selected archive file the user interface, help texts, and additional documentation will be installed in the respective language. The following assumes that you have mounted the EAGLE CD-ROM as */cdrom,* and that you are logged in to your system as root user.

Installing the RPM Archive:

Use *rpm* to install the package:

```
rpm -i /cdrom/english/linux/install/eagle-4.xxe-1.i386.rpm
```

By default this package installs to */opt/eagle*, but you can relocate it to a different directory using rpm’s *-prefix* option. Note that a shell script will be executed that installs a symbolic link to the executable file in */usr/local/bin*, and also copies the manual page to */usr/local/man/man1*. You can use rpm’s *-noscripts* option to avoid this.

Enter the command

```
/opt/eagle/bin/eagle
```

to invoke the product registration (you need to have write access to that directory for doing this!).

Installing the TGZ Archive:

Create a new directory on your system (e.g. */opt/eagle*):

```
mkdir /opt/eagle
cd /opt/eagle
```

Use *tar* to extract the archive:

```
tar xvzf /cdrom/english/linux/install/eagle4.xxe.tgz
```

Change into the directory that has just been extracted from the archive:

```
cd /opt/eagle/eagle-4.xxe
```

Run the installation script:

```
./install
```

Enter the command

```
bin/eagle
```

to invoke the product registration (you need to have write access to that directory for doing this!).

Usage

To use EAGLE you should create a working directory

```
mkdir /home/username/eagle
```

change into that directory

```
cd /home/username/eagle
```

and start the program

```
eagle
```

2.3 Updating an Older Version

Follow the same procedure described in the section on *New Installations* on page 15. Please read the file *update.txt* in the EAGLE directory, in order to familiarize yourself with the changes in the new version of the program. Additional notes on installing an update may be found in the latest *README* files.

For reasons of safety it is good practice to create a backup of your previous data before proceeding!
Files from earlier versions can be used directly with the current one. In case the files were made with a version prior to EAGLE 2.60 you have to convert them with a program named Update26.exe. Detailed information about this can be found on page 243.

2.4 Changing or Extending the License

If you are changing your license you will receive a new User License Certificate with a new installation code, together with a floppy disk and a new license file.

Run EAGLE, and select the item *Product Registration* in the Control Panel under the *Help* menu. You will now be asked for the path to the *license.key* file (on the license disc) and for the installation code. Enter both of these and click *OK*.

The program has now been re-licensed. You can call up the license data at any time in the EAGLE Control Panel by means of the *Help/Product Information* menu.

This method also allows to upgrade a currently installed Freeware.

2.5 Multiple Users and Network Licenses

Multiple-user licenses may be installed separately on different computers, or may be used in a network within the scope of the license conditions. The installation procedure in a network is generally the same as that on a stand-alone computer, and is described in the section on *New Installations* on page 15.

Installing in a Network

The user rights of multiple-user or network licenses differ from those of the single-user version. It does not contain any special network mechanism, and can therefore be used in most networks.

A typical installation can look like this:

The EAGLE program is installed on a server. Library, design, ULP, project and other directories can be freely chosen. After installation, EAGLE is started and licensed from one of the workstations. This requires write access in the *eagle/bin* directory. The license file that has been created, *eagle.key*, is not changed again after the installation. Write access is no longer required.
EAGLE can now be called from all the other workstations. Please ensure that all the workstations call EAGLE in the same way as was used when licensing.

A private working directory (local or on the network) can now be set up for each computer. A user-specific configuration file (*eaglerc.usr* under Windows, or *.eaglerc* under Linux) is located in this directory. Further sub-directories can then, for instance, contain individual projects.

Alternatively it is possible for each computer on which EAGLE is to be available to have a separate license. In this case, copy the *eagle.key* file that was created in the EAGLE program directory (*eagle/bin*) into the computer's private working directory. When first starting the program, enter the installation code and the path to the license file *license.key*.
This procedure is recommended, for instance, for multiple-user licenses for 3 to 5 users who will only work at specific computers.

Special Instructions Under Windows

Path Information

It has been found to be helpful to use the server names in UNC notation when giving the path for calling EAGLE, rather than the drive letters. For example:

```
    \\netservername\eagle\bin\eagle.exe
```

Different Operating Systems at the Working Computers

If network computers having different Windows systems are in use, it is first necessary to perform an installation as described above.

The following situation can arise:
All the computers that are using, for instance, Windows 2000 can run EAGLE. Computers with Windows ME, however, cannot. In order to be able to operate the computers running Windows ME as well, copy the files *eagle.exe* and *eagle.key*, located in the *eagle\bin* directory on the server, to, for instance, *eagle2000.exe* and *eagle2000.key*. Now start all the computers that run Windows 2000, using the command

```
    EAGLE2000
```

Then start EAGLE from a computer with Windows ME, and license it again with the installation code and the license disc. The *eagle.key* file is then valid for all the Windows ME computers. The program is run with

```
    EAGLE
```

You need write authorization for the *eagle\bin* directory to perform the copying and licensing!

Chapter 3

EAGLE Modules and Editions

3.1 EAGLE Modules

A number of EAGLE editions are offered. You can add an Autorouter Module and/or a Schematic diagram Module to the Layout Editor. The term module is used because EAGLE always behaves like one single program. The user interface is identical for all parts of the program.

The Layout Editor, the Basic Module

The basic EAGLE software package comes with the Layout Editor, which allows you to design printed circuit boards (pcbs), plus the Library Editor, the CAM Processor, and the text editor. With the Library Editor you can already design packages (footprints), symbols and devices (for a schematic). The CAM Processor is the program which generates the output data for the production of the pcb (e.g. Gerber or drill files). It is also possible to use User Language programs and script files.

Schematic Module

If you have the schematic module you begin by drawing a circuit diagram. You can generate the associated circuit board at any time with a mouse-click. EAGLE then changes to the Layout Editor, where the packages are placed next to an empty board - connected via airwires (rubber bands). From here you can go on designing with the Layout Editor as usual. Schematic and layout are automatically kept consistent by EAGLE (Forward&Back Annotation). Schematic diagrams can consist of up to 99 sheets.

Autorouter

You can route the airwires automatically if you own the Autorouter module. You can choose single nets, groups of nets or all nets for the automatic routing pass (AUTO command). The program will handle various network classes having different track widths and minimum clearances.

3.2 Different Editions

EAGLE offers various performance/price categories (editions) called Light, Standard, and Professional. The facilities mentioned in this manual always refer to the Professional edition.

Professional Edition

General

- maximum drawing area 64 x 64 inches
- resolution 1/10,000 mm (0.1 microns)
- mm or inch grid
- up to 255 layers, user definable colors
- command (script) files
- C-like User Language for data export and import and the realization of self-defined commands
- easy library editing
- composition of self-defined libraries by Drag&Drop
- simple generation of new package variants from other libraries by Drag&Drop
- free rotation of package variants (0.1-degree steps)
- library browser and convenient component search function
- technology support (e. g. 74L00, 74LS00..)
- output of manufacturing data on plotter, photo plotter and drilling machine or as a graphic data format
- print via the operating systems' printer devices
- user definable, free programmable User Language for generating data for mounting machines, in-circuit tester and milling machines
- Drag&Drop in the Control Panel
- Automatic backup function

Layout Editor

- full SMD support
- support of blind and buried vias
- rotation of elements in arbitrary angles (0.1-degree steps)
- texts can be placed in any orientation
- dynamic calculation of signal lines while routing the layout
- tracks can be layed out with rounded corners in any radius
- mitering to smooth wire joints
- Design Rule Check for board layouts
- copper pouring
- support of different package variants

Schematic Module

- up to 99 sheets per schematic
- simple copying of parts
- online Forward&Back Annotation between schematic and board
- automatic board generation
- automatic generation of supply signals (for IC's)
- Electrical Rule Check (checks logic in the schematic and the consistency of schematic and layout)

Autorouter Module

- fully integrated into basic program
- uses the set of Design Rules you defined for the layout
- change between manual and automatic routing at any time
- ripup&retry algorithm
- user-definable strategy (by cost factors)
- routing grid down to 0.8 mil (0.02 mm)
- no placement restrictions
- up to 16 signal layers (with user definable preferred directions)
- up to 14 supply layers
- full support of blind and buried vias
- recognizes signal classes that define special settings for wire with and minimum distances

Standard Edition

Circuit boards up to a maximum size of 160 mm x 100 mm can be created with the Standard Edition. The components can only be placed within the range of coordinates from (0 0) up to (160 100) mm.

The number of signal layers is limited to 4. This means that two inner layers are possible.

Light Edition

The Light edition permits boards with a maximum size of 100 mm x 80 mm (half eurocard format).

A maximum of 2 signal layers (top and bottom) may be used.

Circuit diagrams can only be drawn on a single sheet.

Chapter 4
A First Look at EAGLE

4.1 The Control Panel

The Control Panel normally appears after starting EAGLE, and this is the program's control center. All the files specific to EAGLE are managed here, and some basic settings can be made. It is similar to the familiar file managers used by a wide variety of applications and operating systems. Each EAGLE file is displayed in the tree view by means of a small symbol.

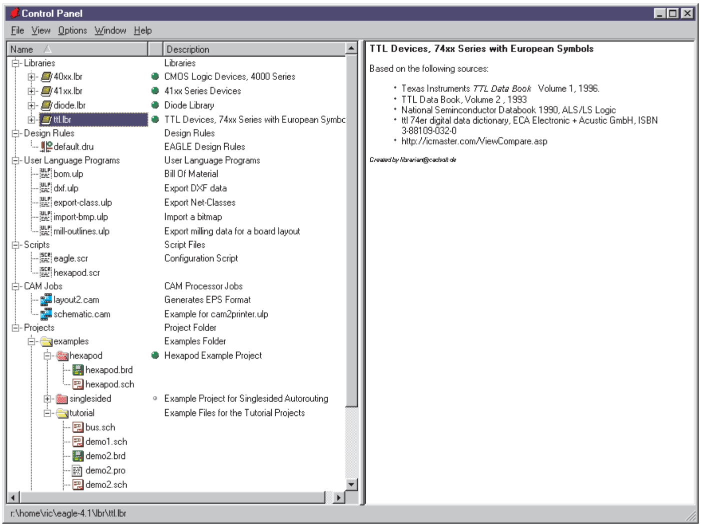
A context menu is opened by clicking with the mouse on an entry in the tree view. This allows you, depending on the object, to carry out a variety of actions (rename, copy, print, open, create new etc.).

The Control Panel supports Drag&Drop. This can also be done between different programs. You can, for instance, copy files, move them, or create links on the desktop. User Language programs or script files that are pulled with the aid of the mouse out of the Control Panel and into an editor window are started automatically. If, for instance, you pull a board file with the mouse into the Layout Editor, the file is opened.

The tree structure provides a quick overview of the libraries, Design Rules, User Language programs, script files, CAM jobs and projects. Special libraries, text, manufacturing and documentation files can belong to a project as well as schematic diagrams and layouts.

The first time it is called, the Control Panel will appear very much as shown in the following diagram. If an object is selected in the tree view, further relevant information is displayed in the right hand part of the window.

Simply click on various folders and files in order to experiment with the Control Panel's facilities.



Name		Description
Libraries		Libraries
40xx.lbr	●	CMOS Logic Devices, 4000 Series
41xx.lbr	●	41xx Series Devices
diode.lbr	●	Diode Library
ttl.lbr	●	TTL Devices, 74xx Series with European Symbc
Design Rules		Design Rules
default.dru		EAGLE Design Rules
User Language Programs		User Language Programs
bom.ulp		Bill Of Material
dxf.ulp		Export DXF data
export-class.ulp		Export Net-Classes
import-bmp.ulp		Import a bitmap
mill-outlines.ulp		Export milling data for a board layout
Scripts		Script Files
eagle.scr		Configuration Script
hexapod.scr		
CAM Jobs		CAM Processor Jobs
layout2.cam		Generates EPS Format
schematic.cam		Example for cam2printer.ulp
Projects		Project Folder
examples		Examples Folder
hexapod	●	Hexapod Example Project
hexapod.brd		
hexapod.sch		
singlesided	○	Example Project for Singlesided Autorouting
tutorial		Example Files for the Tutorial Projects
bus.sch		
demo1.sch		
demo2.brd		
demo2.pro		
demo2.sch		

r:\home\ric\eagle-4.1\lbr\ttl.lbr

TTL Devices, 74xx Series with European Symbols

Based on the following sources:

- Texas Instruments *TTL Data Book* Volume 1, 1996.
- TTL Data Book, Volume 2 , 1993
- National Seminconductor Databook 1990, ALS/LS Logic
- ttl 74er digital data dictionary, ECA Electronic + Acustic GmbH, ISBN 3-88109-032-0
- http://icmaster.com/ViewCompare.asp

Created by librarian@cadsoft.de

Control Panel: On the right, the description of a TTL library

Library Summary

The possibility of displaying the contents of the libraries is particularly interesting. It provides a very rapid overview of the available devices.

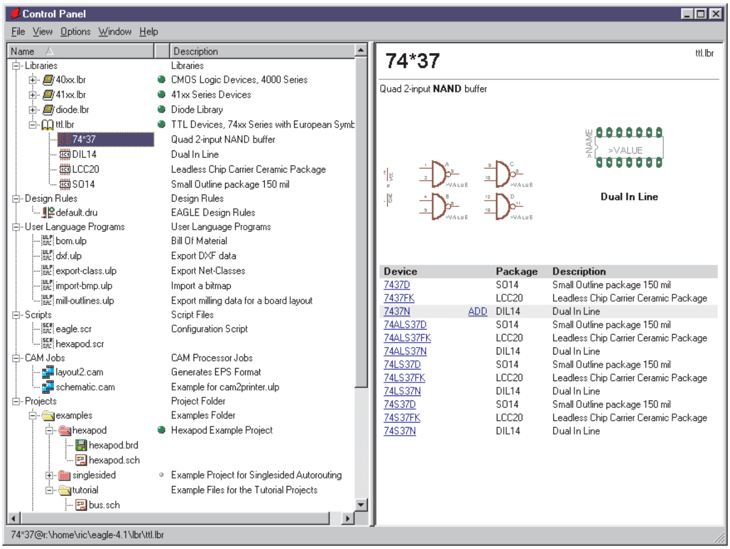
Double-click on the *Libraries* entry. The library branch opens, and you can see the available libraries. In the *Description* field you can see a brief description of the contents. If a library is selected you will see more extensive information about the library in the right hand part of the Control Panel. If you then double-click on a library, the contents will be displayed together with a short description of each element. Devices and packages are marked with a small symbol.

Now select, for example, a device.
The description of the device and a graphical representation of it appear on the right. The available package and technology variants are listed. If you click on one of the package versions, the package shown above will change.
If a Schematic Editor window is open, the entry *ADD* will be shown right of the variant name. Click it and the device is attached to the mouse cursor as soon as it is over the Schematic Editor window. Now you can drop it in the schematic.

If you are only working with the Layout Editor, this will of course also operate with packages. It is, additionally, possible to drag a device from the tree view into a schematic diagram and to place it there by means of Drag&Drop. If it has more than one package version, the ADD dialog opens automatically, so that the desired package can be selected.

The green marker behind the library entry indicates that this library is *in use*. This means that it can be used in the current project. Devices in this library will be examined by the search function in the ADD dialog of the schematic diagram or of the layout. This makes them available for the project. The library will not be examined if the marking is gray.
If starting EAGLE without a project (no *eagle.epf* file is read, the project has been closed before exiting EAGLE last time) and creating a new project (⇒*File/New/Project*) all libraries will be *in use* automatically. However, opening an already existing project, where only certain libraries are *in use* before creating the new project, will adopt this selection.

If the Library Editor window is open you can Drag&Drop a complete Device set or package definition from the Control Panel into the library window. This way you can copy it from one library into another.



Name		Description
Libraries		Libraries
40xx.lbr	●	CMOS Logic Devices, 4000 Series
41xx.lbr	●	41xx Series Devices
diode.lbr	●	Diode Library
ttl.lbr	●	TTL Devices, 74xx Series with European Symb
74*37		Quad 2-input NAND buffer
DIL14		Dual In Line
LCC20		Leadless Chip Carrier Ceramic Package
SO14		Small Outline package 150 mil
Design Rules		Design Rules
default.dru		EAGLE Design Rules
User Language Programs		User Language Programs
bom.ulp		Bill Of Material
dxf.ulp		Export DXF data
export-class.ulp		Export Net-Classes
import-bmp.ulp		Import a bitmap
mill-outlines.ulp		Export milling data for a board layout
Scripts		Script Files
eagle.scr		Configuration Script
hexapod.scr		
CAM Jobs		CAM Processor Jobs
layout2.cam		Generates EPS Format
schematic.cam		Example for cam2printer.ulp
Projects		Project Folder
examples		Examples Folder
hexapod	●	Hexapod Example Project
hexapod.brd		
hexapod.sch		
singlesided	○	Example Project for Singlesided Autorouting
tutorial		Example Files for the Tutorial Projects
bus.sch		

74*37@r:\home\ric\eagle-4.1\lbr\ttl.lbr

74*37

Device		Package	Description
7437D		SO14	Small Outline package 150 mil
7437FK		LCC20	Leadless Chip Carrier Ceramic Package
7437N	ADD	DIL14	Dual In Line
74ALS37D		SO14	Small Outline package 150 mil
74ALS37FK		LCC20	Leadless Chip Carrier Ceramic Package
74ALS37N		DIL14	Dual In Line
74LS37D		SO14	Small Outline package 150 mil
74LS37FK		LCC20	Leadless Chip Carrier Ceramic Package
74LS37N		DIL14	Dual In Line
74S37D		SO14	Small Outline package 150 mil
74S37FK		LCC20	Leadless Chip Carrier Ceramic Package
74S37N		DIL14	Dual In Line

Control Panel: Library Summary with Device View

Design Rules

Special Design Rules can be specified in EAGLE to govern the board design. These can be saved as data sets in special files (**.dru*).

The parameter set that is to govern the current project is specified in the *Design Rules* branch of the tree view. If no data has been provided for the Design Rules (DRC command), EAGLE will itself provide parameters. The marking to the right of the file entry specifies the default parameter set for the current project. The layout will be checked by the DRC in accordance with these criteria. Further information about the DRC and the Design Rules is found starting on page 98.

User Language Programs, Scripts, CAM Jobs

These entries show the contents of the *ulp*, *scr* and *cam* directories. They contain various User Language programs (**.ulp*), script files (**.scr*) and CAM jobs (**.cam*) for the output of data using the CAM Processor. If one of these files is selected in the Control Panel, you will see a full description of the file.

The paths can be set by means of the *Options/Directories* menu. This is discussed in more detail later in this chapter.

Projects

The various projects are managed from the Control Panel. A click on the *Projects* entry displays various folders. These are located under the path set under *Options/Directories/Projects*. It is allowed to define more than one path there.

A project usually consists of a folder (directory) which represents the project by its name and the project's configuration file *eagle.epf.* The folder usually contains all files that belong to your project, for example, schematic and board file, special library files, script files and so on.

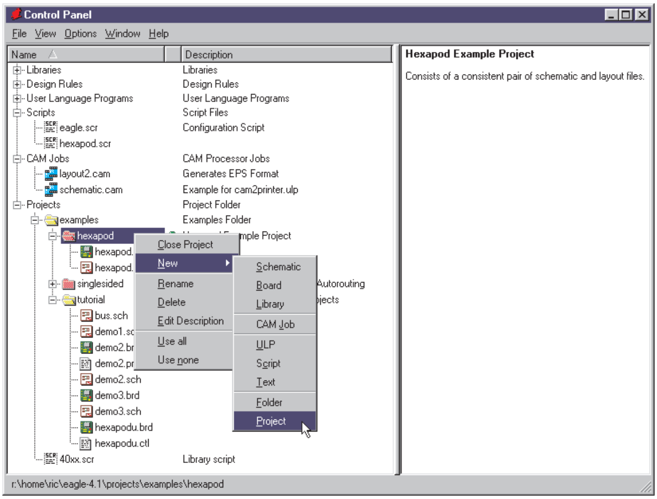
Project directories that contain the project file *eagle.epf* will be marked with a red folder icon. 'Normal' folders will be marked with a yellow icon.

The project to be edited is selected in the *Projects* branch. On the right of the project's name you will find a marker which is either gray or green. With the help of this marker one can open or close projects. Clicking onto a gray marker, loads the project. The marker appears green now. Clicking on the green marker again or clicking onto another gray marker closes the current project respectively opens another project after closing the current one. This way one can switch easily from one project to another.

While closing a project the current settings will be stored in the corresponding project file *eagle.epf*, provided that the option *Automatically save project file* is set in the *Options/Backup* menu.

New projects are created by clicking the right mouse button on a folder entry in this branch. A context menu opens which permits new files and directories to be created and the individual projects to be managed.
Selecting the option *New/Project* invokes a new folder which has to be given the project's title. The project file *eagle.epf* will be created automatically.

You can also use the *File/Open/Project* or the *File/New/Project* menu to open or create a new project.



Name	Description
Libraries	Libraries
Design Rules	Design Rules
User Language Programs	User Language Programs
Scripts	Script Files
eagle.scr	Configuration Script
hexapod.scr	
CAM Jobs	CAM Processor Jobs
layout2.cam	Generates EPS Format
schematic.cam	Example for cam2printer.ulp
Projects	Project Folder
examples	Examples Folder
hexapod	mple Project
hexapod.	
hexapod.	
singlesided	Autorouting
tutorial	jects
bus.sch	
demo1.sc	
demo2.br	
demo2.pr	
demo2.sch	
demo3.brd	
demo3.sch	
hexapodu.brd	
hexapodu.ctl	
40xx.scr	Library script

Hexapod Example Project

Consists of a consistent pair of schematic and layout files.

r:\home\ric\eagle-4.1\projects\examples\hexapod

Context menu for project management

The context menu contains the *Edit Description* item. A description of the project can be entered here, and this is then displayed in the *Description* box.

Menu Bar

The Control Panel allows various actions to be executed and settings made through pull-down menus that are explained below.

File Menu

The *File* menu contains the following items:

New

Creates a new layout (board), schematic, library, CAM job, ULP, script or text file. The *Project* option creates a new project. This initially consists simply of a new directory in which the files for a new project are handled. These will consist as a rule of the schematic diagram and layout, possibly of special libraries, script files, User Language programs, documentation files etc. and of the file *eagle.epf*, in which project-specific settings are stored.

The default directories for the various file types are defined in the *Options/Directories* menu.

CAM jobs are definitions for generating output data which are created with the CAM Processor.

Script and ULP files are text files containing command sequences in the EAGLE command language or the EAGLE User Language. They can be created and edited with the EAGLE text editor, or with any other text editor.

Open

Opens an existing file of the types mentioned above.

Open recent projects

Lists recently used projects.

Save all

All changed files are saved. The current settings for the project are saved in the file *eagle.epf*.

Close project

The project will be closed. Project-specific settings are saved in the *eagle.epf* file of the current project directory.

Exit

The program is terminated. When EAGLE is started again, the last program status is restored, i.e. the windows and other working environment parameters appear unchanged. If there was no project loaded only the Control Panel will be opened next time.

The current status is also saved when you leave EAGLE with Alt-X from any program part.

View Menu

Refresh

The contents of the tree view are updated.

Sort

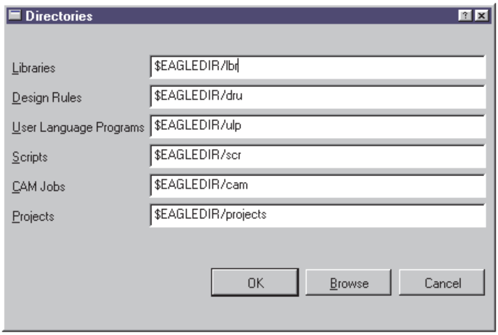
The contents of the tree view will be sorted *by name* or *by type*.

Options Menu

Directories

The default directories for particular EAGLE files are entered in the directories dialog box.
More than one path may be entered for each of these. In the Windows version the entries are separated by semicolons, while a colon is used in

the Linux version. The *Projects* directory is the default directory for the text editor.
The *Projects* directory contains subdirectories, each of which represents a particular project. Each of the project directories contains an EAGLE project file (*eagle.epf*). A project directory and its subdirectories usually contain all the files that are associated with one particular project, such as the schematic diagram and the layout, text files, manufacturing data, documentation files and so on.



Libraries $EAGLEDIR/lbr
Design Rules $EAGLEDIR/dru
User Language Programs $EAGLEDIR/ulp
Scripts $EAGLEDIR/scr
CAM Jobs $EAGLEDIR/cam
Projects $EAGLEDIR/projects

The directories dialog in the Options menu

Type the path directly into the corresponding box, or select the desired directory by clicking on the *Browse* button.

The default settings can be seen in the diagram above. *$EAGLEDIR* stands for the installation's EAGLE directory.

You may also use *$HOME* for your home directory under Linux. Under Windows it is possible to define this environment variable wit the SET command.

If a HOME variable has not been set within the environment, then under Windows EAGLE will suggest the directory *My Documents*.

This directory is defined in the Windows registry in:

```
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersi-
on\Explorer\Shell Folders\Personal
```

It is of course also possible to specify paths with an absolute format.

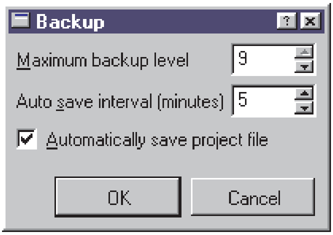
Please note:
The HOME variable must not point to the root directory of a drive.

Backup

When files are saved, EAGLE creates backup copies of the previous files. The *maximum backup level* field allows you to enter the maximum number of backup copies (default: 9). Backup files have different file extensions, enumerated sequentially. Schematic files receive the ending *s#x*, board files *b#x*, and library files *l#x*. *x* can run from 1 to 9. The file with *x* = 1 is the newest one.

The automatic backup function also permits the backup to be scheduled. The time-interval can be between 1 and 60 minutes (default: 5 minutes). The backup files have the endings *b##*, *s##* and *l##* respectively.

All these backup files can be further processed in EAGLE if they are renamed and given the usual file endings (*brd*, *sch*, *lbr*).



Maximum backup level 9

Auto save interval (minutes) 5

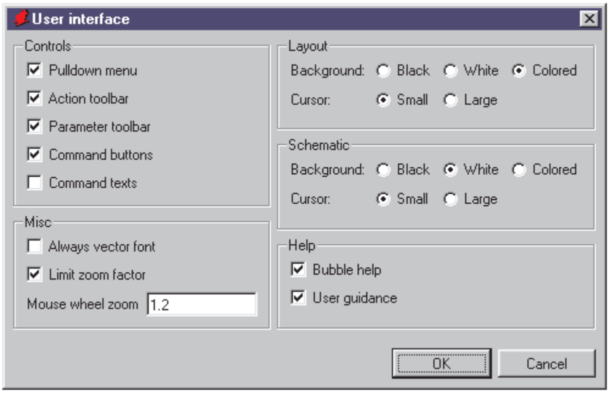
☑ Automatically save project file

Backup dialog

If the option to *Automatically save project file* is chosen, your project is automatically saved when you close the current project or leave the program.

User Interface

The User Interface dialog allows the appearance of the editor windows for the layout, schematic diagram and library to be adjusted to your preferences.
You can also access this menu from the Editor windows.



Controls
☑ Pulldown menu
☑ Action toolbar
☑ Parameter toolbar
☑ Command buttons
□ Command texts

Misc
□ Always vector font
☑ Limit zoom factor
Mouse wheel zoom 1.2

Layout
Background: ○ Black ○ White ● Colored
Cursor: ● Small ○ Large

Schematic
Background: ○ Black ● White ○ Colored
Cursor: ● Small ○ Large

Help
☑ Bubble help
☑ User guidance

Settings for the User Interface

In the *Controls* box you specify which elements are to be displayed in the editor window. If you deactivate all the *Controls*, only the command line will remain for entry. This maximizes the free area available for the drawing.

The option *Always vector font* shows and prints texts with the built-in vector font, independently from the originally used font. Using the vector font guarantees that the output with a printer or the CAM Processor is exactly the same as shown in the editor window. Fonts other than vector font depend on the systems' settings and cannot be controlled by EAGLE. The output of non-vector fonts may differ from the editor's view.
Opening the *User Interface* dialog from one of the Editor windows (for example, the Layout Editor) the *Always vector font* option offers an additional item *Persistent in this drawing*. Setting this option causes EAGLE to save the *Always vector font* setting in the current drawing file. So you can be quite sure that the layout will be shown with vector font at another's person computer (for example, at a boardhouse).
Please see the help function for details (TEXT command).

Limit zoom factor limits the maximum zoom factor in an editor window. Deactivating this option allows to zoom into the drawing as far as one square millimeter.
Please be careful with this option! Some graphics devices won't display a proper drawing screen exceeding a certain zoom factor (16 bit coordinate range). The worst case may result in a system crash due to a sloopy programmed driver.

If you are working with a wheel mouse, you can zoom in and out by turning the mouse wheel. *Mouse wheel zoom* determines the zoom factor. The value 0 switches this function off.

The background color and the appearance of the drawing cursor can be separately adjusted for the layout and the schematic diagram editors. The *background* may be black, white or shown in any other color (*Colored*). The background color definition is described on page 80.

The cursor can be displayed optionally as a *small* cross or as *large* crosshairs.

The check boxes in the *Help* area allow you to switch on or off the popup texts for the command icons (*Bubble help*) and the help texts for the commands in the status line (*User guidance*).
Deselecting the *User guidance* check box allows commands like NET, MOVE, ROUTE, SHOW to show information about the selected object in the satus bar of the editor window.

Window Menu

From the *Window* menu you can choose the window (schematic, board, etc.) to be displayed in the foreground. The number on the left is the window number. It allows you to choose a window when combined with the *Alt* key (e.g. Alt+1 selects window 1).

The combination Alt+0 can be used anywhere in the program to bring the Control Panel into the foreground.

Help Menu

The *Help* menu contains an item for calling the help function, as well as items for installing a new license (*Product registration*) and getting information about the program version etc. (*Product information*).

Product Registration

The registration dialog is called automatically when you start EAGLE the first time. If you want to install an upgrade you must start this dialog from the *Help* menu, and then enter the necessary information according to the *License/Product Registration* section of the help function. All editor windows have to be closed before. Read the notes in the chapter *Installation* for more information.

Product Information

If you contact our technical support you should provide the information you will find under this menu item.

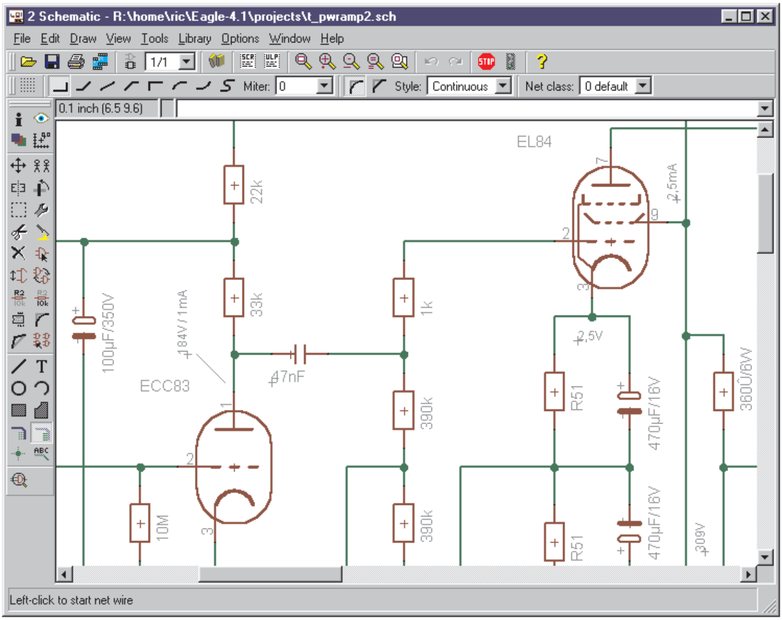
4.2 The Schematic Editor Window

The Schematic Editor window opens when you load an existing schematic or create a new one. There are several ways of opening files in EAGLE.

You can, for instance, load a schematic diagram by means of the *File/Open/Schematic* menu in the Control Panel. Alternatively you can double-click on a schematic diagram file in the tree view.

If you want to create a new schematic diagram, select the menu *File/New/Schematic*. This will open a schematic with the name *untitled.sch* in the current project directory.

If you want to create a schematic diagram straight away in a new project, you may for example click with the right mouse button on a project in the *Projects* entry of the tree view, and select the *New project* option from the context menu. The new project receives a name. Then click on this entry with the right mouse button. Now select *New/Schematic* from the context menu.
A new schematic opens in this project directory.


2 Schematic - R:\home\ric\Eagle-4.1\projects\t_pwramp2.sch

Miter: 0

Style: Continuous

Net class: 0 default

0.1 inch (6.5 9.6)

Left-click to start net wire

The Schematic Editor

On top you will see the **title bar**, which contains the file name, and then the **menu bar**, and the **action toolbar**.

Below the action toolbar there is the **parameter toolbar,** which contains different icons, depending on the active command.

Above the working area you will find the **coordinate display** on the left, with the **command line**, where commands can be entered in text format, to the right of it.

EAGLE accepts commands in different but equivalent ways: as mouse clicks, text via keyboard, or from command (script) files.

On the left of the work space you find the **command toolbar**, which contains most of the Schematic Editor’s commands.

Below, in the **status line**, instructions for the user appear if a command is active.

Each of the toolbars can be displayed or hidden using *Options/User Interface*. It is also possible to rearrange the toolbars within certain limits with the aid of the mouse. The command toolbar, for instance, can also be placed on the right, or the action and parameter toolbars can be placed together on one line.

How You Obtain Detailed Information About a Command

Bubble Help or Tool Tips

If the mouse cursor remains above an icon for longer than a certain time, the name of the EAGLE command appears. You also see a short explanation below in the status line.

For example, move the cursor over the WIRE icon. Bubble help with the word *Wire* appears directly by the cursor. The short description, *Draw lines*, appears in the status line.

If you select the command, a short note appears below in the status line, indicating what would normally be expected as the next action. For instance, if you click on the WIRE icon, the status line will display the instruction: *Left-click to start wire* .

These functions can be activated or cancelled in the Control Panel by means of the *Options/User Interface* menu.

Help Function

If you want to learn more about a command, e.g. the WIRE command, click its icon in the command toolbar, then click the help icon.

As an alternative you can type

HELP WIRE ←

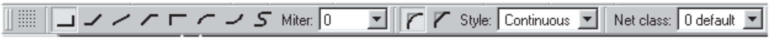
into the command line. The ← character symbolizes the *Enter* key.

Command Parameters

A number of EAGLE commands need additional parameters. Refer to the help pages for a description of the textual entry of parameters (via command line or script file).

Most of the parameters can be entered by clicking the appropriate icons in the parameter toolbar, which changes according to the selected command. These icons also show bubble help explanations.

This is how the parameter toolbar appears when the NET command is activated.



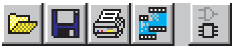
Miter: 0 Style: Continuous Net class: 0 default

On the left is the GRID icon for setting the grid pitch. To the right are buttons for the bend mode (SET WIRE_BEND) of the net line, followed by the miter radius for smoothing line joints with the options straight or rounded (see MITER command). Next to this is the *Style* menu where the type of line is defined. On the far right is a value menu for assigning a *Net class*.

GRID command: This icon is available at any time. It is used to adjust the grid and to select the current unit. In EAGLE, any value relates to the current unit.

The Action Toolbar

This toolbar is composed of the following icons:



From the left: Open file, save file, print file, call CAM Processor, open/create corresponding board file (BOARD command).

1/1

Load, remove, or create a new schematic sheet.

USE command: Select libraries which will be taken into consideration by the ADD dialog. Can also be done with the *Library/Use* menu item or by clicking the markers in the *Libraries* branch of the Control Panel's tree view. The context menu of the entry *Libraries* or of its subfolders contains the entries *Use all* and *Use none* for a quick and simple

selection/deselection of all libraries (of the folder).
This command has to be used in script files in order to choose the library you want to take parts from.

SCRIPT command: Execute a script file. This enables you to execute any command sequence with a few mouse clicks.

RUN command: Start a User Language program (ULP).

WINDOW command: These icons represent different modes of the WINDOW command: Fit drawing into the screen (Alt-F2), zoom in (F3), zoom out (F4), redraw screen (F2), display new area.
That part of the drawing displayed on the screen can be shifted by holding down the *Ctrl* key and moving the mouse at the same time.

To move the current drawing window click the middle mouse button and move your mouse!
Press the Shift key simultaneously and you are allowed to scroll beyoned the drawing boundaries!

UNDO and **REDO**: These commands allow you to cancel previous commands and to execute commands which have previously been cancelled. Function keys: F9 and F10 (default).

Terminates the execution of EAGLE commands (*Edit/Stop command*).

Starts the execution of the active EAGLE command. This is only necessary if further parameters could be entered by the user.

The Schematic Command Toolbar

INFO

Provides information about the object to be selected.

SHOW

Highlights the object to be selected.

DISPLAY

Select and deselect the layers to be displayed. See the *Appendix* for the meaning of the layers. For further details please see help function.

MARK

The following mouse click defines the new origin for the coordinate display. Relative coordinates *(*`R` *x-value y-value)* and polar values *(*`P` *radius angle)* are shown in addition to absolute coordinates in the coordinate display box. If you first click the MARK icon and then the traffic-light icon, only the absolute coordinate values will be displayed again.

MOVE

Move any visible object. The right mouse button rotates the object.

If you move a net over a pin, no electrical connection will be established. If you move the pin of a gate over a net or another pin, an electrical connection (net) will be created.

To move groups of objects: define the group with the GROUP command, click the MOVE icon, then select the group with the right mouse button and move it to the desired location. The right mouse button rotates the group by 90 degrees.

COPY

Copy parts and other objects.

MIRROR

Mirror objects.

ROTATE

Rotate objects by 90 degrees (also possible with MOVE).

GROUP

Define a group which can then be moved, rotated, or copied (with CUT and PASTE) to another drawing. After the icon has been clicked, a rectangular group can be defined by holding down the left mouse button and dragging the cursor to the diagonal corner of the rectangle. If you want to define a group by a polygon, use the left mouse button to determine the corners of the polygon. Then click the right mouse button to close the polygon.

CHANGE

Change the attributes of an object, e.g. the width of a line, the package variant or the size of text. See help for details.

CUT

Transfer the objects of a previously defined group into the paste buffer. See PASTE command. Not identical to the Windows cut function.

PASTE

Insert objects from the paste buffer into the drawing. Restrictions: see help. Not identical to the Windows paste function.

DELETE

Delete visible objects.

Also in combination with GROUP command. If a group has been defined, it can be deleted with the right mouse button.

The DELETE command deletes an entire part when clicking on a gate with the *Shift* key pressed. In that case, the wires connected to the element in the board will not be ripped up.

Clicking on a net or bus wire with the *Shift* key pressed deletes the entire net or bus segment.

ADD

Add library elements to the schematic. A search function helps devices to be found quickly. USE specifies which libraries are available.

PINSWAP

Swap two nets connected to equivalent pins of a device, provided the pins have been defined with the same swaplevel.

GATESWAP

Swap two equivalent gates of a device, provided the gates have been defined with the same swaplevel. In EAGLE terminology, a gate is a part of a device which can be individually placed on a schematic (e.g. one transistor from a transistor array).

NAME

Give names to components, nets, or buses.

VALUE

Provide values for components. Integrated circuits normally get the type (e.g. 74LS00N) as their value.

SMASH

Separate name and value texts from a device, so that they can be placed individually. The size of smashed texts can also be individually changed. Also in combination with GROUP. If a group is defined, you can smash it with a right mouse click.

Keep the *Shift* key pressed while using the SMASH command in order to unsmash texts. They are not editable anymore and appear at their original positions after a window refresh.

MITER

Round off or bevel wire joints (also possible for nets, buses, polygon contours). The grade of mitering is determined by the miter radius. Positive sign results in a rounded joint, negative sign in a bevel.
The miter radius influences some wire bends, too (see help function: SET, *Wire_Bend*).

SPLIT

Insert an angle into a wire or net.

INVOKE

Fetch a particular gate from a device (e.g. gate D before gate C). This command allows you also to add a gate from a device which is located on another sheet. In such a case, type the name of the device (e.g. IC1) into the command line after the INVOKE command has been selected.

WIRE

Draw line (this command is called WIRE because it is used to define electrical connections, i.e. wires, in the Layout Editor). The type of line can be changed with CHANGE STYLE. Clicking the right mouse button changes the bend mode (SET WIRE_BEND).

WIRE can also be used to draw arcs.

Please note the particularities in combination with the *Ctrl* and *Shift* key in the help function.

TEXT

Placing text. Use CHANGE SIZE to alter the height of the text. If the text is using a vector font, CHANGE RATIO will alter the thickness. CHANGE TEXT is used to alter the text itself. CHANGE FONT alters the typeface. You change label texts by assigning a different name to the bus or to a net by means of the NAME command. See also LABEL command.

CIRCLE

Draw a circle. Circles with a width of 0 are drawn as filled circles.

ARC

Draw an arc (also possible with WIRE).
CHANGE CAP FLAT | ROUND defines straight or rounded ends for arcs.

RECT

Draw a rectangle.

POLYGON

Draw a polygon (copper areas in any shape).

BUS

Draw a bus line. The meaning of a bus is more conceptual than physical. It is only a means to make a schematic easier to read. Only nets define an electrical connection. Nets, however, can be dragged out of a bus.

NET

Draw a net. Nets with the same name are connected (even if located on different sheets).

Nets and pins which appear to the eye to be connected are not necessarily electrically connected. Please check with the SHOW command, the ERC, or by exporting a netlist with the EXPORT command. See also MOVE.

JUNCTION

Place the symbol for a net connection. In general, junctions are placed automatically, but nets which cross over can also be joined manually by the JUNCTION command.

LABEL

Place the name of a bus or net as a label. Labels cannot be changed with CHANGE TEXT but rather with the NAME command.

ERC

Perform an Electrical Rule Check and a consistency check for schematic and board.

Commands Not Available in the Command Toolbar

Menu items already explained in the Control Panel section are not discussed here.

The following commands can be entered into the command line. Some of them are available as menu items.

ASSIGN

Assign function keys. The most convenient way of doing this is to use the *Options/Assign* menu.

CLASS

Select and define net classes (*Edit/Net classes...*). A net class specifies the width of a track, the clearance from neighboring signals, and the diameter of vias for the Autorouter and the ROUTE command. These settings are also used in polygons.

CLOSE

Text command for closing an editor window (*File/Close*).

EDIT

Text command for loading a file or a library object. You can, for instance, load a board from the Schematic Editor (EDIT name.brd).

EXPORT

Output lists (especially netlists), directories, script files, or images (*File/Export...*).

LAYER

Choose or define the drawing layer. When using drawing commands the layer can be chosen in the parameter toolbar.

MENU

Specifies the contents of the text menu. See also the example in the appendix. The text menu can be made visible with the aid of *Options/User Interface*. See help function for details.

OPEN

Text command for opening a library for editing (*Library/Open*). This command is <u>not</u> identical to the *File/Open* menu item of the Schematic Editor, which only lets you select schematics. You can use the OPEN command as an alternative to the *File* menu of the Control Panel.

PRINT

Call up the print dialog with the printer icon in the action toolbar or the menu item *File/Print...*. Normally the PRINT command is used to print schematics or check the drawings needed for the pcb production. The actual production data are generated with the CAM Processor. If you want to output your drawing in black and white check the *Black* option (and *Solid*, if you don't want layers to be printed in their different fill styles). The caption text is suppressed unless you check *Caption* (available via the *Page* button) . Set *Page limit* to 1 if your drawing is to be fitted on one page.

QUIT

Quit EAGLE. Identical with the menu item *File/Exit* or Alt-X.

REMOVE

Delete files or schematic sheets.

```
    REMOVE .S3 ←
```

for instance, deletes sheet 3 of the loaded schematic.

SET

Set system parameters and modes. Best done via the *Options/Set* menu item. Please note that not all of the possibilities are available through this dialog. Presettings can be defined in the script file *eagle.scr* by using text commands. Further information can be found in the help function.

UPDATE

The UPDATE command checks the parts in a board or schematic against their respective library objects and automatically updates them if they are different. (*Library/Update...* or *Library/Update all*).
The context menu in the Control Panel's' tree view offers the Options *Use all* and *Use none* for a quick selection of libraries.

USE

Select libraries that will be referred to by the ADD dialog. Click the USE icon in the Layout Editor's or Schematic Editor's action toolbar or select the menu item *Library/Use* to start this command. This selection can also be done by clicking on the library markers (green for selected, gray for deselected) in the *Libraries* branch of the Control Panel's tree view. The context menu of the *Libraries* entry in the tree view contains two items: *Use all* and *Use none* for an easy selection of all or none of the libraries.
The command has to be used in script files to select the library from which one wants to use elements.

WRITE

Text command for saving the currently loaded file. Please note that, in contrast to *Save as*, the name of the currently edited file is never changed when the WRITE command is used.

Mouse Keys

The middle and right mouse button have a special meaning for a number of commands.You can use the middle mouse button only if the operating system knows your mouse is a 3-button mouse, that is your mouse must be installed this way.

If you are working with a wheel mouse you can use the wheel to move the drawing area in an editor window. Selecting another value than 0 for the option *Mouse wheel zoom* in the *Options/User Interface* menu sets a zoom factor for zooming in and out with the mouse wheel.

Mouse clicks in combination with the *Shift*, *Ctrl*, and *Alt* key can have various functions, for example, while selecting objects with MOVE or while drawing lines with WIRE.
The help section on *keyboard and mouse* and the help of the referring command gives you more details.

Selecting Neighboring Objects

If one of two objects which are very close together is to be selected, the individual objects are highlighted one after the other. The user can select the highlighted object with the left mouse button, or proceed to the next one with the right mouse button. The status bar of the editor window shows information about the pre-selected object. See also help function (SET command, SELECT_FACTOR).

4.3 The Layout Editor Window

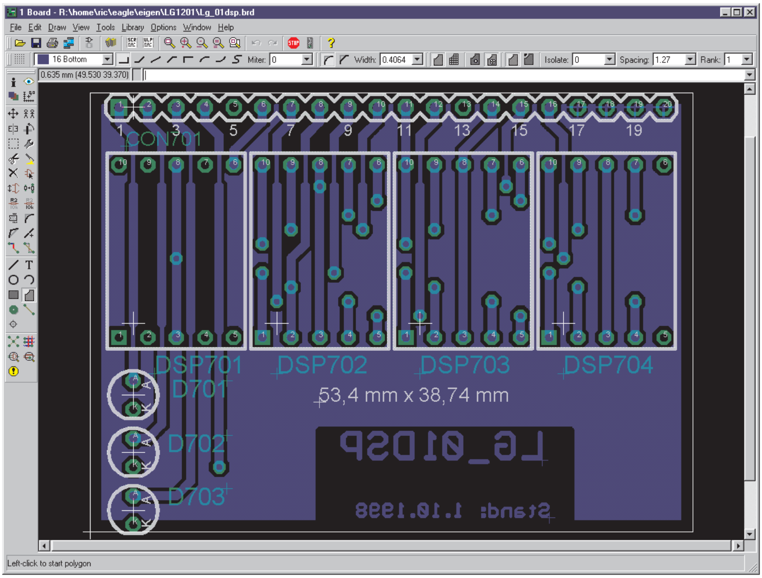
The Layout Editor window opens when you open an existing board file or create a new board. If you own the Schematic Editor you will normally draw a schematic first and then generate the board file with the BOARD command, or by clicking the *Board* icon.

The Layout Editor window appears very much like the Schematic Editor window. Even if you don’t work with the Schematic Editor, you should study the previous section, as most of the information there applies to the Layout Editor, too.

Only the commands in the command toolbar are discussed again, as some commands differ in their use.

Descriptions of commands that cannot be reached through the command toolbar are also to be found in the section concerning the Schematic Editor window.

All of the commands can also be reached through the pull-down menus in the menu bar. This also applies, of course, to the Schematic and Layout Editor windows.



1 Board - R:\home\ric\eagle\eigen\LG1201\Lg_01dsp.brd

16 Bottom Miter: 0 Width: 0.4064 Isolate: 0 Spacing: 1.27 Rank: 1

0.635 mm (49.530 39.370)

Left-click to start polygon

Layout Editor window

The Commands on the Layout Command Toolbar

INFO

Provides information about the object to be selected.

SHOW

Highlights the object to be selected.

DISPLAY

Select and deselect the layers to be displayed. Components on the top side of the board can only be selected if the layer 23, *tOrigins*, is displayed. The same applies to components on the bottom side of the board and layer 24, *bOrigins*. Only those signal layers that are defined in the Layer setup are shown.

See *Appendix* for the meaning of the layers.

Additional information about further options of the DISPLAY command can be found in the program's help.

MARK

The following mouse click defines the new (relative) origin for the coordinate display. Relative coordinates are shown additionally in the form *(*`R` *x-value y-value)*, polar values in the form *(*`P` *radius angle)*. Use the same syntax to enter relative or polar values in the command line.

If you first click the MARK icon and then the traffic-light icon, only the absolute coordinate values will be displayed again.

MOVE

Move any visible object. The right mouse button rotates the object.

The MOVE command cannot connect signals even if a wire (trace) is moved over another wire or a pad. Use ROUTE or WIRE to route signals.

Keeping the *Ctrl* key pressed while selecting an object selects it in a particular manner. Please consult the help function for details (CRICLE, ARC, WIRE, MOVE, ROUTE etc).

See Schematic Editor section for moving groups.

COPY

Copy objects.

MIRROR

Mirror objects. Components can be placed on the opposite side of the board by using the MIRROR command.

ROTATE

Rotate objects (also possible with MOVE). Keep the left mouse button pressed to rotate the selected object by moving the mouse. The parameter toolbar shows the current angle. This can be done with groups (GROUP and right mouse button)as well. Alternatively type in the angle in the *Angle* box or in the command line. More details can be found in the help function.

GROUP

Define a group which can then be moved, rotated, or copied (with CUT and PASTE) to another drawing. After the icon has been clicked, a rectangular group can be defined by holding down the left mouse button and dragging the cursor to the diagonal corner of the rectangle. If you want to define a group by a polygon, use the left mouse button to determine the corners of the polygon. Then click the right mouse button to close the polygon.
To be sure that all elements are selected DISPLAY ALL layers before.

CHANGE

Change the attributes of an object, e.g. the width of a track or the size of a text. If the *Esc* key is pressed after changing a property, the previously used value menu will appear. In this way a new value can be conveniently chosen. See the help pages.

CUT

Transfer the objects of a previously defined group into the paste buffer. See PASTE command. Not identical to the Windows cut function.

PASTE

Insert objects from the paste buffer into the drawing. Restrictions: see help. Not identical to the Windows paste function.

DELETE

Delete visible objects.

If a group has been defined, it can be deleted with the right mouse button.

DELETE SIGNALS erases all signals in the layout, provided there is no consistent schematic opened.

The DELETE command deletes an entire polygon when clicking on a polygon wire with the *Shift* key pressed.

Keeping the *Ctrl* key pressed while clicking on a wire bend will delete the bend. A new direct connection between the next bends will be drawn now.

If objects cannot be deleted, the reason can lie with error polygons related to the DRC command. They can be deleted with the ERRORS command (ERRORS CLEAR). If layer 23, *tOrigins*, or 24, *bOrigins*, is not displayed, components cannot be deleted.

ADD

Add library parts to a drawing. It offers a convenient search function. The USE command determines which libraries are available.

PINSWAP

Swap two signals connected to equivalent pads of a component, provided the pins have been defined with the same swaplevel.

REPLACE

Replace a package with another package from any library. This is permitted as long as you are working in a layout that is not connected to a consistent schematic by Forward&Back Annotation. In that case you must use CHANGE PACKAGE in order to change the package variant.

NAME

Give names to components, signals (polygons as well) or vias.

VALUE

Provide values for components. Integrated circuits normally get the IC type (e.g. 7400) as their value.

SMASH

Separates name and value texts from a component, so that they can be placed individually. The size of smashed texts can also be changed individually.
Also in combination with GROUP. All elements are smashed with a right mouse click in that case.

Keep the *Shift* key pressed while using the SMASH command in order to unsmash texts. They are not editable anymore and appear at their original positions after a window refresh.

MITER

Round off or bevel wire joints (also possible for polygon contours). The grade of mitering is determined by the miter radius. Positive sign results in a rounded joint, negative sign in a bevel.
The miter radius influences some bend modi, too (see help function: SET, *Wire_Bend*).

SPLIT

Insert a bend into a wire.

OPTIMIZE

Joins wire segments in a signal layer which lie in one straight line.

ROUTE

Route signals manually. Airwires are converted to wires.

RIPUP

Convert routed wires (tracks) into unrouted signals (airwires). Change the display of filled (calculated) polygons to outline view.

WIRE

Draw lines and arcs. If used in the layers 1 through 16, the WIRE command creates electrical connections.
The *style* parameter (CHANGE) determines the line type. The DRC and the Autorouter always treat a WIRE as a continuous line, regardless of what *style* is used.
Clicking the right mouse button changes the wire bend (SET WIRE_BEND). Please note the particularities in combination with the *Ctrl* and *Shift* key in the help function.

TEXT

Place a text string. Use CHANGE SIZE to set the height of the text. If the text is using a vector font, CHANGE RATIO will alter the thickness. CHANGE TEXT is used to alter the text itself. CHANGE FONT alters the typeface. The option *Always vector font* (*Options/User Interface*) shows and prints all texts in vector font, regardless of which font is actually set for a particular text.
It is strongly recommended to write texts in copper layers as vector font! So you can be sure that the CAM Processor's output is identical with the text shown in the Layout Editor. See also help function.

CIRCLE

Draw a circle. This command creates restricted areas for the Autorouter if used in the layers 41, *tRestrict*, 42, *bRestrict*, or 43, *vRestrict*. Circles with wire width = 0 are drawn as filled.

ARC

Draw an arc (also possible with WIRE).
CHANGE CAP FLAT | ROUND defines straight or rounded ends for arcs.
If the arc is a part of a trace and both ends are connected to a wire, caps will be round.

RECT

Draw a rectangle. This command creates restricted areas for the Autorouter if used in the layers 41, *tRestrict*, 42, *bRestrict*, or 43, *vRestrict*.

POLYGON

Draw a polygon.

Polygons in the signal layers are treated as signals. They keep an adjustable distance to objects belonging to other signals (copper pouring, flood fill). This enables you to realize different signal areas on the same layer.

The POLYGON command creates restricted areas for the Autorouter if used in the layers *tRestrict*, *bRestrict*, or *vRestrict*, .

For other possibilities of the POLYGON command see help.

VIA

Place a via-hole. Vias are placed automatically if the layer is changed during the ROUTE command. You can assign a via to a signal with the NAME command by changing it’s name to the name of the signal.

SIGNAL

Manual definition of a signal. This is not possible if the Forward&Back Annotation is active. In that case you have to define the connection with the NET command in the Schematic Editor.

HOLE

Define a mounting hole.

RATSNEST

Calculate the shortest airwires and the real mode (filled) display of polygons. The polygon calculation can be deactivated with the SET command (*Options/Set/Misc*). This command will be executed automatically for the selected signal while drawing a wire with ROUTE.

AUTO

Start Autorouter.

ERC

Perform consistency check of schematic and board.

DRC

Define Design Rules and perform Design Rule Check.

ERRORS

Show errors found by the DRC and clear error polygons.

4.4 The Library Editor Window

The Library Editor window opens when you load a library for creating or editing components. A library normally has three different elements: packages, symbols and devices.

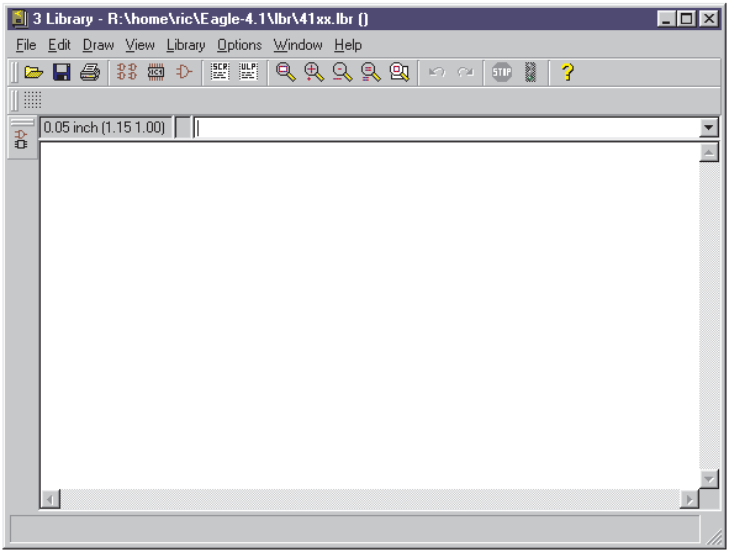
- A package is a device's housing, as will be used in the Layout Editor (on the board).
- The symbol contains the way in which the device will be shown in the schematic.
- The device represents the link between one (or more) symbol(s) and a package. Here we define the connection between a pin of a symbol and the referring pad of the package.
  We call it a Device set if the component exists in more than one package and/or technology variant.

Even if you do not have the schematic module, you can still create and edit symbols and devices.

A library need not contain only real components. Ground or supply symbols as well as drawing frames can also be stored as devices in a library. These symbols do not normally contain any pins. There are also libraries that only contain packages.

Extensive examples of the definition of library elements are to be found in a section entitled *Component Design Explained through Examples*, starting on page 157 in this manual.

When a library is loaded the following window appears first:



3 Library - R:\home\ric\Eagle-4.1\lbr\41xx.lbr ()

0.05 inch (1.15 1.00)

Library Editor window: No element has yet been loaded

Load or Rename Package, Symbol, or Device

The following commands are important for navigating within a library:

EDIT

Load device or package (if you only have the Layout Editor) for editing.

From the left: Load device, load package, load symbol. These icons are shown in the action toolbar.

REMOVE

Delete device/package/symbol from library. Available only through the *Library* menu or the command line. See help.

RENAME

Rename device/package/symbol. Available only through the *Library* menu or the command line. See help.

The Package Editing mode

The definition of a device is described briefly below. There is a more extensive guide in the *Component Design Explained through Examples* section.

The icons available in the command toolbar are equivalent to the identical icons of the Schematic or Layout Editor.

Design New Package

You change into package editing mode through the *Package* icon in the action toolbar. Type in the name of a package, and reply to the confirming question *Create new package 'packagename'?* with *yes*.

Place pads (though-hole contacts) or SMDs (SMD contact areas) with the following commands which are only available in the Package Editor.

PAD

Place the pad of a conventional component.

SMD

Place a SMD pad.

You can change the name of the pads or SMDs with the NAME command.

Use the WIRE, ARC, etc. commands to draw

- the symbol for the silkscreen on layer 21, *tPlace*,
- additional graphical information for the documentation print into layer 51, *tDocu*.

Draw restricted areas for the Autorouter, if needed, in layers 41, *tRestrict*, 42, *bRestrict*, or 43, *vRestrict*, or in layers 39, *tKeepout*, or 40, *bKeepout*, by using the commands CIRCLE, RECT, or POLYGON.

Place mounting holes with the HOLE command, if needed.

Use the TEXT command to place

- the string >NAME in layer 25, *tNames*, serving as a text variable containing the name of the component,
- the string >VALUE in layer 27, *tValues*, serving as a text variable containing the value of the component.

Use the DESCRIPTION command to add a description for the package. Rich Text format can be used for this. You will find further information in the help pages.

Design a New Package from an Existing One

Load the existing package, display all layers, define a group (GROUP command) containing all of the objects, click the CUT icon and then the origin of the drawing area (coordinates `0 0`).
You can also type the CUT command on the command line and give a reference point, for example:

```
    CUT (0 0) ;
```

Open a new package (in the same or another library) for editing, click the PASTE icon and then the origin of the drawing area.
Or enter again the PASTE command with a reference point in the command line, for example:

```
    PASTE (0 0) ;
```

Now edit the objects in whatever way you want.

You can use the coordinate values for the CUT and PASTE commands to move the group in the coordinate system. This may be useful for objects that are not drawn in the correct grid.

GROUP, CUT, and PASTE can be used to copy Packages and Symbols, but not to copy Devices.
See also EXPORT SCRIPT on page 75.

The Symbol Editing mode

Defining a symbol means defining a part of a device which can be placed individually in a schematic. In the case of a 74L00 this could be one NAND gate and the two power pins, defined as another symbol. In the case of a resistor, the device contains only one symbol which is the representation of the resistor.

You now change into symbol editing mode through the *Symbol* icon in the action toolbar. Enter the name of the symbol, and reply to the confirming question *Create new symbol 'symbolname'?* with *Yes*.

Design a New Symbol

Use the commands WIRE, ARC, etc. to draw the schematic representation of the symbol into layer 94, *Symbols*.

Place the pins by using the following PIN command, which is only available in the symbol editing mode:

PIN

Place pins.

You can adjust the pin parameters (*name, direction, function, length, visible, swaplevel*) while the PIN command is active, or with the CHANGE command. The pin parameters are explained starting on page 162 and in the help pages under the keyword PIN. Pin names are changed using the NAME command.

Use the TEXT command to place

- the string >NAME in layer 25, *tNames*, serving as a text variable containing the name of the component,
- the string >VALUE in layer 27, *tValues*, serving as a text variable containing the value of the component.

Design a New Symbol from an Existing One

Load the existing symbol, display all layers, define a group (GROUP command) containing all of the objects, click the CUT icon and then the origin of the drawing area (coordinates `0 0`).
You can also type the CUT command on the command line and give a reference point, for example:

```
    CUT (0 0) ;
```

Open a new symbol (in the same or another library) for editing, click the PASTE icon and then the origin of the drawing area.
Or enter again the PASTE command with a reference point in the command line, for example:

```
    PASTE (0 0) ;
```

Now edit the objects in whatever way you want.

You can use the coordinate values for the CUT and PASTE commands to move the group in the coordinate system. This may be useful for objects that are not drawn in the correct grid.

GROUP, CUT, and PASTE can be used to copy Packages and Symbols, but not to copy Devices.
See also EXPORT SCRIPT on page 75.

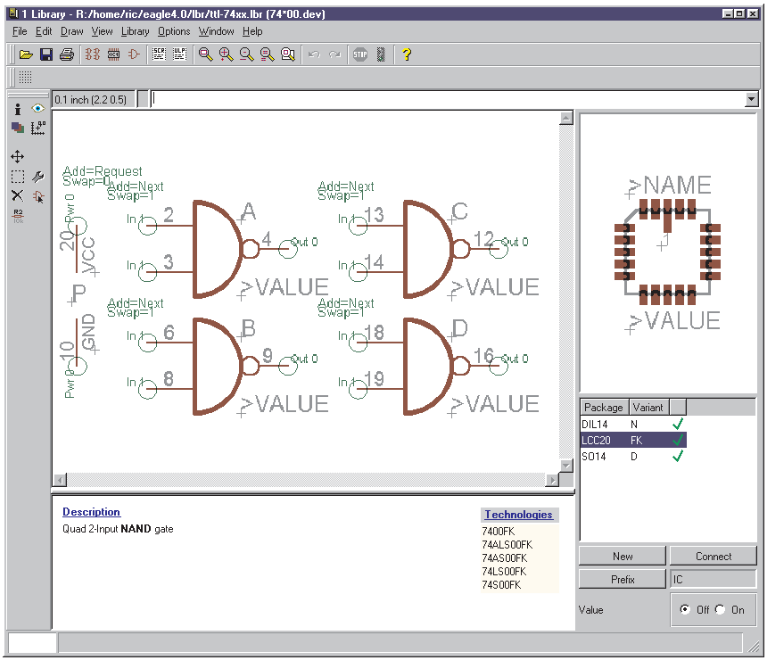
The Device Editing mode

Components are defined as devices. In the device editing mode you do not draw anything, but you define the following:

- which package variant is used,
- which symbol(s) is/are used (called gate within the device),
- which names are provided for the gates (e.g. A, B),
- which technologies are available (e.g. 74**L**00, 74**LS**00, 74**HCT**00),
- if there are equivalent gates which can be interchanged (swaplevel),
- how the gate behaves when added to a schematic (addlevel),

- the prefix for the component name, if a prefix is used,
- if the value of the component can be changed or if the value should be fixed to the device name,
- which pins relate to the pads of the package (CONNECT command)
- whether a description for this component should be stored in the library.

The following diagram shows the fully defined 7400 device with four NAND gates and a supply gate in various package and technology versions.



Package	Variant	
DIL14	N	✓
LCC20	FK	✓
SO14	D	✓

Description

Quad 2-Input **NAND** gate

Technologies

7400FK
74ALS00FK
74AS00FK
74LS00FK
74S00FK

Prefix: IC

Value: ● Off ○ On

Device Editor window

Create Actual Components from Symbols and Packages

Switch to the device editing mode by clicking the *Device* icon of the action toolbar. Type in the device name and confirm the question *Create new device 'devicename'?* with *Yes*.

Use the following commands to create a device.

ADD

Add a symbol to a device. Gate name, swaplevel, and addlevel can be defined in the ADD command in the parameter toolbar, or redefined later with the CHANGE command.

The swaplevel specifies whether there are equivalent gates.

The addlevel defines, for instance, if a gate is to be added to the schematic only on the users request. Example: the power gate of an integrated circuit which is normally not shown on the schematic.

NAME

Change gate name.

CHANGE

Change swaplevel or addlevel.

PACKAGE

Define and name package variant(s). The PACKAGE command is started by clicking on the *New* button in the Device Editor window, or by typing on the command line. Choose the requested package variant.

More information about this can be found on page 199.

CONNECT

Define which pins (gate) relate to which pads (package).

PREFIX

Provide prefix for the component name in the schematic (e.g. R for resistors).

VALUE

In the device mode, VALUE is used to specify whether the component value can be freely selected from within the schematic diagram or the layout, or whether it has a fixed specification.

On: The value can be changed from within the schematic (e.g. for resistors). The component is not fully specified until a value has been assigned.

Off: The value corresponds to the device name, including, when present, assignment of the technology and the package version (e.g. 74LS00N).

Even if the value is *Off*, the value of a component can be changed. The new value is specified after a query checks that the action is intended. If, however, the technology or the package version is altered at a later stage by means of CHANGE PACKAGE or CHANGE TECHNOLOGY, then the value of the component returns to the original value.

TECHNOLOGY

If necessary, various TECHNOLOGIES can be defined for, for example, a logic component.

DESCRIPTION

Compose a description of the device which can also be examined by the search function associated with the ADD dialog.

Information about Copying of packages, symbols and devices can be found on page 205.

4.5 The CAM Processor Dialog

Manufacturing data is generated by means of the CAM Processor. A number of drivers for the data output are available. The drivers are defined in the file *eagle.def* , which can be edited with any text editor.

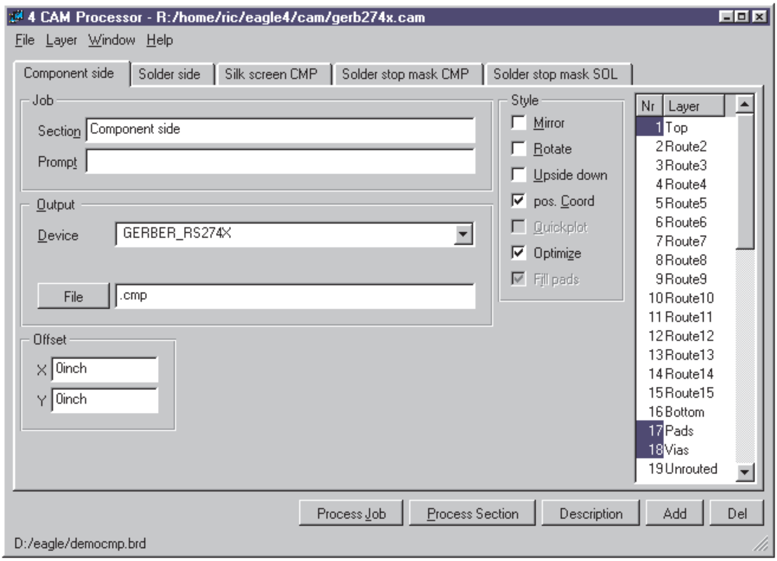
Output to matrix printers, however, is not created with the CAM Processor but with a PRINT command.

The EAGLE license conditions allow you to give the CAM Processor to your board manufacturer. To do this, you only need supply him with a copy of the EAGLE freeware.

Please, however, do not pass on your installation code. This is punishable by law.

The CAM Processor can run without support from any licensing mechanism. Alternatively, your board manufacturer can order the EAGLE freeware directly from CadSoft.

The CAM Processor can also be started directly from the command line. A number of command line parameters can be passed to it when it is called. These are listed in the appendix.



4 CAM Processor - R:/home/ric/eagle4/cam/gerb274x.cam

Section: Component side

Prompt:

Device: GERBER_RS274X

File: .cmp

X: 0inch

Y: 0inch

□ Mirror
□ Rotate
□ Upside down
☑ pos. Coord
□ Quickplot
☑ Optimize
☑ Fill pads

Nr	Layer
1	Top
2	Route2
3	Route3
4	Route4
5	Route5
6	Route6
7	Route7
8	Route8
9	Route9
10	Route10
11	Route11
12	Route12
13	Route13
14	Route14
15	Route15
16	Bottom
17	Pads
18	Vias
19	Unrouted

D:/eagle/democmp.brd

CAM Processor

Generate Data

Load Job File

A job defines the sequence of several output steps. You can, for example, use a job to generate individual files containing the Gerber data for several pcb layers.

A job is loaded with the *File* menu of the CAM Processor or via the Control Panel.

A job is not absolutely essential for output. All the settings can be made manually.

Load Board

Before you can generate an output you must open the *File* menu and load a board. At the bottom left in the CAM Processor window you can see the BRD file for which data is being generated.

Output Parameters

If a job file is loaded, the output parameters are already adjusted. A job can contain several sections with different parameter sets. The various peripheral devices accept different parameters.

If no job is loaded, set the parameters to whatever you need (see page 220).

Start Output

If you want to execute the job which has been loaded, click the *Process Job* button. If you just want to get an output using the currently visible parameter settings, click the *Process Section* button.

Define New Job

Perform the following steps to define a new job:

1. Click *Add*, to add a new section .
2. Set parameters.
3. Repeat 1. and 2. if necessary.
4. Save job with *File/Save job*.

The *Description* button allows to describe the job file. This description will be shown in the Control Panel.

The chapter on *Preparing the Manufacturing Data* contains detailed information on this subject.

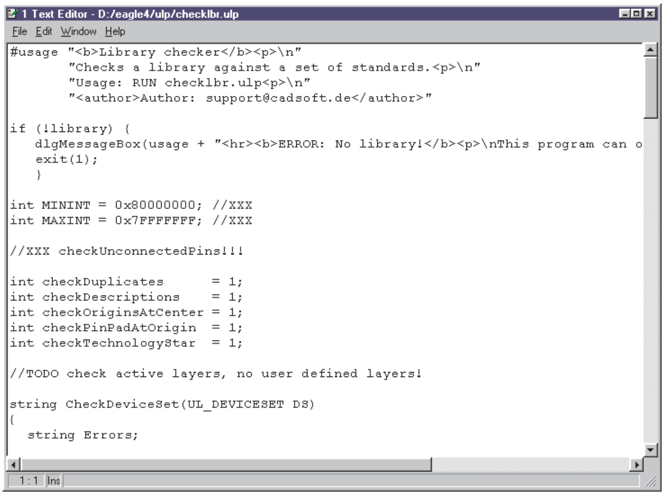
4.6 The Text Editor Window

EAGLE contains a simple text editor.

You can use it to edit script files, User Language programs or any other text file.

When in the text editor, the right mouse button calls up a context menu.

The menus bring you to a variety of functions such as commands for printing, copying and cutting, searching, replacing (also regular expressions), changing font and size, and so on.



```
#usage "<b>Library checker</b><p>\n"
       "Checks a library against a set of standards.<p>\n"
       "Usage: RUN checklbr.ulp<p>\n"
       "<author>Author: support@cadsoft.de</author>"

if (!library) {
   dlgMessageBox(usage + "<hr><b>ERROR: No library!</b><p>\nThis program can o
   exit(1);
   }

int MININT = 0x80000000; //XXX
int MAXINT = 0x7FFFFFFF; //XXX

//XXX checkUnconnectedPins!!!

int checkDuplicates      = 1;
int checkDescriptions    = 1;
int checkOriginsAtCenter = 1;
int checkPinPadAtOrigin  = 1;
int checkTechnologyStar  = 1;

//TODO check active layers, no user defined layers!

string CheckDeviceSet(UL_DEVICESET DS)
{
  string Errors;
```

The Text Editor

Chapter 5
Principles for Working with EAGLE

5.1 Command Input Alternatives

As an alternative to the mouse, the EAGLE Schematic, Layout, and Library Editors allow you enter all commands:

- on a command line by typing text commands,
- via function keys,
- via script files.
- via User Language programs.

In any case it is necessary to understand the syntax of the EAGLE command language which is described in the following section. The commands are described in detail on the help pages.

Command Line

When entering commands into the command line of an editor window you may abbreviate key words as long as they cannot be mistaken for another key word, or you may use small or capital letters (the input is not case sensitive). The following command line entries are equivalent:

```
CHANGE WIDTH 0.024
```

is equivalent to

```
cha wi 0.024
```

The actual unit for values is set in the GRID menu.

Most commands can be executed whilst declaring coordinate values in the command line.

Examples:

```
MOVE IC1 (2.50 1.75) ;
```

Element IC1 moves to position 2.50 1.75 in the layout.

```
    MIRROR U1;
```

Part U1 will be mirrored to the bottom side of the board.

```
    HOLE 0.15 (5 8.5) ;
```

Place a hole with drill diameter 0.15 at position 5 8.5.

```
    VIA 'GND' 0.070 round (2.0 3.0) ;
```

A round shaped via with a diameter of 0.070 belonging to signal GND will be placed at position 2.0 3.0.

History Function

You can recall the most recently entered commands by pressing Crsr-Up (↑) or Crsr-Down (↓) and edit them. The *Esc* key deletes the contents of the command line.

Function Keys

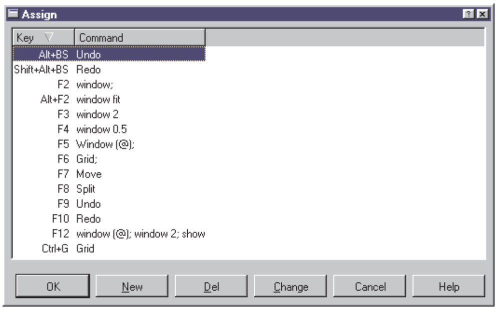
Texts may be allocated to the function keys and to combinations of those keys with Alt, Ctrl and Shift, if not occupied by the operating system or a Linux Window Manager (for example F1 for help). If a function key is pressed, this corresponds to the text being typed in via the keyboard. Since every command is capable of being entered as text, every command, together with certain parameters, can be assigned to a function key. Even whole sequences of commands can be assigned to a function key in this way.

The command

```
    ASSIGN
```

displays the current function key assignments. Changes to the key assignments can be carried out in the assign window.
The *New* button can be used to define a new key assignment. A click on *Del* will delete a marked entry, while *Change* alters an existing definition. *OK* closes the dialog and saves the definitions, while *Cancel* aborts the dialog.

These settings can also be made via the *Options/Assign* menu in the Schematic or Layout Editor.



Key	Command
Alt+BS	Undo
Shift+Alt+BS	Redo
F2	window;
Alt+F2	window fit
F3	window 2
F4	window 0.5
F5	Window (@);
F6	Grid;
F7	Move
F8	Split
F9	Undo
F10	Redo
F12	window (@); window 2; show
Ctrl+G	Grid

The dialog for the ASSIGN command

To predefine certain assignments you can also use the ASSIGN command in the file *eagle.scr* (see page 83).

Examples:

The combination of Ctrl + Shift + G displays a grid of 0.127mm:

```
    ASSIGN CS+G 'GRID MM 0.127 ON;';
```

The combination of Alt + F6 changes the layer to Top and starts the ROUTE command:

```
    ASSIGN A+F6 'LAYER TOP; ROUTE';
```

The combination of the keys Alt + R displays only the layers Top, Pads, Vias and Dimension first and then starts the print out with the default printer:

```
    ASSIGN A+R 'DISPLAY NONE 1 17 18 20; PRINT;';
```

The combination of Alt + 0 brings the Control Panel into the foreground. The combinations Alt + 1 upto 9 are assigned to the various editor windows, according to the window number which is shown in the respective title bar.

Script Files

Script files are a powerful tool. They can contain long sequences of commands, such as the specification of specific colors and fill-patterns for all layers. On the other hand they might contain netlists converted from the data of other programs.

The SCRIPT command is used to execute script files.

Many User Language programs (ULP) create script files that can be read in order to modify a layout or a schematic.

EAGLE outputs an entire library, for instance, as a script file with the aid of the EXPORT command (*Netscript* option). This file can be modified with a text editor, after which it can be read in again. This allows changes to be made to a library quite easily.

There is more information about script files and export commands later in this chapter.

Mixed Input

The various methods of giving commands can be mixed together.

You can, for instance, click the icon for the CIRCLE command (which corresponds to typing `CIRCLE` on the command line), and then type the coordinates of the center of the circle and of a point on the circumference in this form

```
(2 2) (2 3) ←
```

on the command line.

The values used above would, if the unit is currently set to *inch*, result in a circle with a radius of one inch centered on the coordinate (2 2). It is irrelevant whether the CIRCLE command is entered by icon or by typing on the command line.

5.2 The EAGLE Command Language

You only need a knowledge of the EAGLE command language if you want to make use of the alternative input methods discussed in the previous section.

The syntax of the EAGLE command language will be discussed in this section, and typographical conventions, which are important for understanding the descriptions, will be specified.

Typographical Conventions

Enter key and Semicolon

If EAGLE commands are entered via the command line they are finished with the *Enter* key. In some cases a command must have a semicolon at the end, so that EAGLE knows that there are no more parameters. It is a good idea to close all commands in a script file with a semicolon.

The use of the *Enter* key is symbolized at many places within this handbook with the ← sign.

However in the following examples neither the *Enter* key sign nor the semicolon are shown, since all of these commands can be used both on the command line and within script files.

Bold Type or Upper Case

Commands and parameters shown here in UPPER CASE are entered directly. When they are entered, there is no distinction made between upper and lower case. For example:

```
Syntax:
  GRID LINES
Input:
  GRID LINES
  grid lines
```

Lower Case

Parameters shown here in lower case are to be replaced by names, numbers or keywords. For example:

Syntax:

```
GRID grid_size grid_multiple
```

Input:

```
GRID 1 10
```

This sets the grid to 1 mm (assuming that the current unit is set to mm). Every tenth grid line is visible. The figures 1 and 10 are placed into the command instead of the placeholders *grid_size* and *grid_multiple*.

Underscore

In the names of parameters and keywords the underscore sign is often used in the interests of a clearer representation. Please do not confuse it with an empty space. As can be seen in the example above, *grid_size* is a single parameter, as is *grid_multiple*.

If a keyword contains an underscore sign, such as COLOR_LAYER does in the command

```
SET COLOR_LAYER layer_name color_word
```

then the character is to be typed in just like any other. For example:

```
SET COLOR_LAYER BOTTOM BLUE
```

Spaces

Wherever a space is permissible, any number of spaces can be used.

Alternative Parameters

The | character means that the parameters are alternatives. For example:

Syntax:

```
SET BEEP ON|OFF
```

Input:

```
SET BEEP ON
```

or

```
SET BEEP OFF
```

The beep, which is triggered by certain actions, is switched on or off.

Repetition Points

The .. characters mean either that the function can be executed multiple times, or that multiple parameters of the same type are allowed. For example:

Syntax:

```
DISPLAY option layer_name..
```

Input:

```
DISPLAY TOP PINS VIAS
```

The layer number can alternatively be used:

```
DISPLAY 1 17 18
```

More than one layer is made visible here.

If a layer (in this case Bottom) is to be hidden:

```
DISPLAY -16
```

Mouse Click

The following sign

•

usually means that at this point in the command an object is to be clicked with the left mouse button. For example:

Syntax:

```
MOVE • •..
```

Input:

```
MOVE ← (or click the icon)
Mouse click on the first element to be moved
Mouse click on the destination
Mouse click on the second element to be moved
and so on.
```

You can also see from these examples how the repetition points are to be understood in the context of mouse clicks.

Entering Coordinates as Text

The program sees every mouse click as a pair of coordinates. If it is desired to enter commands in text form on the command line, then instead of clicking with the mouse it is possible to enter the coordinates through the keyboard in the following form:

(x y)

where x and y are numbers representing units as selected by the GRID command. The textual input method is necessary in particular for script files.

The coordinates of the current cursor position can be fetched with `(@)`. For example:

```
WINDOW (@);
```

Examples of coordinate entry in text form:

You want to enter the outline of a circuit board with precise dimensions.

```
GRID MM 1;
LAYER DIMENSION;
WIRE 0 (0 0) (160 0) (160 100) (0 100) (0 0);
GRID LAST;
```

The first step is to switch to a 1 mm grid. The dimension layer is then activated. The WIRE command then first sets the line width to 0 and draws a rectangle with the aid of the four given coordinates. The last command returns the grid to whatever had previously been selected, since circuit boards are usually designed using inches.

Relative values:
It is possible to use relative coordinate values in the form (R x y) which refer to a reference point set with the MARK command before. If you don't set a reference point the absolute origin of the coordinate system will be taken.

Setting a via relative to the reference point:

```
GRID MM 0.5;
MARK (20 10);
VIA (R 5 12.5);
MARK;
```

First the grid is set to Millimeters, then the reference point at the position (20 10) is placed. The via is located at a distance of 5 mm in x and 12.5 mm in y direction from the this point. Then the reference point is removed.

Polar values:
Polar coordinates are given in the form of (P radius angle).

```
GRID MM;
MARK (12.5 7.125);
LAYER 21;
CIRCLE (R 0 0) (R 0 14.5);
PAD (P 40 0);
PAD (P 40 120);
PAD (P 40 240);
```

This examples shows how to set the reference point at position (12.5 7.125). Then a circle with a radius of 40 mm is drawn in layer 21, tPlace. Three pads are placed on the circumference with an angle of 120°.

Right Mouse Click:
The > character within parenthesis represents a right mouse click. That way one can move a whole group, for example:

```
MOVE (> 0 0) (10 0);
```

The previously selected group will be moved 10 units in x direction.

Modifier:
Within parenthesis one can use some modifiers. For combinations you don't have to care about the order:

A	represents the pressed *Alt* key, the alternativ grid
C	represents the pressed *Ctrl* key
S	represents the pressed *Shift* key
R	relativ coordinates
P	polar coordinates
>	right mouse click

C and S cause miscellaneous commands to behave in different manners. More information can be found in the help function of the respective command.

If the commands are being read from a script file, each one must be closed with a semicolon. In the above cases the semicolons can be omitted if the commands are being entered via the keyboard and each is being closed with the *Enter* key.

A component in the Layout Editor is to be transposed to a specified position.

```
GRID MM 1;
MOVE IC1 (120 25) ;
```

Alternatively:

```
MOVE (0.127 2.54) (120 25);
```

IC1 is located at coordinates (0.127 2.54) and is moved to position (120 25). The current position of a device can be obtained with the aid of the INFO command.

When a symbol is defined, a pin is placed at a certain position.

```
    PIN 'GND' PWR NONE SHORT R180 (0.2 0.4) ;
```

You draw a rectangular forbidden area in layer 41 *tRestrict*:

```
    LAYER TRESTRICT;
    RECT (0.5 0.5) (2.5 4) ;
```

5.3 Grids and the Current Unit

EAGLE performs its internal calculations using a basic grid size of 1/10 000 mm (0.1 micron). Any multiple of that can be set as the working grid.

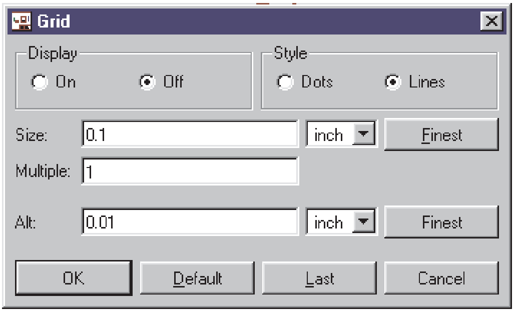
Microns, mils, inches and mm can be used for the unit. The current unit as set with the GRID command applies to all values.

You should always use the pre-set 0.1 inch grid for schematic diagrams. This grid should also be used when defining schematic diagram symbols in the Library Editor.

When starting the design of circuit boards or libraries it pays to give prior thought to the question of which grid size (or sizes) will be used as a basis. For example, it is only the origin of a package that will be pulled onto the board's placement grid. All other objects constituting the package (such as pads) are placed relative to that point on the board, just as it was defined in the library.

The basic rule for boards is: always make the grid as big as possible and as small as necessary.

Various grid sizes can be preset in the *eagle.scr* file for different types of editor windows (see page 79).



Display: ○ On ● Off

Style: ○ Dots ● Lines

Size: 0.1 inch

Multiple: 1

Alt: 0.01 inch

The Grid Menu

The current grid *Size* is set in the grid menu. The units chosen in the combo box are used.
The *Multiple* option indicates how many grid lines are displayed. If, for instance, the value 5 is entered at *Multiple*, every fifth line will be displayed.

The *Alt* line allows to set an alternative grid which can be activated by pressing the *Alt* key (while, for example, MOVE, ROUTE, ADD, or WIRE is active). This can be very useful for placing parts in a dense layout or arranging labels in the schematic. If you decide not to place it in the alternative grid and release the *Alt* key before placing it, the object stays in its origin grid.

Style specifies the way it is displayed: *Lines* or *Dots.*
The options *On* and *Off* under *Display* switch the grid display on or off.

Finest sets the finest grid that is possible.
Clicking on *default* will select the editor's standard grid.
The *last* button returns to whatever grid was previously set.

Beginning with a certain zooming limit, grid lines are not displayed anymore. This limit can be set in the menu *Options/Set/Misc, Min. visible grid size*.

Grid lines and grid dots can have any color. Click the colored button of the respective palette (depends on the background color) in the menu *Options/Set/Colors* and select the color as requested. This can also be done in the command line, for example:

```
SET COLOR_GRID BLUE
```

Instead of the color name the color number can be given, as well. It can be in the range 0 .. 63. The shown color depends on the (self-)defined colors of the current palette.

The *Redraw* option causes EAGLE to redraw grid lines/dots whenever changes concerning the grid are made in the editor window. This is not necessary for a black background.

See also the hints concerning *Color settings* on page 80.

5.4 Names and Automatic Naming

Length

Names in EAGLE can have any desired length. There is no limit.

Forbidden Characters

No names may contain spaces or umlauts. Quotation marks, semicolons, and other exotic characters (above 127 in the ASCII table) should be avoided as far as possible.

Device names must not contain either question marks or asterisks, since these characters are used as placeholders for package variants and technologies.

Commas must be avoided in pad names.

Part-bus names must not contain colons, commas or square brackets.

Automatic Naming

If a name is given together with one of the commands PIN, PAD, SMD, NET, BUS or ADD, then other names will be derived from it as long as the command is still active.

The name is simply typed into the command line before placing the element (while it is attached to the mouse). Note that the name must be placed within simple quotation marks. Entry is completed with ← .

The examples illustrate how automatic naming functions:

```
    ADD DIL14 'U1' ←  • • •
```

fetches three DIL14 elements to the board and names them U1, U2 and U3 (• corresponds to a mouse click).

```
    PAD OCT '1' ←  • • • •
```

places four octagonal pads with the names 1, 2, 3, and 4.

If the name consists of only one character from A...Z, then the following objects receive the following letters of the alphabet as names:

```
    ADD NAND 'A' ← • • • •
```

fetches four NAND gates with the names A, B, C and D. If the generated name reaches Z, then names with the default prefix will again be generated (e.g. G$1).

5.5 Import and Export of Data

EAGLE provides a number of tools for data exchange.

- Script files for importing
- The export command for exporting
- EAGLE User Language programs for import and export.

The User Language is very flexible, but does call for a suitable program to be created. You will find further details in the section on *The EAGLE User Language*.

Script Files and Data Import

The script command makes a universal tool available to the EAGLE user for data import.

Since every EAGLE operation can be carried out with the aid of text commands, you can import all types of data with the aid of a script file. A script file can in turn call other script files.

Script files can be created with a simple text editor. The prerequisite for the development of your own script files is that you understand the EAGLE command language. You will find the precise functioning and the syntax of the individual commands in the EAGLE help pages.

The file *euro.scr* in the *eagle/scr* directory, which draws the outline of a eurocard with corner limits, provides a simple example.

If a netlist is to be imported into a board design which already contains the appropriate components, then a script file of the following form is necessary:

```
SIGNAL GND IC1 7 IC2 7 J4 22;
SIGNAL VCC IC1 14 IC2 14 J4 1;
```

A *Netscript* of this sort can easily be created from the schematic diagram by the EXPORT command.

You will get a further impression of the power of importing, if you output a library with the EXPORT command (*Script* option). The script file that is generated provides an instructive example for the syntax of the script language. It can be examined with any text editor. If SCRIPT is then used to read this file into an empty library, a new library file will be created.

Comments can be included following a #-character.

The execution of a script file can be stopped by clicking the Stop icon.

File Export Using the EXPORT Command

The EXPORT command has the following modes:

DIRECTORY

Outputs a list of the contents (devices, symbols, and packages) of the currently loaded library.

NETLIST

Outputs a netlist for the currently loaded schematic or board in an EAGLE-specific format. It can be used to check the connections in a drawing. Use a ULP to obtain other formats.

NETSCRIPT

Outputs a netlist of the currently loaded schematic in the form of a script file. The netscript can be read into the board file with the help of the SCRIPT command.
This could be possibly suggestive if there are differences in the netlist between schematic and layout. In this case export the netscript from the schematic, delete all signals in the layout with DELETE SIGNALS (all traces are lost!), and import the netscript with the SCRIPT command.

PARTLIST

Outputs a component list for the schematic or board.

PINLIST

Outputs a pin/pad list for the schematic or board, listing the connected nets.

SCRIPT

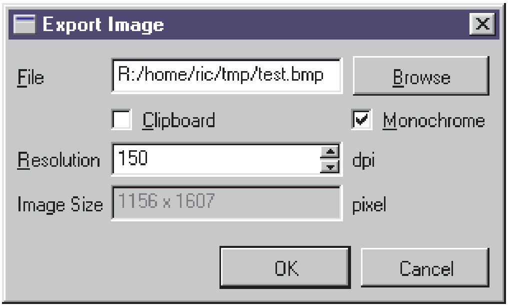
Outputs the currently loaded library in the form of a script file.
This script can be modified with a text editor in order to generate, for example, a user defined library, or to copy parts of one library into another. The modified script file can be imported into a new or an already existing library with the help of the SCRIPT command.
The script file also serves as a good example for the EAGLE command syntax.
In order to avoid loss of precision the grid unit in the script file is set to Millimeters.

IMAGE

The option *Image* allows you to generate files in various graphic formats.

The following formats are available:

bmp	Windows Bitmap file
png	Portable Network Graphics file
pbm	Portable Bitmap file
pgm	Portable Grayscale Bitmap file
ppm	Portable Pixelmap file
tif	Tag Image file
xbm	X Bitmap file
xpm	X Pixmap file



Export Image

File: R:/home/ric/tmp/test.bmp

□ Clipboard ☑ Monochrome

Resolution: 150 dpi

Image Size: 1156 x 1607 pixel

Settings for graphic file output

Click the Browse button, select the output path, and type in the graphic file name with its extension. The file extension determines the graphic file type.

To generate a black and white image activate the option *Monochrome*. To make the image available via the system's clipboard set the *Clipboard* option. The *Resolution* can be set in dots per inch. The resulting *Image Size* will be shown in the lowest field.

Further graphic formats, like HPGL, Postscript (PS), or Encapsulated Postscript (EPS), can be generated with the help of the CAM Processor.
The User Language Program dxf.ulp *generates DXF data*.

5.6 The EAGLE User Language

EAGLE contains an interpreter for a C-like User Language. It can be used to access any EAGLE file. Since version 4 it has also been able to access external data. It is possible, with very few restrictions, to export data from EAGLE, import a wide range of data into EAGLE, or to manipulate data within EAGLE. ULPs can, for example, manipulate a layout file or a library directly. The manipulation may be more transparent if a script file is first created and read. The script file contains the appropriate commands for the data manipulation.

The program examples included (**.ulp*) will provide some insight into the capacity of the User Language. They are located in your installation's ULP directory. A description of the way in which a ULP works is located in the file header. This is also displayed in the Control Panel or in the usage box when the program is called. User Language programs must be written in a text editor that does not add any control codes.

A ULP is started with the RUN command, or by dragging a ULP from the Control Panel into an editor window (Drag&Drop). To cancel the execution of an ULP click the Stop icon.

EAGLE prompts a message in the status bar, *Run: finished*, if the User Language program has been ended.

The language is described in detail in the EAGLE help pages, under the keyword User Language.

Typical applications for ULPs:

- Creating parts lists in various formats.
  See also page 230.
- Output in graphical formats.
- Data output for component insertion machines, in-circuit testers etc.
- Linking to an external database.
- Manipulation of the silk screen print, the solder stop mask, and so on.

A lot of valuable ULPs can be found on our web pages. Take a look at **http://www.cadsoftusa.com/download.htm**.

5.7 Forward&Back Annotation

A schematic file and the associated board file are logically linked by automatic Forward&Back Annotation. This ensures that the schematic and the board are always consistent

As soon as a layout is created with the BOARD command (or by means of the *Board* icon), the two files are consistent. Every action performed on the schematic diagram is simultaneously executed in the layout. If, for instance, you place a new device, the associated housing will appear on the layout at the edge of the board. If a net is placed, the signal lines are simultaneously drawn in the layout. Certain operations such as the placement or deletion of signals are only allowed in the schematic. The Layout Editor does not permit these actions, and issues an appropriate warning. Renaming devices or changing their values, for example, are permitted in both files.

The EAGLE help pages contain a closer description of the technical details.

It is not necessary for you, as the user, to pay any further attention to this mechanism. You only have to ensure that you do not work on a schematic whose associated board file has been closed, and vice versa. This means that both files must always be loaded at the same time. Otherwise they loose consistency, and the annotation can no longer work.

If you have, however, once edited the board or the schematic separately, the electrical rule check (ERC) will check the files for consistency when they are loaded. If inconsistencies are found, the ERC opens an error file with appropriate messages about the schematic diagram and the layout.

Correct the layout or the schematic diagram in accordance with the error messages in the ERC file. Run the ERC frequently to check progress, until the two files are consistent again. The message at the end of the ERC file must be:

Board and schematic are consistent.

Then save the files.

5.8 Configuring EAGLE Individually

There are a number of settings that permit the program to be adjusted for individual needs. We distinguish between program, user and project-specific settings.

Basic program settings that will apply to every user and every new project are made in the ***eagle.scr*** file. Under Windows, personal preferences are stored in the file ***eaglerc.usr***, or, under Linux, in ***~/.eaglerc***.
EAGLE remembers settings that only apply to one particular project in the ***eagle.epf*** project file.

Values that, for instance, only apply to one specific board, such as the Design Rules, special layer colors, unique newly defined layers or the grid setting are stored directly in the layout file. This also applies, of course, to schematic diagram and library files.

Configuration Commands

Most of the options are usually set by means of the *Options* menus of the individual EAGLE editor windows.

The Control Panel allows settings to be made for *Directories*, file *Backup* and the appearance of the editor window (*User interface*). These options are described in the chapter on the *Control Panel* under the *Options menu* heading, starting on page 30.

Through the *User interface* settings it is possible to select the icon-based menu or a configurable text menu.
The MENU command allows the text menu to be given a hierarchical configuration by means of a script file. There is an example of this in the appendix.

The *Options* menu in the editor windows for schematic diagrams, layouts and libraries contains, in addition to the *User interface* item, two further entries:
Assign and *Set*.

The ASSIGN command alters and displays the assignment of the function keys. You will find information about this on page 64.

General system parameters are altered with the SET command.

The CHANGE command allows a variety of initial settings for object properties.

The GRID command sets the grid size and the current unit. Further information about this starts on page 71.

SET Command

Most common options of the SET command are available in the *Settings* window of the menu *Options/Set*. This window can be reached also by entering

```
SET
```

on the command line.

The number of available layers shown in the DISPLAY or LAYER menu can be set with the option *Used_Layers*. That way it is possible to hide unused layers for clarity reasons.

```
SET USED_LAYERS 1 16 17 18 19 20 21 23 25 27
29 31 44 45 51;
```

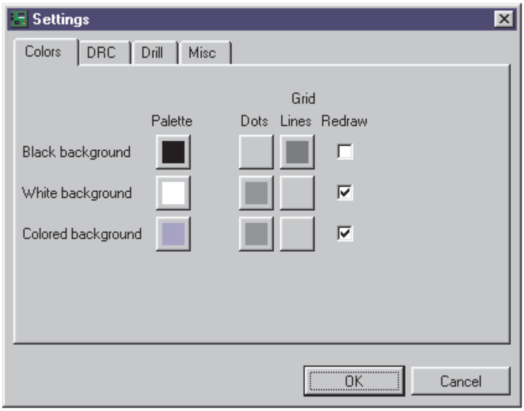
stored in the file *eagle.scr* shows only the mentioned layers. After

```
SET USED_LAYERS ALL;
```

all layers are available again.

Color Settings

The *Colors* tab contains settings for layer and background colors and colors for grid lines or dots.



		Grid		
	Palette	Dots	Lines	Redraw
Black background	■			☐
White background	□			☑
Colored background				☑

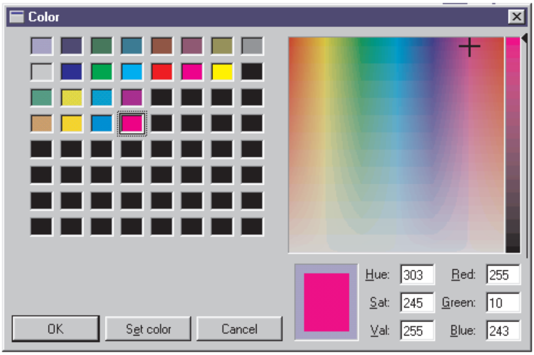
Settings Window: Color Settings

Three color palettes are available: for black, white and colored background. Each palette allows a maximum of 64 color entries, which can be given any RGB value.
By default EAGLE uses the first 16 values (entries 0..15). Eight colors followed by further eight so-called highlight colors.
The first entry of the palette determines the background color. In the white palette, however, it is not possible to change the background color because it's needed for print-outs, which normally are made on white paper.

The image above shows three buttons in the *Palette* column. Click on one of them. For example, the button for *Colored Background*. The *Color* window opens now.

On the left an 8 x 8 matrix is visible that contains already 16 pre-defined colors. There are alternating eight 'normal' colors with their corresponding eight highlight colors. A color of the palette at position x can be given the corresponding highlight color at position x+8.

In order to define new values select an unused box of the matrix and adjust the new color with the help of the color selection area and the saturation bar on the right. Click *Set Color* to apply your color. Now select a new color box in the matrix and repeat the procedure for the next color. You may also enter values for *Red*, *Green*, *Blue* or *Hue*, *Sat*, *Val* directly.



Hue: 303 Red: 255
Sat: 245 Green: 10
Val: 255 Blue: 243

Color Window: Defining Colors

The image above shows four further pairs of colors (normal and corresponding highlight color vertically arranged).

In order to change the color palette for an editor window select the appropriate *Background* in the menu *Options/User Interface*.

Alternatively the color definition and change of palette can be made in a script file or in the command line.

```
SET PALETTE <index> <rgb>
```

defines a color, where the color value has to be given hexadecimal. Example:

```
SET PALETTE 0 0xFFFF00
```

sets the background color (entry 0) to yellow (corresponds to the decimal RGB value 255 255 0 which is hexadecimal FF FF 00). Hexadecimal values are marked by a leading `0x`.

To activate the black color palette type in the command line:

```
SET PALETTE BLACK
```

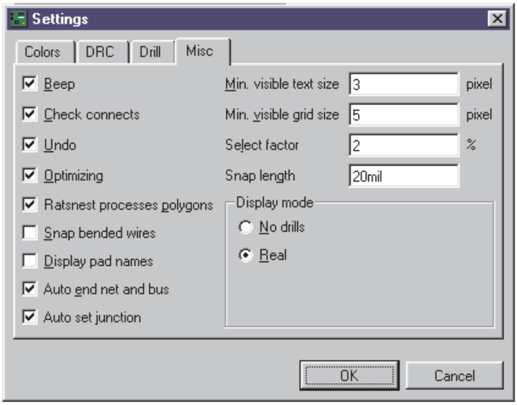
The new palette will become visible after refreshing the drawing area with the WINDOW command.

The color assignment for layers is done with the DISPLAY command or with SET COLOR_LAYER.

More details about the syntax can be found in the help function.

Miscellaneous SET Options

The *Misc* tab of the *Settings* window contains the most common options, which are switched on or off by check boxes. Some options allow entering values.



☑ Beep
☑ Check connects
☑ Undo
☑ Optimizing
☑ Ratsnest processes polygons
☐ Snap bended wires
☐ Display pad names
☑ Auto end net and bus
☑ Auto set junction

Min. visible text size 3 pixel
Min. visible grid size 5 pixel
Select factor 2 %
Snap length 20mil

Display mode
○ No drills
● Real

Settings at *Options/Set/Misc*

Options overview:

Beep:
Switches on/off the confirmation beep. Default: on.

Check connects:
Activates the package check while placing parts in the schematic. Default: on.

Undo:
Switches on/off the undo/redo buffer. Default: on.

Optimizing:
Enables the automatic removal of bends in straight lines. Default: on.

Ratsnest processes polygons:
The contents of polygons will be calculated with the RATSNEST command. Default: on.

Snap bended wires:
If active, wire bends always lie on a grid intersection point. Due to this fact it can happen that traces don't adhere 45° angles. Default: off.

Display pad names:
Pad names are displayed in the Layout or Package Editor. Default: off.

Auto end net and bus:
If placing a net on a pin or a bus the net drops from the mouse cursor. Default: on.

Auto set junction:
Ending a net on another net a junction will be set automatically. Default: on.

Min. visible text size:
Only texts with the given minimum size are displayed. Default: 3 pixels.

Min. visible grid size:
Grid lines/dots which are closer than the given minimum distance are no longer displayed on the screen. Default: 5 pixels.

Select factor:
Within this radius (given in % of the height of the current drawing window) EAGLE offers objects for selection. Default: 2%.

Snap length:
Ending a wire with the ROUTE command within this radius around a pad or smd will snap it on the middle of the pad/smd automatically. This is also true for airwires which are dynamically calculated while ROUTE is active. Default: 20 mil.

Display mode:
Pads/vias are shown with a drill hole (*real*) or without it (*No drills*). Default: real.

Changes can also be made by typing in the complete SET command. Entering

```
SET POLYGON_RATSNEST OFF
```

or, in short

```
SET POLY OFF
```

for instance switches off polygon calculation for the RATSNEST command.

The help system provides you with more instructions about the SET command.

The eagle.scr File

The script file ***eagle.scr*** is automatically executed when an editor window is opened or when a new schematic diagram, board or library file is created, unless a project file exists.

It is first looked for in the current project directory. If no file of this name exists there, it is looked for in the directory that is entered in the *Script* box in the *Options/Directories* dialog.

This file can contain all those commands that are to be carried out whenever an editor window (other than the text editor) is opened.

The *SCH*, *BRD* and *LBR* labels indicate those segments within the file which are only to be executed if the Schematic, Layout or Library Editor window is opened.

The *DEV*, *SYM* and *PAC* labels indicate those segments within the file which are only to be executed if the device, symbol or package editor mode is activated.

Commands which are defined before the first label (normally *BRD:*) are valid for all Editor windows.

If, because of the specifications in a project file, EAGLE opens one or more editor windows when it starts, it is necessary to close these and to reopen them so that the settings in ***eagle.scr*** are adopted. It is, as an alternative, possible simply to read the file ***eagle.scr*** through the SCRIPT command.

Comments can be included in a script file by preceding them with #.

Example of an eagle.scr file:

```
# This file can be used to configure the editor windows.

Assign AF3 'Window 4;';
Assign AF4 'Window 0.25;';
Assign AF7 'Grid mm;';
Assign AF8 'Grid inch;';

# net class configuration
grid mm;
Class 0 Default 0.3 0.2 0.6;
Class 1 Power 0.8 0.5 0.8;
grid default;

BRD:
#Menu Add Change Copy Delete Display Grid Group Move\
#     Name Quit Rect Route Script Show Signal Split \
#     Text Value Via Window ';'  Wire Write Edit;

Grid inch 0.05 on;
Grid alt inch 0.01;
Set Pad_names on;
Set Width_menu 0.008 0.01 0.016;
Set Drill_menu 0.024 0.032 0.040;
Set Size_menu 0.05 0.07 0.12;
Set Used_layers 1 16 17 18 19 20 21 22 23 24 25 26 \
                27 28 39 40 41 42 43 44 45;
Change width 0.01;
Change drill 0.024;
Change size 0.07;

SCH:
Grid Default;
```

```
Change Width 0.006;
#Menu Add Bus Change Copy Delete Display Gateswap \
#     Grid Group Invoke Junction Label Move Name Net \
#     Pinswap Quit Script Show Split Value Window ';' \
#     Wire Write Edit;

LBR:
#Menu Close Export Open Script Write ';' Edit;

DEV:
Grid Default;
#Menu Add Change Copy Connect Delete Display Export \
#     Grid Move Name Package Prefix Quit Script Show \
#     Value Window ';' Write Edit;

SYM:
Display all;
Grid Default On;
Change Width 0.010;

#Menu Arc Change Copy Cut Delete Display Export \
#     Grid Group Move Name Paste Pin Quit Script \
#     Show Split Text Value Window ';' Wire Write Edit;

PAC:
Grid Default On;
Grid Alt inch 0.005;
Change Width 0.005;
Change Size 0.050;
Change Smd 0.039 0.039;

#Menu Add Change Copy Delete Display Grid Group \
#     Move Name Pad Quit Script Show Smd Split Text \
#     Window ';' Wire Write Edit;
```

The eaglerc File

Under Windows, user-specific data is stored in the file ***eaglerc.usr***, or, under Linux, in ***~/.eaglerc***. This file is stored in the user's home directory. If there is no *home* environment variable set, the Windows registry entry will be taken. More about this can be found in the section *Directories* on page 31 .

It contains information about the:

- SET command (*Options/Set* menu)
- ASSIGN command (function key assignments)
- User Interface
- Currently loaded project (path)

EAGLE looks for the configuration file in various locations in the given sequence and executes them (if existing):

<prgdir>/eaglerc	(Linux and Windows)
/etc/eaglerc	(Linux only)
*$HOME/.eagler*c	(Linux)
$HOME/eaglerc.usr	(Windows)

These files should not be edited.

With the help of various User Language programs it is also possible to define settings in EAGLE. Visit our web site:
http://www.cadsoftusa.com/download.htm.

EAGLE Project File

If a new project is created (by clicking the right mouse button on an entry in the *Projects* branch of the tree view and then selecting *New/Project* in the context menu in the Control Panel), a directory is first created which has the name of the project. An ***eagle.epf*** configuration file is automatically created in every project directory.
EAGLE takes note of changes to object properties that are made with the CHANGE command during editing and the contents of the *Width*, *Diameter*, and *Size* menus in the project file.
It also contains information about the libraries *in use* for this project.
The position and contents of the active windows at the time the program is closed are also saved here. This assumes that the *Automatically save project file* option under *Options/Backup* in the Control Panel is active. This state will be recreated the next time the program starts.

Those who have previously worked on projects with **EAGLE 3.5x** can most easily adapt them to the structure that applies from Version 4.0 onwards as follows:

- Create a directory with the project name
- Copy the old project file under the name *eagle.epf* into this directory
- Copy all the files belonging to the project into this directory

The path that leads to these project directories is specified in the Control Panel under *Options/Directories/Projects*.

Chapter 6
From Schematic to Finished Board

This chapter illustrates the usual route from drawing the schematic diagram to the manually routed layout. Particular features of the schematic diagram or Layout Editor will be explained at various points. Use of the Autorouter and the output of manufacturing data will be described in subsequent chapters.

We recommend to create a project (folder) first. Details about this can be found on page 28.

6.1 Creating the Schematic Diagram

The usual procedure is as follows:

Devices are taken from existing libraries and placed on the drawing area. The connecting points (pins) on the devices are then joined by nets (electrical connections). Nets can have any name, and can be assigned to various classes. Power supply voltages are generally connected automatically. In order to document all the supply voltages in the schematic diagram it is necessary to place at least one so-called *supply symbol* for each voltage.

Schematic diagrams can consist of a number of pages (but not in the Light edition). Nets are connected across all the pages if they have the same name.

It is assumed that libraries containing the required components are available. The definition of libraries is described in its own chapter.

It is possible at any time to create a layout with the BOARD command or with the *Board* icon. As soon as a layout exists, both files must always be loaded at the same time. This is necessary for the association of the schematic diagram and the board to function. There are further instructions about this in the section on *Forward&Back Annotation*.

Open the Schematic Diagram

You first start from the Control Panel. From here you open a new or existing schematic diagram, for instance by means of the *File/Open* or the File/New menus, or with a double click on a schematic diagram file in the directory tree. The schematic diagram editor appears.

Create more schematic sheets if needed. For that purpose, open the combo box in the action toolbar with a mouse click, and select the *New* item. A new sheet will be generated (see page 37). Another way to get a second sheet is to type in

```
EDIT .S2
```

on the command line. If, however, you do not in fact want the page, it is deleted with

```
REMOVE .S2
```

the entire sheet.

Set the Grid

The grid of schematic diagrams should always be 0.1 inch, i.e. 2.54 mm. Nets and symbol connection points (pins) must lie on a common grid. Otherwise connections will **not** be created between nets and pins.

Place Symbols

First you have to make available the libraries you want to take elements from with the USE command. Only libraries which are *in use* will be recognized by the ADD command and its search function. More information concerning the USE command can be found on page 37.

Load Drawing Frame

It is helpful first to place a frame. The ADD command is used to select devices from the libraries.

When the ADD icon is clicked, the ADD dialog opens.

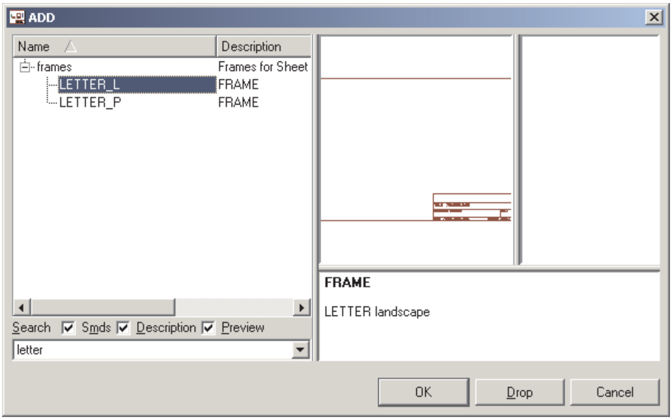
It shows all the libraries that are made available with the USE command, first. You can expand the library entries for searching elements manually or you can use the search function .

A letter format frame is to be used. Enter the search key *letter* in the *Search* line at the lower left, and press the *Enter* key. The search result shows a number of entries from *frames.lbr*. If you select one of the entries (*LETTER_P*), a preview is shown on the right, assuming that the *Preview* option is active. Disabling the option *Description* does not take into consideration the part's descriptive texts.

Clicking *OK* closes the ADD window, and you return to the schematic diagram editor. The frame is now hanging from the mouse, and it can be put down. The bottom left hand corner of the frame is usually at the coordinate origin (0 0).

Library names, device names and terms from the device description can be used as search keys. Wildcards such as * or ? are allowed. A number of search keys, separated by spaces, can be used.

In the Layout Editor's ADD dialog package names and terms from the package descriptions (*Description* enabled) can be searched for.



Name	Description
frames	Frames for Sheet
LETTER_L	FRAME
LETTER_P	FRAME

Search ☑ Smds ☑ Description ☑ Preview

letter

FRAME

LETTER landscape

ADD dialog: Results from the search key *letter*

The ADD command may also be entered via the command line or in script files. The frame can also be placed using the command:

```
add letter_p@frames.lbr
```

Wildcards like `*` and `?` may also be used in the command line. The command

```
add letter*@frames.lbr
```

for example opens the ADD windows and shows various frames in letter format to select.

The search will only examine libraries that are *in use*. That means that the library has been loaded by the USE command (*Library/Use*).

If a certain library should not be listed in the ADD dialog anymore, select the library name in the tree and click the *Drop* button. Now it is not *in use* anymore.

Place Circuit Symbols (Gates)

All further devices are found and placed by means of the mechanism described above. You decide a package variant at this early stage. It can easily be changed later if it should turn out that a different package form is used in the layout.

If you have placed a device with ADD, and then want to return to the ADD dialog in order to choose a new device, press the *Esc* key or click the ADD icon again.

Give the devices names and values (NAME, VALUE).

If the text for the name or the value is located awkwardly, separate them from the device with SMASH, and then move them to whatever position you prefer with MOVE. Clicking with DELETE on either of the texts makes it invisible. Use the *Shift* key with SMASH to get the texts at their original positions. The texts are now no longer separated from the device (unsmash). The same effect has DELETEing the remaining text.

MOVE relocates elements, and DELETE removes them. With INFO or SHOW information about an element is displayed on the screen.

ROTATE turns gates by 90°. The same can be done with a right mouse click while the MOVE command is active.

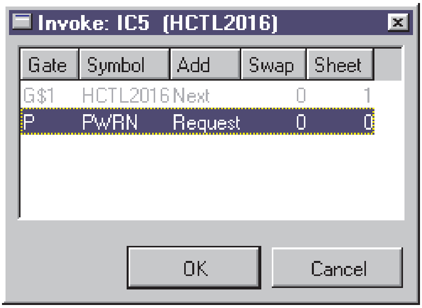
Multiple used parts may be copied with the COPY command. COPY places always a new part even if it consits of several gates and not all of them are already used.

Repeated circuitry can be reproduced in the schematic diagram with the aid of the GROUP, CUT and PASTE commands. First make sure that all the layers are made visible (DISPLAY ALL).

Hidden Supply Gates

Some devices are defined in the libraries in such a way that the power supply pins are not visible on the schematic diagram. Visibility is not necessary, since all the power pins with the same name are automatically connected, regardless of whether or not they are visible.

If you want to connect a net directly to one of the hidden pins, fetch the gate into the schematic diagram with the aid of the INVOKE command. Click on the INVOKE icon, and then on the device concerned, assuming that it is located on the same sheet of the schematic diagram. If the gate is to be placed on a different schematic diagram sheet, go to that sheet, activate INVOKE, and type the name of the device on the command line. Select the desired gate in the INVOKE window, then place it. Then join the supply gate to the desired net.



Invoke: IC5 (HCTL2016)

Gate	Symbol	Add	Swap	Sheet
G$1	HCTL2016	Next	0	1
P	PWRN	Request	0	0

INVOKE: Gate P is to be placed

Devices with Several Gates

Some devices consist not of one but of several gates. These can normally be placed onto the schematic diagram one after another with the ADD command. To place a certain gate you can use the gate name directly.

Example:

The device *74*00* from the *74xx-eu* library consists of for NAND gates named *A* to *D* and one power gate named *P*. If you want to place the gate *C* first, use the gate name with the ADD command:

```
ADD 'IC1' 'C' 74LS00@74xx-eu.lbr
```

See also help function for the ADD command.

As soon as one gate has been placed, the next one is attached to the mouse (addlevel is *Next*). Place one gate after another on the diagram. When all the gates in one device have been used, the next device is brought in.

If the gates in one device are distributed over several sheets, place them first with ADD, change to the other sheet of the schematic diagram, and type, for example

```
INVOKE IC1
```

on the command line. Select the desired gate from the INVOKE window.

Wiring the Schematic Diagram

Draw Nets (NET)

The NET command defines the connections between the pins. Nets begin and end at the connection points of a pin. This is visible when layer 93, *Pins*, is displayed (DISPLAY command).

Nets are always given an automatically generated name. This can be changed by means of the NAME command. Nets with the same name are connected to one another, regardless of whether or not they appear continuous on the drawing. This applies even when they appear on different sheets.

If a net is taken to another net, a bus, or a pin connecting point, the net line ends there and is connected. If no connection is made when the net is placed, the net line continues to be attached to the mouse. This behavior can be changed through the *Options/Set/Misc menu* (using the *Auto end net and bus* option). If this option is deactivated, a double click is needed to end a net. Nets are shown on layer 91, *Nets*.

Nets must end exactly at a pin connecting point in order to be joined. A net is not joined to the pin if it ends somewhere on the pin line.

EAGLE will inform you about the resulting net name or offer a selection of possible names if you are connecting different nets.

The JUNCTION command is used to mark connections on nets that cross one another. Junctions are placed by default. This option, (*Auto set junction*), can also be deactivated through the *Options/Set/Misc* menu.

Nets must be drawn with the NET command, not with the WIRE command.

Do not copy net lines with the COPY command. If you do this, the new net lines won't get new net names. This could result in unwanted connections.

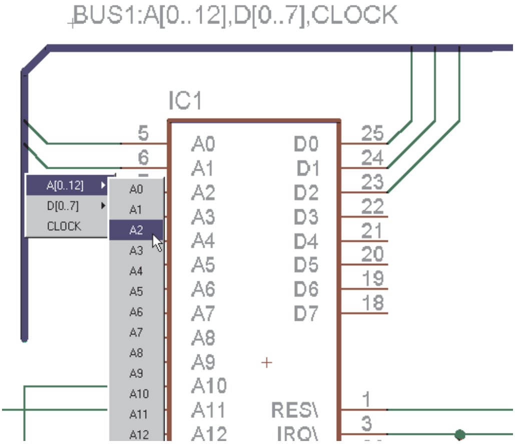
If the MOVE command is used to move a net over another net, or over a pin, no electrical connection is created. To check this, you can click the net with the SHOW command. All the connected pins and nets will be highlighted. If a gate is moved, the nets connected to it will be dragged along.

An identifier can be placed on a net with the LABEL command. Provided you have defined an alternative grid, labels can be arranged comfortably in a finer grid with the *Alt* key pressed.

Drawing a bus (BUS)

Busses receive names which determine which signals they include. A bus is a drawing element. It does not create any electrical connections. These are always created by means of the nets and their names. The associated menu function is a special feature of a bus. A menu opens if you click on the bus with NET. The contents of the menu are determined by the bus name.

The bus in the diagram is named *Bus1:A[0..12],D[0..7],Clock*.



Bus menu

Clicking on the bus line while the NET command is active, opens the menu as illustrated above. The name of the net that is to be placed is selected from here.

The index of a partial bus name may run from 0 to 511.

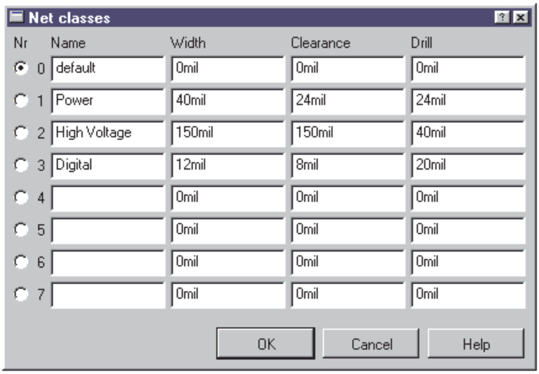
The help system provides you with more information about the BUS command.

Specifying Net Classes

The CLASS command specifies a net class (also *Edit/Net classes*) . The net class specifies the minimum track width, the minimum clearance from other signals and the hole diameter for vias in the layout for a specific class of signals. The default setting is net class 0, which means that none of these things are specified. A maximum of 8 net classes is possible.

Net classes can be changed at a later date by means of the CHANGE command (the *Class* option).

These settings can also be left until you reach the layout stage.



Nr		Name	Width	Clearance	Drill
●	0	default	0mil	0mil	0mil
○	1	Power	40mil	24mil	24mil
○	2	High Voltage	150mil	150mil	40mil
○	3	Digital	12mil	8mil	20mil
○	4		0mil	0mil	0mil
○	5		0mil	0mil	0mil
○	6		0mil	0mil	0mil
○	7		0mil	0mil	0mil

Net classes: Various settings

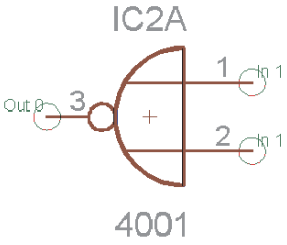
For a permanent definition of net classes, as for all projects, one can use the CLASS command in the *eagle.scr* file. See example on page 83.

Pinswap and Gateswap

Pins or gates that have the same swaplevel can be exchanged with one another. These properties are specified either when the symbol is defined (pinswap) or when the device is created (gateswap).

Provided the swaplevel of two pins is the same, they can be exchanged for one another. Display layer 93, *Pins*, in order to make the swaplevel of the pins visible.

Pins or gates may not be swapped if the swaplevel = 0.



Swaplevel: Pins layer is visible

Input pins 1 and 2 have swaplevel *1*, so they can be exchanged with one another. The output pin, 3, which has swaplevel *0*, cannot be exchanged.

You can find the swaplevel of a gate by means of the INFO command.

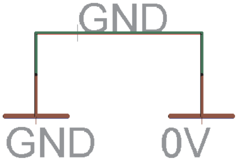
Power Supply

Pins defined as having the direction *Pwr* are automatically wired up. This is true, even if the associated power gate has not explicitly been fetched into the schematic. The name of the Pwr pin determines the name of the voltage line. This is already fixed by the definition of the symbols in the library.

If nets are connected to a device's Pwr pins, then these pins are not automatically wired. They are joined instead to the connected net.

For every Pwr-pin there must be at least one pin with the same name but the direction *Sup* (a supply pin). There must be one on every sheet. These Sup pins are fetched into the schematic in the form of power supply symbols, and are defined as devices in a library (see *supply.lbr*). These devices do not have a package, since they do not represent components. They are used to represent the supply voltages in the schematic diagram, as is required by the electrical rule check (ERC) for the purposes of its logical checks.

Various supply voltages, such as 0 V or GND, which are to have the same potential (GND, let's say), can be connected by adding the corresponding supply symbols and connecting them with a net. This net is then given the name of that potential (e.g. GND).



Supply symbols

If a supply pin (with pin direction *Sup*) is placed on a net (with ADD, MOVE) then this net segment will receive the name of the supply pin. The user will be informed if the net name changes as soon as a supply pin is connected. If the last supply pin on a net is deleted, it is given a new, automatically generated name.

Check and Correct Schematic

A schematic diagram must be checked with the aid of the Electrical Rule Check (ERC), when the design of the schematic diagram has been completed, if not before. The ERC writes all its warnings and error messages into a text file (**.erc*), and displays this file in a the text editor window.

Please check all the messages and make corrections in the schematic, if necessary. Run ERC again to check your efforts.

Please remember:
The ERC checks the schematic diagram according to a rigid set of rules. It can sometimes happen that an error message generated from your schematic can be tolerated.

If necessary, make an output of the net and pin lists (EXPORT).

SHOW allows nets to be traced in the schematic diagram.

Experience has shown that printing out the schematic diagram by means of the PRINT command is helpful when searching for faults.

In the case of a corresponding board file, the ERC also checks the consistency between schematic and board. If there are no differences, ERC reports *Board and schematic are consistent* at the end of the text file. Otherwise the message *Board and schematic are **not** consistent* will be prompted. The differences will be listed in the ERC file above.
See also page 78 for further information concerning Forward&Back Annotation.

Points to Note for the Schematic Editor

Superimposed Pins

Pins will be connected if the connection point of an unconnected pin is placed onto the connection point of another pin. Pins will not be connected if you place a pin that is already connected to a net line onto another pin.

Open Pins when MOVEing

If an element is moved then its open pins will be connected to any nets or other pins which may be present at its new location. Use UNDO if this has happened unintentionally.

Merge Different Schematics

The commands GROUP, CUT, and PASTE can be used to copy a schematic or parts of it into another one. To do this open a schematic, display all layers (DISPLAY ALL), select all objects that should be copied with GROUP, and select the group with CUT and a right mouse click. Open another schematic now and use the PASTE command.

While inserting a group EAGLE checks the objects' names and compares them with those already existing in the schematic. If there are matches the affected objects to be inserted get new unequivocal names.

This works also with board files, provided there is no consistent schematic available.

6.2 Considerations Prior to Creating a Board

Checking the Component Libraries

The EAGLE component libraries are developed by practicing engineers, and correspond closely to present-day standards. The variety of components available is, however, so wide that it is impossible to supply libraries which are suitable for every user without modification.

There are even different packages which are supplied by various manufacturers using the same identification! Manufacturers recommend very different sizes for SMD pads, and these depend again on the soldering procedure being applied.

In short: You cannot get away without checking the components, in particular the package definitions, being used when laying out.

In the case of SMD components please take particular care to ensure that the package from the library agrees with the specifications of your component. Housings from different manufacturers with the same name but different dimensions are often found.

Agreement with the Board Manufacturer

If you plan to have your pcb films made using a Gerber photoplotter, then, if you have not already done so, now is the time to inquire at your board maker whether he stipulates any particular values for the following parameters:

- track width
- shape of solder lands
- diameter of solder lands
- dimensions of SMD pads
- text size and thickness
- drill hole diameters
- number of signal layers
- in case of multilayer boards: manufacturing directions for blind and buried vias and composition of the board
  (see page 121)
- distance between different potentials
- parameters concerning solder stop mask

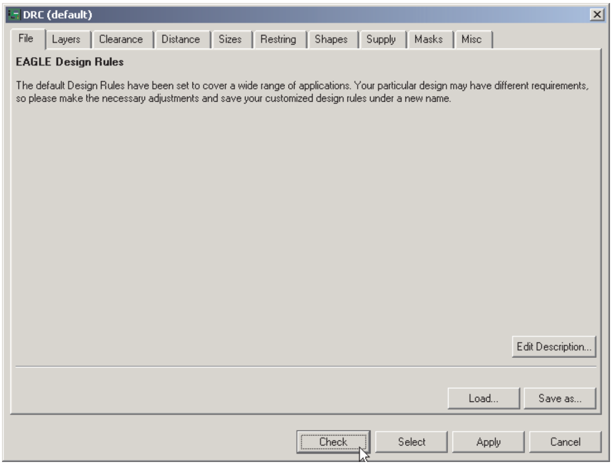
You will save yourself time and money if you take these stipulations into account in good time.

You will find more details on this in the section on the *Preparing of Manufacturing Data* (Chapter 9).

Specifying the Design Rules

All the parameters relevant to the board and its manufacture are specified in the Design Rules. Use the menu *Edit/Design Rules..* or the DRC command to open the Design Rules window. Opening the window with the DRC command allows to start the Design Rule Check with the *Check* button after setting the parameters. The layout can be checked at any time.

The following window appears when *DRC* has been called:



EAGLE Design Rules

The default Design Rules have been set to cover a wide range of applications. Your particular design may have different requirements, so please make the necessary adjustments and save your customized design rules under a new name.

DRC: Adjusting the Design Rules

General Principles

The first time that you call the DRC, the Design Rules are provided by the program. If necessary, adjust the values to suit your requirements. When you have finished the adjustments, start the error check by clicking on *OK*. This will result in the Design Rules being stored in the layout file itself.

By clicking on *Select* you specify the region of the layout that is to be examined. Simply drag a rectangle over the desired region with the mouse. The error check will then automatically start.

Only signal layers visible before the Design Rule Check has been started are examined (DISPLAY).

The *Apply* button stores the values that are currently set in the layout file. This means that the values that have so far been chosen are not lost if you do not immediately start the error check and if you want to leave the DRC dialog via the *Cancel* button.
Changes to various Design Rules are immediately displayed in the Layout Editor after clicking on *Apply*.

The Design Rules can be saved in a special Design Rules file (*.*dru*) by the use of the *Save as..* button.

To apply a set of Design Rules to a board, you can drag any *dru* file of the *Design Rules* branch of the tree view in the Control Panel into the Layout Editor window.

The *Load..* button allows any particular set of Design Rules to be loaded from a file.

Change can be used to alter the descriptive text for the current parameter set. The description usually appears in the DRC dialog (*File* tab), as can be seen in the foregoing diagram. Rich Text format can be used with the text. You will find notes on this in the help system.

Usually one sets the common Design Rules with the *Edit/Design Rules* menu first and starts the Design Rule Check when required with the DRC command. But it is also possible to adjust the Design Rules if you use the DRC command.

If you open the Design Rules dialog with the menu Edit/Design Rules.. *the Select button will not be shown. In this case* OK *ends the dialog and saves the parameters in the board file.*

If you use the DRC command, the Check *button starts the Design Rule Check.*

The Design Rules dialog offers a range of different options that can be selected through the tabs. The options include:

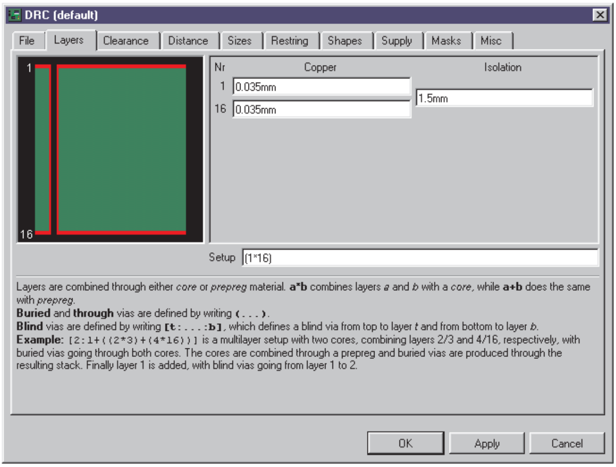
File	Manage the Design Rules
Layers	Number of copper layers, structure of multilayer boards, kind and length of vias, thickness of copper and isolation layers
Clearance	Distances between objects in the signal layers representing signals that may be different or the same
Distance	Distances from the board edge and between holes
Sizes	Minimum track width and hole diameter, particularly for micro and blind vias
Restring	Width of the remaining ring at pads and (micro) vias
Shapes	Shapes of pads and SMDs
Supply	Thermal and insulation symbols in supply layers
Mask	Values for solder stop and solder cream masks
Misc	Additional checks

Notes on the Parameters

Most parameters are explained with the help of a graphical display. As soon as you click with the mouse on a parameter line, the associated display appears.

Layers

Define the number of signal layers and the kind of vias (blind or buried vias) here. With the help of a mathematic expression in the *Setup* line the proper structure of the board — the appropriate combination of cores and prepregs and the resulting facilities for vias — can be defined.



Nr	Copper	Isolation
1	0.035mm	1.5mm
16	0.035mm	

Setup (1*16)

Layers are combined through either *core* or *prepreg* material. **a*b** combines layers *a* and *b* with a *core*, while **a+b** does the same with *prepreg*.
Buried and **through** vias are defined by writing **(...)**.
Blind vias are defined by writing **[t:...:b]**, which defines a blind via from top to layer *t* and from bottom to layer *b*.
Example: `[2:1+((2*3)+(4*16))]` is a multilayer setup with two cores, combining layers 2/3 and 4/16, respectively, with buried vias going through both cores. The cores are combined through a prepreg and buried vias are produced through the resulting stack. Finally layer 1 is added, with blind vias going from layer 1 to 2.

Design Rules: Layer Setup

In most cases (for simple two or more layer boards) the vias are drilled through all layers. The image above shows the default setup for a two layer board. The expression `(1*16)` defines one core with layers 1 and 16, which can be connected with vias. Parenthesis around the expression define through-hole (continuous) vias.

Basic examples:

1 layer:
`16` Only layer 16, no vias.

4 layers, vias through all layers:
`(1*2+15*16)` Two cores are affiliated with each other.

6 layers, vias through all layers:
`(1*2+3*14+15*16)` Three cores are affiliated with each other.

The fields for *Copper* and *Isolation* are used to define the thickness of copper and isolation layers. These settings are only relevant for complex multilayer boards that use blind or micro vias.

The commands DISPLAY, LAYER, WIRE, and ROUTE work only with those signal layers defined in the Layer setup.

Further information and examples about the *Layer setup* can be found in the section *Multilayer Boards* beginning with page 121.

Loading a board file that was made with an older version causes EAGLE to check which signal layers contain wires. These layers appear in the layer setup. Please adjust it if necessary.

Minimum Clearances

Clearance refers to the minimum distances between tracks, pads, SMDs and vias of different signals, and between SMDs, pads and vias of the same signal.
Setting the values for *Same signal* checks to 0, disables the respective check.

Distance allows settings to be made for the minimum distances between objects in layer 20, *Dimension*, in which the board outline is usually drawn, and between holes.

Setting the value for *Copper/Dimension* to *0* switches off the minimum clearance check between copper and dimension. In this case EAGLE does not recognize holes that are placed on wires.
Polygons don't keep their distance to objects in layer 20, Dimension, either!

Nets that belong to special net classes are an exception here. Values for minimum clearance and hole diameters defined by means of the CLASS command apply to these, provided they are higher than those given in the Design Rules.

EAGLE's minimum resolution is 1/10.000 of a millimeter. That is why the DRC can report only distance and clearance errors which are larger than 1/10.000 mm.

Sizes

The minimum values for track width and for hole diameter allowed in the layout are selected here.

Here you set the aspect ratio of drill depth to drill diameter for boards that contain blind vias. Please contact your boardhouse for this information! If the boardhouse specifies, for example, an aspect ratio of `1:0.5` you have to enter the value `0.5` in the line *Min. Blind Via Ratio*.

For micro vias you have to set the minimum drill diameter in the line *Min. MicroVia*. Setting this value higher than the value in *Minimum Drill* means that there are no micro vias used (default).

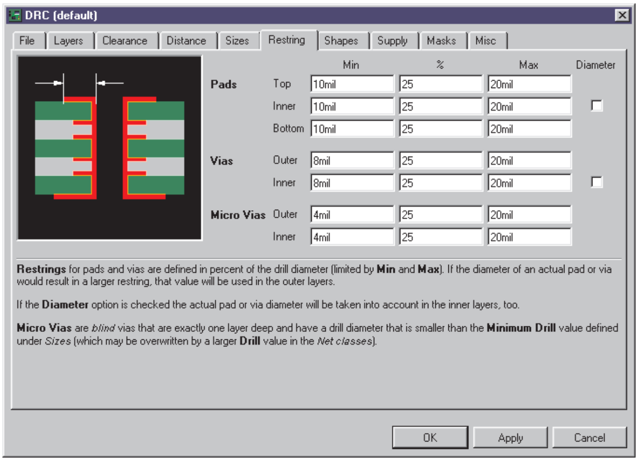
Restring

The settings made under *Restring* determine the width of the ring remaining at pads, vias, and micro vias. The remaining ring refers to the ring of copper that remains around a hole after a pad or via has been drilled. Different selections can be made for the width of the remaining ring in the inner and outer layers. Pads may also differ between the *Top* and *Bottom* layers.

Usually the value is expressed as a percentage of the hole diameter. Minimum and maximum values can additionally be specified.

As soon as you change a parameter and click on the *Apply* button you can directly see the effects in the layout. If you want to use different values for the upper and lower layer (or different shapes, see *Shapes* tab), it is recommended to set the layer color for layers 17, *Pads*, and 18, *Vias*, the same as the background color (black or white). In this case you can recognize the real size and shape of the pad/via in its respective layer.

The INFO command shows the via diameter separately for outer and inner layers, and additionally the initially user-defined value.



		Min	%	Max	Diameter
Pads	Top	10mil	25	20mil	
	Inner	10mil	25	20mil	☐
	Bottom	10mil	25	20mil	
Vias	Outer	8mil	25	20mil	
	Inner	8mil	25	20mil	☐
Micro Vias	Outer	4mil	25	20mil	
	Inner	4mil	25	20mil	

Restrings for pads and vias are defined in percent of the drill diameter (limited by **Min** and **Max**). If the diameter of an actual pad or via would result in a larger restring, that value will be used in the outer layers.

If the **Diameter** option is checked the actual pad or via diameter will be taken into account in the inner layers, too.

Micro Vias are *blind* vias that are exactly one layer deep and have a drill diameter that is smaller than the **Minimum Drill** value defined under *Sizes* (which may be overwritten by a larger **Drill** value in the *Net classes*).

Design Rules: Restring setting

The diagram illustrates the template for setting the width of the residual ring. The standard value for the restring around holes is 25% of the hole diameter. Since the width of the ring on small holes specified this way would soon fall below a technically feasible value, a minimum value (here: 10 mil for pads, 8 mil for vias, 4 mil for micro vias) is specified here. It is also possible to specify a maximum value.

Example:
The ring around a hole with 40 mil diameter is 10 mil (25%). It therefore lies in between the maximum and minimum values.
If the hole is only 24 mil in diameter (e.g. for a via), the calculation

yields a restring value of only 6 mil. For a board made in standard technology this is extremely fine, and cannot easily be made. It might well involve extra costs. In this case a minimum value of 10 mil is given.

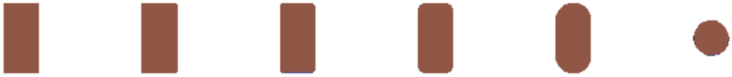
If you like to define a restring with a fixed width, use the same value for minimum and maximum. The value in percent has no effect in this case.

Set the check boxes *Diameter*, and the via/pad diameter will be taken into consideration for the inner layers. This can be of interest if a predefined pad or via diameter exceeds the calculated value. Otherwise the pad or via in the inner layers would be smaller than in the outer layers. If you want pads/vias to have the same diameter in all layers, set the option *Diameter.*
The option is set off, by default, for new created boards, but will be set on for boards that are updated from version 3.5 or prior because in these versions pads and vias had the same diameter in all layers. Thus the update process does not change the original layout.

All the values can also be given in millimeters (for example 0.2mm).

Shapes

Smd:
A rounding factor can be specified here for SMD pads. The value can be between 0% (no rounding) and 100% (maximum rounding).



Roundness: 0 - 10 - 25 - 50 - 100 [%]. **Right: 100%, square**

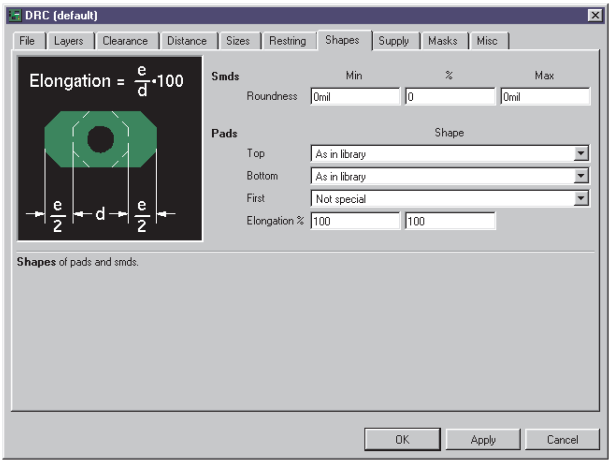
A square SMD has been placed instead of an oblong one on the far right of the diagram. After assigning the property *Roundness* = 100%, the SMD becomes round.

Pad:
This is where the form of the pads is specified. It is possible to give different settings for the top and bottom layers.

The *As in library* option adopts the form defined in the Package Editor. Clicking on *Apply* shows the change immediately in the Layout Editor.

Pads within inner layers are always round, the diameter being determined by the restring settings.

Provided a pad was given the *First* flag in the library one can specify a certain shape for all those pads in the layout.



Smds — Min / % / Max

Roundness: 0mil | 0 | 0mil

Pads — Shape

Top: As in library

Bottom: As in library

First: Not special

Elongation %: 100 | 100

Shapes of pads and smds.

Design Rules: Adjusting Pad Shapes

Elongation defines the aspect ratio of length to width of *Long* and *Offset* pads (see image). The value is given in percent. Click with the mouse into the field *Long* or *Offset* and the image on the left shows the corresponding calculation rule.
100% is equivalent to an aspect ratio of 2:1. 0% results in a normal octagon pad with an aspect ratio of 1:1. The maximum is 200% (ratio 4:1).

Notes on the display in the Layout Editor:

If pads or vias have different shapes on different layers, the shapes of the currently visible (activated with DISPLAY) signal layers are displayed on top of each other.

If the color selected for layer 17, *Pads*, or 18, *Vias*, is 0 (which represents the current background color), the pads and vias are displayed in the color and fill style of their respective layers. If no signal layer is visible, pads and vias are not displayed.

If the color selected for layer 17, *Pads*, or 18, *Vias*, is not the background color and no signal layers are visible, pads and vias are displayed in the shape of the uppermost and undermost layer.

This also applies to printouts made with PRINT.

Supply

Specifies the settings for thermal and annulus symbols. These special symbols are generated in supply layers that are created automatically (layer name starting with a $-sign).

EAGLE calculates the required dimensions for annulus and thermal apertures based on the pad's drill diameter and the annulus and thermal parameters set in the Design Rules:

Gap defines the gap between the insulation bridges on a thermal symbol. The specification is made as a % of the hole diameter, and is constrained by maximum and minimum values.

The *Isolate* values for *Thermal* and *Annulus* determine the width of the thermal bridge or ring.

The *Restring* option determines whether the insulation bridge of the thermal symbol should be drawn immediately at the edge of the hole or at a distance from the hole given by the restring value (*Restring* tab, *Inner* setting).
If the *Restring* option for *Annulus* is deactivated, a filled circle is generated instead of the annulus ring. This is the default setting.

The isolate value for *Thermal* also applies to polygons. It determines the distance between a polygon and the restring of the pad or via that is joined to the polygon through a thermal symbol.

The *Generate thermals for vias* flag permits thermal symbols at through-hole contacts. Otherwise vias are fully connected to the copper plane. This applies also for polygons. But you can disable this option for individual polygons (CHANGE THERMALS OFF).

Calculations for automatically created symbols in supply layers:

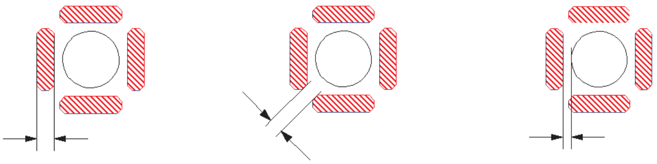
Annulus:

```
    Inner diameter = pad hole diameter + 2 * restring
    Outer diameter = max(pad-hole diameter, inner diameter)
                         + 2*isolate
```

Default: restring = 0 (fully filled), isolate = 20 (mil).

Thermal:

```
    Inner diameter = pad hole diameter + 2 * restring
    Outer diameter = inner diameter + 2 * isolate
```

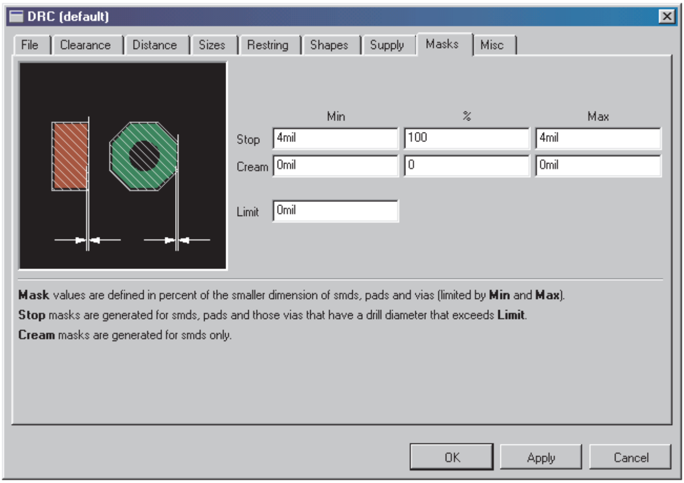


Isolate **Gap** **Restring**

Pads or smds marked with the flag *NOTHERMALS* (resp. CHANGE THERMALS OFF) in the Package Editor will be connected basically without thermal symbols.

Masks

Settings for the overmeasure of the solder stop mask (*Stop*) and the solder cream mask (*Cream*) are made here.



	Min	%	Max
Stop	4mil	100	4mil
Cream	0mil	0	0mil

Limit 0mil

Mask values are defined in percent of the smaller dimension of smds, pads and vias (limited by **Min** and **Max**).
Stop masks are generated for smds, pads and those vias that have a drill diameter that exceeds **Limit**.
Cream masks are generated for smds only.

Design Rules: Settings for Solder Stop and Cream Frame

The default value for solder stop is 4 mil, i.e. minimum value is maximum value is 4 mil. The percent value has no effect in this case.
The value for the cream frame is set to 0, which means that it has the same dimensions as the smd.

If values are given in percent, in the case of smds and pads of the form *Long* or *Offset*, the smaller dimension is the significant one. The values are constrained by minimum and maximum values.

The value for *Cream* is given positively, as is *Frame*, although its effect is to reduce the size of the solder cream mask (creamframe).
The solder cream mask is only generated for SMDs, and is displayed on layer 31, *tCream*, or layer 32, *bCream*. The solder stop mask is drawn in layers 29, *tStop*, or 30, *bStop*.

Setting the flag *STOP* or *CREAM* (only for SMD) to *OFF* for a pad or smd at the package definition forbids EAGLE to generate a solder stop mask or a cream frame for it.

Limit determines, together with the hole diameter, whether or not a via is to be covered with solder stop lacquer.

Example:
The default value for *Limit* is 0. This means all vias get a solder stop symbol. They are free of solder stop laquer.
Set the *Limit* = 24:
All through-plated holes with diameters up to 24 mil don't get a solder stop symbol (they are laquered), but vias with larger hole diameters get

a solder stop symbol.
For vias with hole diameters below the *Limit* the *STOP* flag can be set (CHANGE STOP ON). EAGLE generates a solder stop mask then.

Misc

Here you can select/deselect various checks:

Check grid examines whether elements lie precisely on the grid currently set by the GRID command. This test is not always worthwhile, since in many cases devices built to both metric and imperial grids are in use at the same time. No common grid can be found in such a case.

Check angle ensures that all tracks are laid at whole multiples of 45 degrees. This test is normally switched off, but can be activated if required.

Check font (de-)selects the font check.
The DRC checks if texts are written in vector font. Text which is non-vector font is marked as an error. This check is necessary due to the fact that the CAM Processor can't work with others than vector font for the generation of manufacturing data.
Assumed you use proportional font text in the bottom layer, place it between two tracks, and use the CAM Processor to generate Gerber files, it could happen that the tracks are shorted by the text (height and length of the text can change)!
Default: on.

Check restrict can be set off if copper objects should not be checked against restricted areas drawn in layers 39, *tRestrict*, and 40, *bRestrict*.
Default: on.

Stop after xx errors specifies the maximum number of errors in the error list. As soon as this many errors have been found, the DRC stops examining the layout and automatically opens the error list.

Advice on handling errors is found later in this chapter.

The DRC treats a line (wire) whose *Style* is LongDash, ShortDash or DashDot in the same way as a continuous line.
If a wire drawn with one of these styles is laid as a signal, the DRC reports a *Wirestyle* error.

6.3 Create Board

After you have created the schematic, click the *Board* icon.

An empty board is generated, next to which the components are placed, joined together with airwires. Supply pins are connected by those signals which correspond to their name, unless another net is explicitly joined to them.

The board is linked to the schematic by the Forward&Back Annotation. Provided that both files are always loaded during editing they are guaranteed to remain consistent. Alterations made in one file are automatically carried out in the other.

If, for example, the schematic diagram is loaded and edited without the layout, consistency is lost. The Forward&Back Annotation no longer functions. Differences must then be rectified manually with the aid of the error messages provided by the ERC.

Without the Schematic Editor

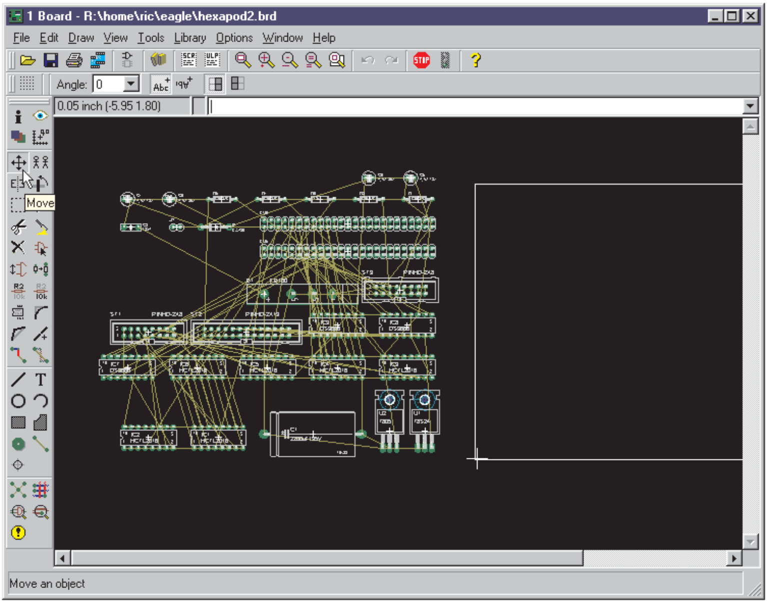
If you work without a Schematic Editor, you must generate a new board file, place the packages with the ADD command and define the connections with the SIGNAL command.

To understand this process, please read the section on *Placing Components* on page 88, and the section on *Specifying Net Classes* on page 93. These two points apply to the Layout Editor as well as to the Schematic Editor.

The remaining procedures are identical for users with and without the Schematic Editor.

Specify the Board Outline

A board that has just been generated from a schematic diagram initially appears as in the following diagram:


1 Board - R:\home\ric\eagle\hexapod2.brd
File Edit Draw View Tools Library Options Window Help
Angle: 0
0.05 inch (-5.95 1.80)
Move
Move an object

Board command: Create the layout from the schematic

The devices are automatically placed at the left of the board. The board outline is drawn as a simple line in layer 20, *Dimension*. The outline of an eurocard is displayed in the Professional and Standard editions, a half-eurocard in the Light edition.

If you wish, you may alter the size or shape of the empty board with the MOVE and SPLIT commands. Please note the hints in the help function for WIRE and MITER. It's easily possible to draw round outlines, too.
You can also delete the outline and place a frame from a library (such as *19inch.lbr*) with ADD.
A script file can, alternatively, be read by the SCRIPT command. The *euro.scr* file, for instance, can be used. Simply type

```
SCRIPT EURO
```

on the command line.

The board outline serves simultaneously as a boundary for the autorouter (if present).

Arrange Devices

Drag the various elements to the desired positions. You use the MOVE command for this purpose. Devices can be clicked on directly, or addressed by name.

If, for example, you type

```
MOVE R14
```

onto the command line, the device named R14 will be attached to the mouse cursor, and can be placed.

Precise positioning results from input such as:

```
MOVE R14 (0.25 2.50)
```

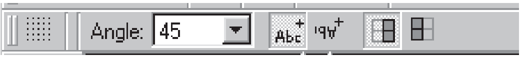
R14's locating point is now located at these coordinates.

Keep the *Ctrl* key pressed while selecting a component in order to let its origin jump at the mouse cursor and move it onto the currently used grid.

A group of devices can be transposed by combining the GROUP and MOVE commands. Click on the GROUP icon and then draw a frame around the desired elements, click MOVE, and then click within the group with the right mouse button in order to select it. With a click of the left mouse button you can place the group at the desired location.

ROTATE, or a click with the right mouse button while the MOVE command is active turns a device through 90 degrees. This also applies to groups.

In order to place a component in any angle you may specify the rotation directly with the ADD command or later with ROTATE or MOVE in the parameter toolbar.



Parameter toolbar for ROTATE, MOVE, ADD, COPY, TEXT

Next to the *Angle* box are the buttons for the *Spin* and *Mirror* flag.

Entering

```
ROTATE R45 'IC1' ;
```

in the command line adds a rotation of 45° to the current position of element IC1. Assumed you tried, for example, to rotate the element with the ROTATE command and pressed mouse button, and you decided that it is not possible to obtain the exact rotation angle this way (because of a too coarse grid) type in the command line:

```
ROTATE =R45 'IC1' ;
```

The rotation of IC1 is now exactly 45°. The = sign stands for absolute values. The initial position does not matter.

If, for example, a smd should be placed on the bottom side of the board you may add the *Mirror* flag, as in:

```
    ROTATE =MR45 'IC1' ;
```

An additional *Spin* flag causes texts to be written upside down (by a rotation of 180°), that means they can be read from the top view:

```
    ROTATE =SMR180 'IC1' ;
```

The *Spin* flag is alternating, i. e. using it again causes the text to be displayed 'normal' again.

Check frequently whether the placement is favorable or not. To do this, use the RATSNEST command. This calculates the shortest connections of the airwires.

The position of particular devices can be displayed by typing the device name onto the command line or by clicking directly on an element while the SHOW command is active.

Clicking with INFO onto a device provides detailed information about it.

If the text for the name or the value is located awkwardly, separate them from the device with SMASH and move them to whatever position you prefer with MOVE. At the same time EAGLE shows a line from the text's origin to the belonging element. Clicking with DELETE on either of the texts makes it invisible.

Holding down the *Shift* key while the SMASH command is active the name and value texts will appear at their original positions again. They are no longer editable and are so to speak unsmashed.

Please keep in mind that the CAM Processor ***always*** *uses vector font for generating manufacturing data.*
We recommend to write texts in the layout always in vector font (at least in the signal layers). If you do so the shown text meets exactly reality. Further information can be found on pages 33 and 119.

Boards with Components on Both Sides

If the board is also going to have components on the underside, the MIRROR command is used. It causes devices on the underside to be inverted. SMD pads, the silk screen and the layers for the solder stop and solder cream masks are automatically given the correct treatment here.

While ADD, COPY, MOVE, or PASTE is active it is possible to mirror an element or a selected group with the middle mouse button.

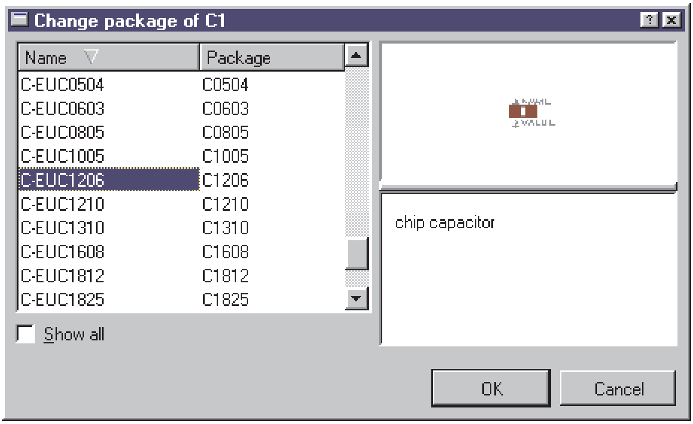
It is not necessary for the Package Editor to define devices in the library as being on the bottom side.

Exchanging Packages

If, as the layout is developed, you want to replace the selected package variant with a different one, then you can use either the CHANGE or the REPLACE command, depending on the situation.

CHANGE Command

It is assumed that the layout and the schematic diagram are consistent and the Device has been defined with more than one package variant. Click on the CHANGE icon and select the *Package* option. You select the desired package, and confirm it with *OK*, in the dialog that then appears.



Name	Package
C-EUC0504	C0504
C-EUC0603	C0603
C-EUC0805	C0805
C-EUC1005	C1005
C-EUC1206	C1206
C-EUC1210	C1210
C-EUC1310	C1310
C-EUC1608	C1608
C-EUC1812	C1812
C-EUC1825	C1825

chip capacitor

CHANGE package dialog

If the *Show all* option is active, the package versions for all the technologies available for this device are displayed. If this option is not active you will only see packages that are defined in the selected technology.

The package can also be exchanged from within the schematic diagram.

Devices with a single package variant have to be modified in the Library Editor before. Add further package variants as needed. See page 186 *Choosing the Package Variants* for further information.

If you want to change a package which you gave a new value with the help of the VALUE command before, the value text of the new package will drop to the original value although the device has been defined with VALUE *Off* in the library. See also page 57.

REPLACE command

If you have a layout without an associated schematic diagram, you exchange the package with the aid of the REPLACE command. REPLACE opens the window that is familiar from the ADD dialog, in which it is possible to search for devices. When the package has been chosen you click on the device that is to be replaced in the layout.

The REPLACE command operates in two ways, chosen with the SET command:

```
SET REPLACE_SAME NAMES;  (default)
SET REPLACE_SAME COORDS;
```

The first mode permits packages whose pad or SMD names are identical to be exchanged. The connecting areas can have any position.

In the second case (replace_same coords) the pads or SMDs in the new package must be located at the same coordinates (relative to the origin). The names may differ.

The text for the name and value of a device is only exchanged if they have not been separated from the device with SMASH.

The new package can come from a different library, and can contain additional pads and SMDs. Connections on the old package that were connected to signals must exist correspondingly in the new package. The new package can even have fewer connections, provided that this condition is satisfied.

Changing the Technology

It is possible to change the technology of a device in the layout at any time , provided there are various technologies defined in the library. Use the CHANGE command with the *Technology* option. This procedure is identical to the one described before in which package forms are exchanged using CHANGE.

Define Forbidden Areas

Areas in the form of rectangles, polygons or circles in layers 41, *tRestrict*, 42, *bRestrict*, and 43, *vRestrict*, are forbidden for the Autorouter (see RECT, POLYGON, CIRCLE). No tracks may be drawn in the top or bottom layers inside these areas. Through-plated holes may not be located in a vRestrict region. These regions are examined by the DRC.

Routing — Placing Tracks Manually

The ROUTE command allows the airwires to be converted into tracks. A click on the center mouse button while a track is being laid allows a change of layer. A via is placed automatically. Clicking with the right mouse button changes the way in which the track is attached to the mouse and how it is laid (SET command, *Wire_Bend* parameter). Among them are two modi which allow to use a wire as a 90-degree or as a free-definable arc.

The signal's name and net class will be displayed in the status bar, as far as the *User guidance* has been deactivated in the menu *Options/User Interface*. When a signal line has been completely laid, EAGLE confirms that there is a correct connection with a short beep as it is placed.

If there is no longer a wiring route available for some signals, other tracks are relocated with MOVE and SPLIT, or the properties of tracks (width, layer) are modified with CHANGE.

If a through-plated hole is to be placed at a certain point, this can be done with the VIA command. The via is given the name of a signal using NAME.

If you want to begin a wire of a partially routed signal at another position than the start or end point of an airwire (for example, at a farther position on a wire that belongs to the same signal) simply press the *Ctrl* key. EAGLE generates a new airwire from this position (see also help function for ROUTE).

Airwires with length of 0 (for example, from *Top* to *Bottom* layer) are drawn as a cross in layer 19, *Unrouted*.

Ending a wire at the same position where another wire of the same signal but in another layer already exists and pressing the *Shift* key at the same time causes EAGLE to place a via. Otherwise it won't.

If you intend to design a multilayer board and use blind and buried or micro vias, please note the details (also for the VIA command) in the section about *Multilayer Boards* beginning at page 121.

While laying out wires EAGLE calculates the shortest connection to the closest point of the current signal automatically. This connection is represented by an airwire.
As the routing proceeds it is helpful to run the RATSNEST command frequently, in order to recalculate all the airwires.

Use RIPUP if you want to convert the whole or part of a track or a via that has been laid back to a signal line. By clicking on a track it is decomposed between the nearest bends. If you click on this location again (on the airwire), the whole signal branch back to the nearest pads is decomposed. If you want to undo the whole of the signal, click RIPUP and enter the name of the signal on the command line. More than one may be entered at the same time.

The command

```
    RIPUP GND VCC +5V
```

converts the three signals GND, VCC and +5V back to airwires.

```
    RIPUP ! GND VCC
```

on the other hand converts all signals apart from GND and VCC to airwires.

```
    RIPUP ;
```

converts all signals (that are visible in the editor) into airwires. To truly include every track, all the layers in which tracks have been drawn must be visible (DISPLAY).

If you want to use wires as arcs or try to smooth the wire bends see the hints concerning the MITER command in the help function. The miter radius determins how the wire joints are mitered. A positive value generates a rounding, a negative one a straight line. The miter radius influences some bend modi (0, 1, 3, 4; see SET command) and is shown additionally in the parameter toolbar of the commands SPLIT, ROUTE, WIRE, and POLYGON.

While WIRE or ROUTE is acitve it is possible to click through the previously mentioned wire bends (bend modi) with the right mouse button. EAGLE knows eight modi (0..7).
Holding down the *Shift* key while clicking the right mouse button reverses the direction of selection.
Holding down the *Ctrl* key allows to toggle between complementary wire bends.

If you want to have only some wire bends available for the right mouse button, you can define this, for example, in the *eagle.scr* file.
Supposed you want to work with wire bends number 2, 5, 6, and 7 use the following syntax:

```
    SET WIRE_BEND @ 2 5 6 7 ;
```

However, if you want to use another bend mode you can always chose it from the parameter toolbar.

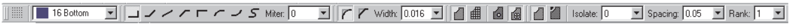
It is also possible to leave the track laying to the Autorouter which has its own chapter in this manual.

Defining a Copper Plane with POLYGON

EAGLE can fill regions of a board with copper. Simply draw the borders of the area with the POLYGON command. You give the polygon a signal name, using NAME followed by a click on the border of the polygon. Then all the elements that carry this signal are connected to the polygon. Both pads and, optionally, vias (as specified in the Design Rules) are joined to the copper plane through thermal symbols. Elements not carrying this signal are kept at a specified distance.

RATSNEST calculates and displays the surface area. RIPUP, followed by a click on the polygon border, makes the content invisible again. The content of the polygon is not saved in the board file. When you first load the file, you will only see the outline of the polygon. It is only calculated and displayed again by RATSNEST.

Various options can be changed via the parameter toolbar, either as the polygon is being drawn or, with CHANGE, at a later stage.



16 Bottom
Miter: 0
Width: 0.016
Isolate: 0
Spacing: 0.05
Rank: 1

POLYGON command: Parameter toolbar

Width:
Line thickness with which the polygon is drawn. Select the largest possible width. That avoids unnecessary quantities of data when the board is sent for manufacture. If the wire width is lower than the resolution of the output driver in the CAM Processor, a warning is issued.
A finer line width permits the polygon to have a more complex **shape.**

Pour:
Specifies the filling type. The whole area (*Solid*) or a grid (*Hatch*).

Rank:
Overlapping polygons must not create any short-circuits. *Rank* can therefore be used to determine which polygons are to be subtracted from others. A polygon with rank = 1 has the highest priority in the Layout Editor (no other polygon drawn in the layout is ever subtracted from it), while one with rank = 6 has the lowest priority. As soon as there is an overlap with a higher rank, the appropriate area is cut out from the polygon with rank = 6. Polygons with the same rank are compared by the DRC.

Polygons that are created in the Package Editor can be assigned the rank 0 or 7. Rank = 0 has highest priority and will not be touched by other polygons. Polygons with rank = 7 will be pushed away from all other polygons drawn in the layout.

Spacing:
If the option *Hatch* is chosen for *Pour*, this value determines the spacing of the grid lines.

Isolate:
Defines the value that the polygon must maintain with respect to all other elements not part of its signal. If higher values are defined for special elements in the Design Rules or net classes, the higher values apply.

Thermals:
Determines whether pads in the polygon are connected via thermal symbols, or are completely connected to the copper plane. This also applies to vias, assuming that the option has been activated in the Design Rules.

The width of the thermal connectors is calculated as the half of the pad's drill diameter. The width has to be in the limits of a minimum of the wire width and a maximum of twice the wire width of the polygon.

Don't choose the polygon's width too fine, otherwise the thermal connectors won't resist the current load.

Orphans:
Determines if a polygon may contain areas (islands) which are not electrically connected to the polygon's signal.
If Orphans is set *Off* such un-connected areas won't be drawn.

When drawing a polygon, please take care to ensure that the outline is not drawn more than once (overlapping) anywhere, and that the polygon outline does not cross over itself. It is not possible for EAGLE to compute the contents of the area in this case.
An error message Signalname contains an invalid polygon! *is issued, and the RATSNEST command is aborted.*
If this message appears, the outline of the polygon must be corrected. Otherwise, manufacturing data cannot be created by the CAM Processor.
The CAM Processor automatically computes the polygons in the layout before generating its output.

DRC — Checking the Layout and Correcting Errors

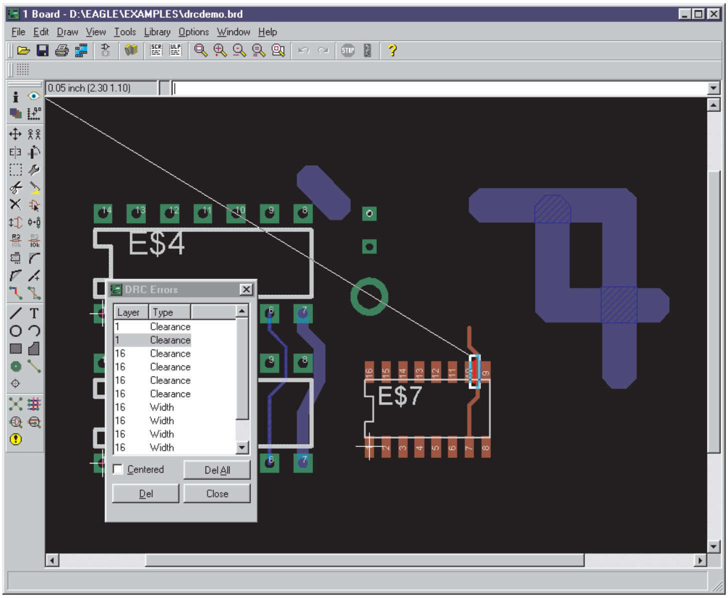
The Design Rule Check (DRC) is carried out at the end of the board design, if not before. If you have not yet specified any Design Rules for the layout, this is your last opportunity. See the section on *Specifying the Design Rules* from page 98.

If the DRC finds errors, an error window opens automatically. This lists all the errors. The window can be opened at any time by means of the ERRORS command.

If an error in the list is selected, a line points to the corresponding location. The error is marked. Correct the error in the layout. The DRC error window remains in the foreground.

Clicking on the *Del* button will delete the selected error entry. *Del all* deletes all the error marks from the layout.

The dialog is ended with *Close*.


1 Board - D:\EAGLE\EXAMPLES\drcdemo.brd
File Edit Draw View Tools Library Options Window Help
0.05 inch (2.30 1.10)
E$4
E$7
DRC Errors
□ Centered
Del All
Del
Close

Layer	Type
1	Clearance
1	Clearance
16	Clearance
16	Clearance
16	Clearance
16	Clearance
16	Width
16	Width
16	Width
16	Width

DRC error list in the Layout Editor

Signal layers which are not visible (DISPLAY command) will not be checked by the Design Rule Check!

Error messages and their meaning

Angle:
Tracks are not laid in an angle of 0, 45, 90 or 135°. This check can be switched on or off in the Design Rules (*Misc* tab).
Default: off.

Blind Via Ratio:
The limit of the ratio of depth to drill diameter is exceeded. In this case you have to adjust the via's drill diameter (see Design Rules, *Sizes* tab).

Clearance:
Clearance violation between copper elements. The settings of the Design Rules' *Clearance* tab and the value for *Clearance* of a given net class will be checked. In addition the *Isolate* value will be taken into consideration for polygons with the same rank and polygons which are defined as part of a package.
To deactivate the clearance check between elements that belong to the same signal, use the value 0 for *Same signals* in the *Clearance* tab.

Dimension:
Distance violation between smds, pads, and connected copper objects and a dimension line (drawn in Layer 20, *Dimension*), like the board's outlines. Defined through the value for *Copper/Dimension* in the

Design Rules (*Distance* tab).
Setting the value *Copper/Dimension* to *0* deactivates this check.
In this case polygons do not keep a minimum distance to objects in layer 20, *Dimension*, and holes! The DRC will not check if holes are placed on tracks!

Drill Distance:
Distance violation between holes. Defined by the value *Drill/Hole* in the Design Rules (*Distance* tab).

Drill Size:
Drill diameter violation in pads, vias, and holes. This value is defined in the Design Rules (*Minimum Drill, Sizes* tab). It is also possible to define a special value (*Drills*) for vias in a given net class (CLASS command).

Keepout:
Restricted areas for components drawn in layer 39, *tKeepout,* or 40, *bKeepout*, lie one upon another. This ceck is made only if layers 39 and 40 are displayed.

Layer Abuse:
Layer 17, *Pads*, or 18, *Vias*, contains objects which can't be determined by EAGLE. Those layers are exclusively dedicated to pads and vias. Better move such objects into another layer.

Layer Setup:
This error is shown if an object in a layer is found that is not defined by the Layer setup is found. The same for vias that do not follow the settings of the Layer setup, for example, if a via has an illegal length (blind/buried vias).

Micro Via Size:
The drill diameter of the micro via is smaller than the value given in the *Sizes* tab.

No vector font:
The font check (Design Rules, *Misc* tab) recognizes text in a signal layer which is not written in EAGLE's internal vector font.
If you want to generate manufacturing data with the help of the CAM Processor the texts, at least in the signal layers, ought to be written in vector font. This is the only font the CAM Processor can work with. Otherwise the board will not look the same as it is shown. Change the font with the help of the command CHANGE FONT or use the option *Always vector font* in the Layout Editor's *Options/User Interface* menu:
If activated, the Layout Editor shows all texts in vector font. This is the way the manufactured board will look like.
Activating the sub-option *Persistent in this drawing* saves the setting in the drawing file. If you send the layout file, for example, to the boardhouse you can be sure that the vector font will be displayed also at his system.

No real vector font:
The font check (Design Rules, *Misc* tab) recognizes text in a signal layer which is not written in EAGLE's internal vector font although it is displayed as vector font in the Layout Editor window. This situation arises if the option *Always vector font* in the menu *Options/User Interface* is active.
See error message *No vector font* for further details.

Off Grid:
The object does not fit onto the currently chosen grid.
This check can be switched on or off in the Design Rules (*Misc* tab). Default: off.

Restrict:
A wire drawn in layer 1, *Top,* or 16, *Bottom,* or a via lies in a restricted area which is defined in layer 41 or 42, *t/bRestrict*.

Width:
Minimum width violation of a copper object. Defined by *Minimum Width* in the Design Rules (*Sizes* tab) or, if defined, by the track parameter *Width* of a referring net class. Also texts in signal layers will be checked.

Wire Style:
The *wire styl*e of a wire which is connected to a signal is not *Continuous*.

Net, device and pin lists can be output by means of EXPORT or by various User Language programs.

Creating Manufacturing Data

Prints of the layout can be created for purposes of documentation using the PRINT command. Layers that have been activated with DISPLAY are printed. The printout has the same appearance as the layout has on the screen.
Exception: Origin marks of texts and grid lines are not printable.

The manufacturing data is typically generated with the CAM Processor. A variety of drivers are available to create drilling data, Gerber data, plotter data and so forth. We will return to this subject in a specific chapter later.

In order to save costs, it may be worth supplying the layout to the board manufacturer in the form of a multiple board. You can reproduce the layout to create a multiple board with the GROUP, CUT and PASTE commands. It should be noted that this will change the board’s silk screen, since devices and signals receive new names when the layout is placed. Ensure that all the layers are displayed before using the GROUP and CUT commands.

Any other data format can be created with the aid of ULPs (User Language programs). See the short description of ULPs in the Control Panel.

6.4 Multilayer Boards

You can develop multilayer boards with EAGLE. To do this, you use one or more inner layers (*Route2* to *Route15*) as well as the layers *Top* and *Bottom* for the top and undersides. You display these layers when routing.

Before starting the routing of the layout you should be aware of the number of signal layers to use, if vias should go through all layers, or if you have, due to the complexity of the layout, to work with blind, buried or micro vias. In this case you realy ought to contact your boardhouse to inform you about the possible structure of the board and the resulting costs to be expected.
Also think about how to use the inner layers: As a normal inner signal layer or as a supply layer? On this, for example, depend reasonable via lengths that have to be defined in the Layer setup.

Inner Layer

Inner layers are used the same way as the outer layers *Top* and *Bottom*. They can be filled with copper areas (polygons) as well. A special status have the so-called supply layers which are generated automatically.

Before using inner layers you must define them in the Design Rules, *Layers*-Tab. More details can be found in the following sections and on page 100.

Signal Layers

You use the ROUTE command as usual to place tracks in those inner layers which are provided for signals. Eagle will itself ensure that the tracks are connected by way of plated-through holes to the appropriate signals in the outer layers. The settings of the Layer setup in the Design Rules apply for this procedure.

Supply Layers with Polygones and More than One Signal

Areas of the board can be filled with a particular signal (e.g. ground) using the POLYGON command. The associated pads are then automatically connected using thermal symbols. The isolate value for the thermal symbols is specified in the Design Rules (DRC command, *Supply* tab). The width of the connecting bridge depends on the line thickness with which the polygon is drawn (see page 117). You can also specify whether or not vias are to be connected through thermals. The minimum clearances from elements carrying other signals specified in the Design Rules are maintained (*Clearance* and *Distance* tabs). Changes are shown in the layout when the polygon is next computed (RATSNEST command).

In this way you can also create layers in which several areas are filled with different signals. In that case you can assign different ranks (priorities) for the polygons. The rank property determines which polygon is subtracted from others if they overlap. Rank = 1 signifies the highest priority in the layout; nothing will be subtracted from such a polygon (with the exception of polygons with rank = 0, drawn as a part of a package in the Package Editor). Rank = 6 signifies the lowest priority. Polygons with the same rank are compared by the DRC.

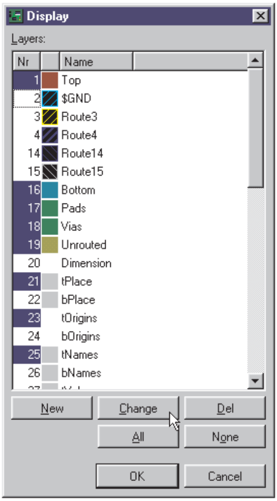
Please read the notes regarding polygons in the section on *Defining a Copper Plane* on page 115.

Do not choose the wire width for polygons too fine! This can lead to huge amounts of plot data and problems for the manufacturing process.

Power Supply Layer with One Signal

A power supply layer with one signal is implemented by renaming one of the Route2...15 layers in such a way that the new name consists of the corresponding signal name prefixed by a $ sign. If, for example, the signal called GND is to be realized as a power supply layer, then a layer is specified as having the name $GND.

The DISPLAY menu shows only those layers defined in the Layer setup of the Design Rules!

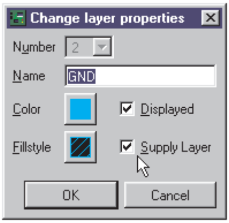


Display

Layers:

Nr	Name
1	Top
2	$GND
3	Route3
4	Route4
14	Route14
15	Route15
16	Bottom
17	Pads
18	Vias
19	Unrouted
20	Dimension
21	tPlace
22	bPlace
23	tOrigins
24	bOrigins
25	tNames
26	bNames

For this purpose you activate the DISPLAY command. You select layer 2 from the menu with the mouse. Then click on the *Change* button.



Change layer properties

Number: 2

Name: GND

Color ☑ Displayed

Fillstyle ☑ Supply Layer

Enter the signal name for the layer, and activate the *Supply layer* check box. If the layer is to be visible in the layout immediately, then click also on the *Displayed* check box.

The command corresponding to this example would be:

```
    LAYER 2 $GND
```

This specifies that layer number 2 (previously known as *Route2*) is henceforth known as $GND, and that it will be treated as a power supply layer.

Supply layers are displayed and plotted inversely!

Pads are connected to power supply layers with what are known as thermal symbols, or are isolated with annulus symbols. Thermal symbols usually just have four thin bridges as a conductive connection to the through-plated hole. They are used because the high thermal conduction of a continuous copper plane would result in the pad being no longer solderable. The shape and size of the annulus and thermal symbols are specified in the Design Rules (DRC command, *Supply* tab).

To keep the board edges free of copper draw an isolating wire around the layout. This avoids possible short circuits between adjacent (supply) layers. If you want to use the Autorouter, however, these wires must not be drawn until the routing process has been finished. Automatically generated supply layers have to be disabled in the Autorouter setup.

No additional signals or signal polygons may be drawn in these layers. That would lead inevitably to an unusable circuit board!

Details on Supply Layers and Autorouter:

The Autorouter takes into account all inner layers, and so delivers the full set of patterns for multilayer boards. It connects SMD pads to inner layers with vias.

Assumed only vias that can connect all layers (from layer 1 to 16) are used (through vias), the preferred direction for the Autorouter has to be set to *N/A* for power supply layers (this means the layer is not available for routing). This is only possible for layers that do not contain any wires.

If you are using, among vias through all layers, blind, buried, or micro vias the Autorouter can't work with supply layers. In that case you have to change the supply layer into an inner layer with polygon:
Therefore deselect the *Supply* flag in the *Change* window of the DISPLAY menu and draw a polygon in the inner layer. The polygon has to be given the name of the supply signal.

Multilayer Boards with Through Vias

This type should be preferred if possible. Vias go through all signal layers and will be drilled at the end of the production process. The production costs are relatively moderate.

Layer Setup

The settings concerning layer composition and number of signal layers are made in the Design Rules, *Layers* tab, *Setup*. See page 100.

For through vias the setup is very simple. No considerations about thickness of copper and isolation layers are necessary.

Simply join two layers by an asterisk (like *1*2* or *3*16*) to one core and combine several cores. This is symbolized by a plus character (like in *1*2+3*4*). The isolation layer between two copper layers is called prepreg. To express the possibility to have vias through all layers the whole expression is set into paranthesis.

Examples:

4 layers:	`(1*2+3*16)`
6 layers:	`(1*2+3*4+5*16)`
8 layers:	`(1*2+3*4+5*6+7*16)`

Here vias always have the length 1-16. They are reachable from all layers (see also the help function for VIA).

Multilayer with Blind and Buried Vias

In high density boards it is often necessary to use blind and burid vias. These kinds of vias don't connect all layers, but are only reachable from a certain number of layers. How these layers are connected depends on the manufacturing process of the board which has to be determined in the Layer setup in the Design Rules.

Please contact your boardhouse before starting your work! Check which Layer setup is suitable for your purpose.

Disambiguation

Core:
The non-flexible kernel which is coated with copper on one or on both sides.

Prepreg:
Flexible glueing or isolating layer which is used in the manufacturing process of a multilayer board to press inner and outer layers onto each other.

Layer Stack:
A pack of any number of layers consisting of cores and prepregs which are handled together in the current step of production.

Buried Via:
The production process of this via does not differ from a through (normal) via. The current layer stack will be drilled through completely. In the following production steps the already drilled vias can be covered (buried) by pressing further cores and prepregs on the current layer stack. If the via is not visible on the completed board we call it a buried via.

Blind Via:
Blind vias connect an outer layer with any inner layer but don't go through all copper layers. The specialty of a blind via lies in the production process. The current layer stack is not drilled all through. The drill hole has a certain depth depending on the number of layers that should be allowed to be connected with each other. Blind vias have to follow a given ratio of depth to drill diameter. Please contact your boardhouse to get information about this. This ratio has to be defined in the *Sizes* tab as *Min. Blind Via Ratio.*

Micro Via:
The micro via is a special case of a blind via. It has a maximum depth of one layer and a very small drill diameter. See page 131.

Layer Setup

Combining cores and prepregs allows many variants. In the following section some examples show the function of the Layer setup.
Please read this paragraph entirely. Even if you intend to design a four layer board, for example, it is most advisable to read also all the other examples for a better understanding.

4-Layer Board

Example 1:

Layers 1, 2, 3 and 16 are used.

Board structure: One core inside, outside prepregs.

Connections: 1-2 (blind vias), 2-3 (buried vias) and 1-16 (through vias)

The setup expression looks like this:

```
[2:(1+(2*3)+16)]
```

Explanation:

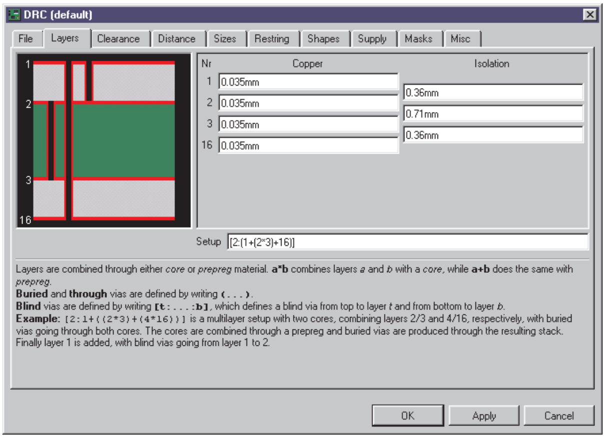
`2*3`
Layers 2 and 3 form the core.

`(2*3)`
Parenthesis allow buried vias from 2 to 3.

`(1+``(2*3)`**`+16)`**
On both sides of the core copper layers are pressed on with prepregs. The outer parenthesis define continuous vias from 1-16.

`[2:``(1+(2*3)+16)`**`]`**
In square brackets and separated by a colon blind vias are defined. Here from layer 1 to 2.

The following image shows the related setup expression in the *Layers* tab of the Design Rules.



Nr	Copper	Isolation
1	0.035mm	0.36mm
2	0.035mm	0.71mm
3	0.035mm	0.36mm
16	0.035mm	

Setup [2:(1+(2*3)+16)]

Layers are combined through either *core* or *prepreg* material. **a*b** combines layers *a* and *b* with a *core*, while **a+b** does the same with *prepreg*.
Buried and **through** vias are defined by writing `(...)`.
Blind vias are defined by writing `[t:...:b]`, which defines a blind via from top to layer *t* and from bottom to layer *b*.
Example: `[2:1+((2*3)+(4*16))]` is a multilayer setup with two cores, combining layers 2/3 and 4/16, respectively, with buried vias going through both cores. The cores are combined through a prepreg and buried vias are produced through the resulting stack. Finally layer 1 is added, with blind vias going from layer 1 to 2.

Example 1: Layer Setup for a 4-Layer Board

Blind vias have to keep a certain ratio of via depth to drill diameter. For this reason it is necessary to specify values for the layer thickness. These values are given by your boardhouse! You are supposed to contact it in either case before starting the layout!

Type in the values in the *Copper* (thickness of copper layer) and *Isolation* (thickness of isolation layer) fields as shown in the image.

Example 2:

Layers 1, 2, 3, and 16 are used.

Board structure: One core inside, outside prepregs.

Connections: 1-2, 3-16 (blind vias), 1-16 (through vias)

Setup expression:

```
[2:(1+2*3+16):3]
```

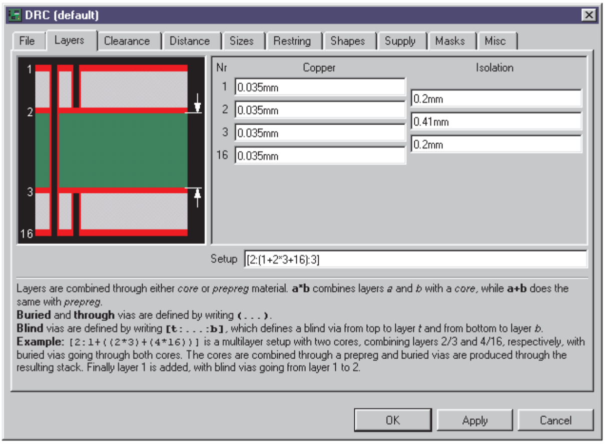
Explanation:

`2*3`
Layers 2 and 3 form the core.

1+`2*3`**+16**
On both sides of the core copper layers are pressed on with prepregs.

(`1+2*3+16`**)**
The outer parenthesis define through vias from 1-16.

[2:`(1+2*3+16)`**:3]**
In square brackets and separated by a colon blind vias are defined. Here from layer 1 to 2 and 16 to 3.



Nr	Copper	Isolation
1	0.035mm	0.2mm
2	0.035mm	0.41mm
3	0.035mm	0.2mm
16	0.035mm	

Setup [2:(1+2*3+16):3]

Layers are combined through either *core* or *prepreg* material. **a*b** combines layers *a* and *b* with a *core*, while **a+b** does the same with *prepreg*.
Buried and **through** vias are defined by writing **`(...)`**.
Blind vias are defined by writing **`[t:...:b]`**, which defines a blind via from top to layer *t* and from bottom to layer *b*.
Example: `[2:1+((2*3)+(4*16))]` is a multilayer setup with two cores, combining layers 2/3 and 4/16, respectively, with buried vias going through both cores. The cores are combined through a prepreg and buried vias are produced through the resulting stack. Finally layer 1 is added, with blind vias going from layer 1 to 2.

Example 2: Layer Setup for a 4-Layer Board

6-Layer Board

Example 3:

Layers 1, 2, 3, 4, 5, and 16 are used.

Board structure: Two cores, prepregs outside.

Connections: 2-3, 4-5 (buried vias), 1-16 (through vias)

Setup expression:

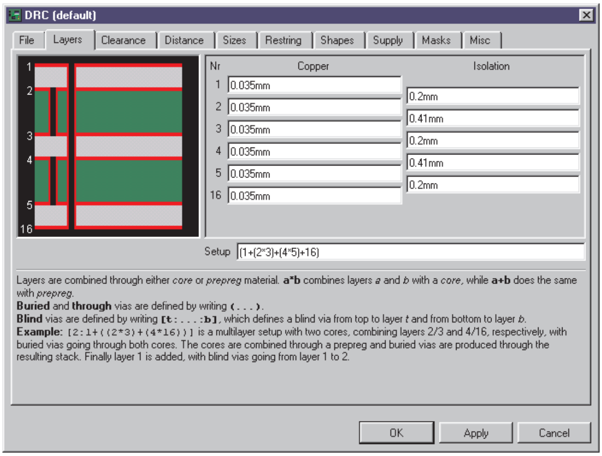
```
(1+(2*3)+(4*5)+16)
```

Explanation:

`(2*3)+(4*5)`
Two cores with buried vias are pressed together.

1+`(2*3)+(4*5)`**+16**
This layer stack is covered with outer layers 1 and 16 which are isolated with prepregs.

(`1+(2*3)+(4*5)+16`**)**
The whole expression in parenthesis defines through vias from 1-16.



Nr	Copper	Isolation
1	0.035mm	0.2mm
2	0.035mm	0.41mm
3	0.035mm	0.2mm
4	0.035mm	0.41mm
5	0.035mm	0.2mm
16	0.035mm	

Setup (1+(2*3)+(4*5)+16)

Layers are combined through either *core* or *prepreg* material. **a*b** combines layers *a* and *b* with a *core*, while **a+b** does the same with *prepreg*.
Buried and **through** vias are defined by writing `(...)`.
Blind vias are defined by writing `[t:...:b]`, which defines a blind via from top to layer *t* and from bottom to layer *b*.
Example: `[2:1+((2*3)+(4*16))]` is a multilayer setup with two cores, combining layers 2/3 and 4/16, respectively, with buried vias going through both cores. The cores are combined through a prepreg and buried vias are produced through the resulting stack. Finally layer 1 is added, with blind vias going from layer 1 to 2.

Example 3: Layer Setup for a 6-Layer Board

The values for layer thickness used in these examples are fictive. Please contact your boardhouse to get the allowed values.

Example 4:

Layers 1, 2, 3, 4, 5, and 16 are used.

Board structure: One core, on each side two prepregs.

Connections: 3-4 (buried vias), 2-4 (blind vias in inner layer stack), 1-16 (through vias)

Setup expression:

```
(1+[4:2+(3*4)+5]+16)
```

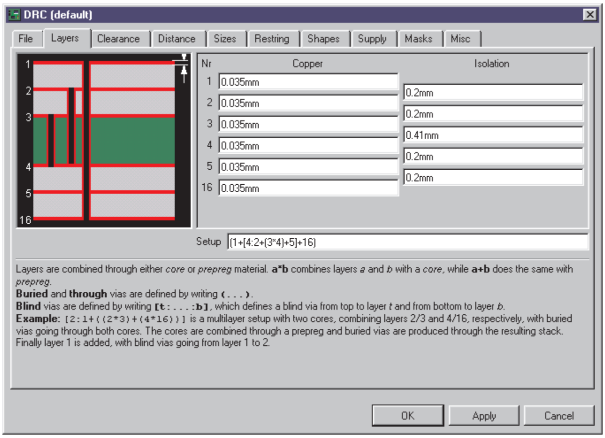
Explanation:

2+(3*4)+5
The core with buried vias. One prepreg on each side.

[4:2+(3*4)+5]
Blind vias from layer 2 to 4.

1+[4:2+(3*4)+5]**+16**
On this layer stack a prepreg on each side is pressed on.

(1+[4:2+(3*4)+5]+16)
Parnethesis allow through vias from 1 to 16.



Nr	Copper	Isolation
1	0.035mm	
		0.2mm
2	0.035mm	
		0.2mm
3	0.035mm	
		0.41mm
4	0.035mm	
		0.2mm
5	0.035mm	
		0.2mm
16	0.035mm	

Setup (1+[4:2+(3*4)+5]+16)

Layers are combined through either *core* or *prepreg* material. **a*b** combines layers *a* and *b* with a *core*, while **a+b** does the same with *prepreg*.
Buried and **through** vias are defined by writing **(...)**.
Blind vias are defined by writing **[t:...:b]**, which defines a blind via from top to layer *t* and from bottom to layer *b*.
Example: [2:1+((2*3)+(4*16))] is a multilayer setup with two cores, combining layers 2/3 and 4/16, respectively, with buried vias going through both cores. The cores are combined through a prepreg and buried vias are produced through the resulting stack. Finally layer 1 is added, with blind vias going from layer 1 to 2.

Example 4: Blind Vias in the Inner Layer Stack

8-Layer Board

Example 5:

Layers 1, 2, 3, 4, 5, 6, 7, and 16 are used.

Board structure: Three cores, prepregs outside.

Connections: 1-3, 6-16 (blind vias), 2-3, 4-5, 6-7 (buried vias), 1-16 (through vias).

Setup expression:

```
[3:1+(2*3)+(4*5)+(6*7)+16:6]
```

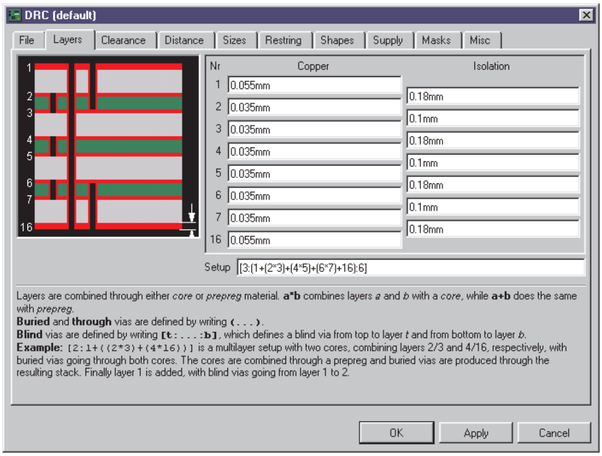
Explanation:

(2*3)+(4*5)+(6*7)
Three cores, each with buried vias, are pressed together and isolated with prepregs.

1+(2*3)+(4*5)+(6*7)**+16**
Outer copper layers 1 and 16 which are isolated through prepregs are pressed onto this layer stack.

(1+(2*3)+(4*5)+(6*7)+16**)**
Parenthesis allow through vias from 1-16.

[3:(1+(2*3)+(4*5)+(6*7)+16)**:6]**
Blind vias from 1-3 and 16-6.



Nr	Copper	Isolation
1	0.055mm	0.18mm
2	0.035mm	0.1mm
3	0.035mm	0.18mm
4	0.035mm	0.1mm
5	0.035mm	0.18mm
6	0.035mm	0.1mm
7	0.035mm	0.18mm
16	0.055mm	

Setup [3:(1+(2*3)+(4*5)+(6*7)+16):6]

Layers are combined through either *core* or *prepreg* material. **a*b** combines layers *a* and *b* with a *core*, while **a+b** does the same with *prepreg*.
Buried and **through** vias are defined by writing `(...)`.
Blind vias are defined by writing `[t:...:b]`, which defines a blind via from top to layer *t* and from bottom to layer *b*.
Example: `[2:1+((2*3)+(4*16))]` is a multilayer setup with two cores, combining layers 2/3 and 4/16, respectively, with buried vias going through both cores. The cores are combined through a prepreg and buried vias are produced through the resulting stack. Finally layer 1 is added, with blind vias going from layer 1 to 2.

Example 5: Layer Setup for an 8-Layer Board

Hints For Working With Blind, Buried, and Micro Vias

VIA command

Depending on the Layer setup vias can have different lengths. The parameter toolbar of the VIA command shows all available lengths in the *Layer* box. When routing manually (ROUTE command) EAGLE takes the shortest possible via length in order to change layers. It is also possible that vias at the same position are elongated.

The via length can be changed with the CHANGE VIA command. Select the value from the according menu and click the via with the left mouse button.

Alternatively use the command line:

```
    CHANGE VIA 2-7
```

and a click on the via changes the length from layer 2 to 7.
If the given via length is not defined in the Layer setup it will be elongated to the next possible length or, if this is not possible, an error message will be generated.

```
    VIA 'GND' 1-4 (1.05 2)
```

places a via that belongs to the signal GND and reaches from layer 1 to 4 at position (1.05 2).

ROUTE Command

If you want to change the layer while laying-out the board, EAGLE always takes the shortest possible via (CHANGE LAYER). It is also possible that a via at the same position is elongated automatically.

If micro vias are enabled in the Design Rules, EAGLE sets a micro via when routing from an SMD and immediately changing to the next inner layer.

Autorouter

The Autorouter supports blind and buried vias.

If you decide to use the Autorouter for a multilayer board that uses blind and buried vias it is not allowed to work with automatically generated supply layers (see page 122). In this case inner layers can be defined with the help of polygons.

Already existing supply layers can be transformed into layers with supply polygons by

- renaming the layer (no $ sign at the beginning of the name),
- removing possibly existing isolation wires,
- drawing a polygon covering the whole layout, and
- naming the polygon with the favored signal name.

Micro Via — A Special Case of Blind Via

In contrast to blind vias that can reach several layers deep into the board the micro via connects an outer layer with the next inner layer. The drill diameter of a micro via is relatively small. Presently the usual values are about 0.1 to 0.05 mm.

For manufacturing resons micro vias, as blind vias, have to follow a certain aspect ratio of depth to drill diameter. This ratio defines the maximum via depth for a certain drill diameter.

The proper value can be learned from your boardhouse.

Set this value in the Design Rules, *Sizes* tab, *Min. Blind Via Ratio.* Assumed the boardhouse demands the ratio as 1:0.5 you have to enter `0.5` for *Min. Blind Via Ratio*.

Additionally the Design Rule Check verifies the minimum drill diameter for mircro vias given in *Min. MicroVia*. If this value is higher than the value for *Minimum Drill* (default), micro vias won't be checked.

The diameter of micro vias is set in the *Restring* tab of the Design Rules.

6.5 Updating Components (Library Update)

The UPDATE command allows components in a schematic diagram or a layout to be replaced by components defined in accordance with the current libraries. This function is of particular interest for existing projects. If, in the course of development, the definitions of packages, symbols or devices in the libraries are changed, the existing project can be adapted to them.

The menu item *Library/Update* causes all the components in a project to be compared with the definitions in the current libraries. If EAGLE finds differences, the components are exchanged.

Those libraries on the path specified for *Libraries* in the Control Panel under *Options/Directories* will be examined.

It is also possible to update components from one particular library. Type the UPDATE command on the command line, stating the library, for instance as:

```
UPDATE linear
```

or

```
UPDATE /home/mydir/eagle/library/linear.lbr
```

or select the library in the File dialog of the *Library/Update...* menu item.

In the case you want to replace parts from one library with parts from another library you can use the command:

```
UPDATE old-lbr-name = new-lbr.name
```

Old-lbr-name represents the name of the library as shown by the INFO command in the layout or schematic. *New-lbr-name* stands for the library from which you want to take elements. You may add paths as well. Please see the help function for more information.

In many cases you will be asked whether gates, pins or pads should be replaced according to name or according to position. This always happens if library objects are renamed, or if their position (sequence) is changed.

If too many changes are made in the library at one time (e.g. pin names and pin positions are changed) it is not possible to carry out an automatic adaptation. In such a case it is possible either to carry out the modifications to the library in two steps (e.g. first the pin names and then the pin positions), or the library element can be given a new name, so that it is not exchanged.

If Forward&Back Annotation is active, the components are replaced in the schematic diagram and in the layout at the same time.

You will find further information on the program's help pages.

After any library update, please carry out both an ERC on the schematic and a DRC on the layout!

Individual components can, for instance, be updated with the aid of the ADD command. If you use ADD to fetch a modified component from a library, you will be asked whether all the older definitions of this type should be updated.

After the update you can delete the component that you just fetched.

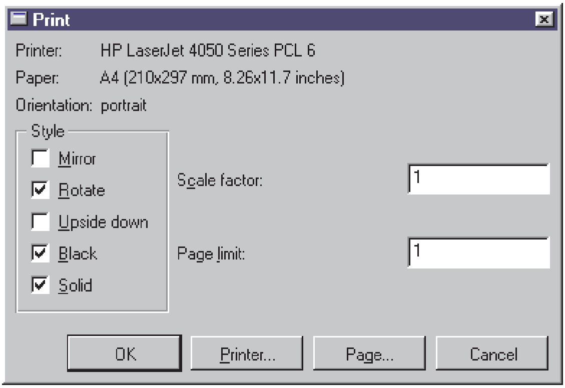
Again here it is wise to carry out an ERC and a DRC after the update!

6.6 Print Out Schematic and Layout

Schematic diagrams, boards and also library elements can be printed out with the PRINT command.

Using DISPLAY you should first select the layers that you want to print. The basic rule is: if you can see it in the editor, you will see it on the print.
Crosses marking coordinate origins for text, grid lines and polygons that can’t be calculated (only the polygon contours are visible in the Layout Editor after RATSNEST) are exceptions to this rule.

When the printer icon on the action toolbar is clicked, the PRINT dialog opens.



Print

Printer: HP LaserJet 4050 Series PCL 6

Paper: A4 (210x297 mm, 8.26x11.7 inches)

Orientation: portrait

Style

- □ Mirror
- ☑ Rotate
- □ Upside down
- ☑ Black
- ☑ Solid

Scale factor: 1

Page limit: 1

The PRINT window

The currently selected printer is shown at the top of the window. The selected printer can be altered by means of the *Printer...* button.

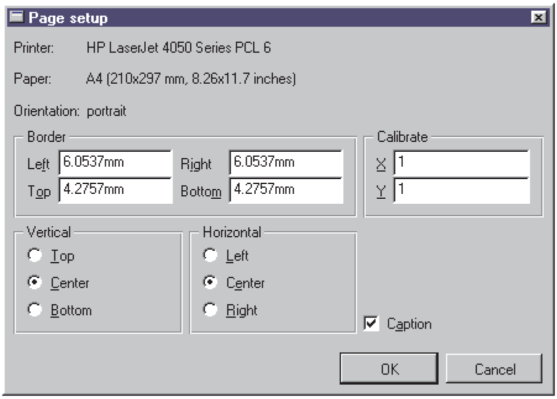
Style permits a number of output options to be selected:
Mirror inverts the drawing from left to right, *Rotate* turns it 90 degrees, and *Upside down* turns it through 180 degrees. If both are activated, a rotation of 270 degrees is the result.
If the *Black* option is chosen, a black-and-white printout is made.

Otherwise the print will be either in color or gray scale, depending on the printer.
Solid causes each element to be entirely filled. If you want to see the different filling patterns of the individual layers, then deactivate this option.

The *Scale factor* specifies the scale of the drawing.
If *Page limit* is set to 0, the printer will use whatever number of pages is needed to print the output at the selected scale. If a different value is selected, EAGLE will adjust the scale of the drawing to fit it onto the stated number of pages. This can mean that, under unfavorable circumstances, the selected scale cannot be used.

It is possible to select which sheets from a schematic diagram are printed using the *Sheets* box. This only appears in the Schematic Editor.

The *Page...* button brings you to another window that permits various page settings.



Page setup

Printer: HP LaserJet 4050 Series PCL 6

Paper: A4 (210x297 mm, 8.26x11.7 inches)

Orientation: portrait

Border
Left: 6.0537mm
Right: 6.0537mm
Top: 4.2757mm
Bottom: 4.2757mm

Calibrate
X: 1
Y: 1

Vertical
○ Top
● Center
○ Bottom

Horizontal
○ Left
● Center
○ Right

☑ Caption

PRINT: Page Setup

The edges of the print can be defined with the aid of the four entry boxes under *Border*. The values may be entered in mm or in inches. If you have changed the values and want to use the printer driver's standard settings again, simply enter a *0*.

The *Vertical* and *Horizontal* boxes allow the position of the printout on the sheet to be specified.

Calibrate allows correction factors for the x and y directions to be entered. This allows linear errors in the dimensional accuracy of the print to be corrected.

The *Caption* option switches the appearance of the title, printing date, filename and the scale of the print on or off.

If, when a layout is printed, the drill holes in the pads and vias are not to be visible, select the *No Drills* option for the *Display mode* by way of the menu item *Options/Set/Misc*.

The PRINT command can also be given directly on the command line, or can be run by a script file.

Information about the selection of options is available on the help pages.

For all those who have experiences with previous EAGLE versions:
EAGLE comes with an User Language Program named *cam2print.ulp* that converts your previous CAM jobs into script files in order to use them with the PRINT command.

Another User Language Program *cam2dxf.ulp* allows to use a previous CAM job for generating DXF data.

Chapter 7
The Autorouter

7.1 Basic Features

- Any routing grid (min. 0.02 mm)
- Any placement grid (min. 0.1 micron)
- SMDs are routed on both sides
- The whole drawing area can be the routing area (provided enough memory is available)
- The strategy is selected via control parameters
- Simultaneous routing of various signal classes with various track widths and minimum clearances
- Common data set (Design Rules) for the Design Rule Check and the Autorouter
- Multilayer capability (up to 16 layers can be routed simultaneously, not only in pairs)
- Full support of blind and buried vias
- The preferred track direction can be set independently for each layer: horizontal and vertical, true 45/135 degrees (important for inner layers!)
- Ripup and retry for 100 % routing strategy
- Optimization passes to reduce vias and smooth track paths
- Prerouted tracks are not changed

7.2 What Can be Expected from the Autorouter

The EAGLE Autorouter is a “100 %” router. This means that boards which, in theory, can be completely routed will indeed be 100 % routed by the Autorouter, provided - and this is a very important restriction - the Autorouter has unlimited time. This restriction is valid for all 100 % Autorouters whatsoever. However, in practice, the required amount of time is not always available, and therefore certain boards will not be completed even by a 100 % Autorouter.

The EAGLE Autorouter is based on the ripup/retry algorithm. As soon as it cannot route a track, it removes prerouted tracks (ripup) and tries it again (retry). The number of tracks it may remove is called ripup depth which is decisive for the speed and the routing result. This is, in principle, the previously mentioned restriction.

Those who expect an Autorouter to supply a perfect board without some manual help will be disappointed. The user must contribute his ideas and invest some energy. If he does, the Autorouter will be a valuable tool which will greatly reduce routine work.

Working with the EAGLE Autorouter requires that the user places the components and sets control parameters which influence the routing strategy. These parameters must be set carefully if the best results are to be achieved. They are therefore described in detail in this section.

7.3 Controlling the Autorouter

The Autorouter is controlled by a number of parameters. The values in the current Design Rules, the net classes and special Autorouter control parameters all have an effect.

The Design Rules specify the minimum clearances (DRC commands for setting *Clearance* and *Distance*), the via diameter (*Restring* setting) and the hole diameter of the vias (*Sizes* setting). The minimum track width is also specified.

The net classes - if any are defined - specify special minimum clearances, track widths and the hole diameters for vias carrying particular signals.

There is also a range of special cost factors and control parameters that can be changed via the Autorouter menu. They affect the route given to tracks during automatic routing. Default values are provided by the program. The control parameters are saved in the BRD file when the layout is saved. You can also save these values in an Autorouter control file

(**.ctl*). This allows a particular set of parameters to be used for different layouts. Neither Design Rules nor the data for various net classes are part of the control file.

A routing process involves a number of separate basic steps:

Bus Router

Normally the bus router (whose parameters are set such that the bus structures can be optimally routed) is the first pass. This step may be omitted.

Busses, as understood by the Autorouter, are connections which can be laid as straight lines in the x or y direction with only a few deviations. Busses are only routed if there is a layer with an appropriate preferred direction.

The bus router takes only those signals into consideration that belong to net class 0.

Routing Pass

The actual routing pass is then started, using parameters which make a 100 % routing as likely as possible. A large number of vias are deliberately allowed to avoid paths becoming blocked.

Optimization

After the main routing pass, any number of optimization passes can be made. The parameters are then set to remove superfluous vias and to smooth the track paths. In the optimization passes tracks are removed and rerouted one at a time. This can, however, lead to a higher degree of routing, since it is possible for new paths to be freed by the changed path of this track.

The number of optimization passes must be specified before starting the Autorouter. It is not possible to optimize at a later stage. Once the routing job has been completed all the tracks are considered to have been prerouted, and may no longer be changed.

Any of the steps mentioned above may be separately activated or deactivated.

7.4 What Has to be Defined Before Autorouting

Design Rules

The Design Rules need to be specified in accordance with the complexity of the board and of the manufacturing facilities available. You will find a description of the procedure and of the meanings of the individual parameters in the section on *Specifying the Design Rules* on page 95.

Track Width and Net Classes

If you have not already defined various net classes in the schematic diagram you now have the opportunity, before running the Autorouter, of specifying whether particular signals are to be laid using special track widths, particular clearances are to be observed, or whether certain drill diameters are to be used for vias for particular signals. Please consult the help pages (CLASS command) or the section on *Specifying Net Classes* on page 93 for information about the definition of net classes.

If no special net classes are defined, the values from the Design Rules apply. The value *Minimum width* in the *Sizes* tab determines the track width, the values for minimum clearances/distances are taken from the *Clearance* and *Distance* tabs. The diameter of vias is defined by the values in the *Restring* tab.

Did you set values in the Design Rules and for net classes? In this case the Autorouter uses the higher value.

Grid

The Design Rules determine the routing and placement grid. The minimum routing grid is 0.02 mm, which is about 0.8 mil.

Placement Grid

Although the Autorouter does permit any placement grid, it is not a good idea to place the components on a grid that is too fine. Two good rules are:

- The placement grid should not be finer than the routing grid.
- If the placement grid is larger than the routing grid, it should be set to an integral multiple of the routing grid.

These rules make sense if, for example, you consider that it might be possible, within the Design Rules, to route two tracks between two pins of a component, but that an inappropriate relationship between the two grids could prevent this (see diagram).

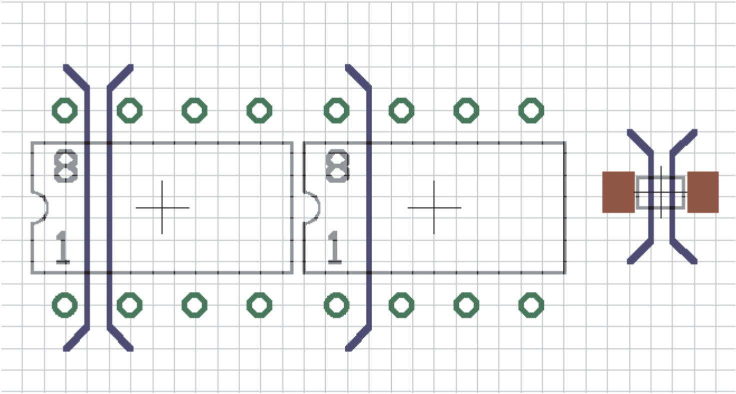
Routing Grid

The Autorouter grid has to be set in the AUTO command setup menu (Routing Grid). This is not the same as the currently used grid in the Layout Editor window that you have selected with the GRID command.

Bear in mind that for the routing grid the time demand increases exponentially with the resolution. Therefore select as large a grid as possible. The main question for most boards is how many tracks are to be placed between the pins of an IC. To answer this question, the selected Design Rules (i.e. the minimum spacing between tracks and pads or other tracks) must of course also be considered.

The result is:

The two grids must be selected so that component's pads are located on the routing grid.



Track patterns with different placement grids

There are of course exceptions, such as with SMDs to which the opposite may apply, namely that a position outside of the routing grid leads to the best results. In any event the choice of grid should be carefully considered in the light of the Design Rules and the pad spacing.

The example above may clarify the situation:

For the component on the left, the pads are placed on the routing grid. Two tracks can be routed between two pads. The pads of the component in the middle are not on the routing grid, and therefore only one track can be routed between them.

On the right you see the exception from the rule shown for SMD pads, which are placed between the routing grid lines so that one track can be routed between them.

When choosing the grid, please also ensure that each pad covers at least one grid point. Otherwise it can happen that the Autorouter is unable to route a signal, even though there is enough space to route it. In this case the Autorouter issues the message *Unreachable SMD at x y* as it starts. The parameters *x* and *y* specify the position of the SMD pad.

The default value for the routing grid is 50 mil. This value is sufficient for simple through-hole layouts. Working with SMD components demands a finer routing grid. Usual values are 25, 12.5, 10, or 5 mil.

Please remember that finer values require significantly more routing memory.

Memory Requirement

The amount of routing memory required depends in the first place on the selected routing grid, the area of the board and the number of signal layers in which tracks are routed.

The static memory requirement (in bytes) for a board can be calculated as follows:

```
 number of grid points x number of signal layers x 2
```

Note that a supply layer generated by giving the appropriate name ($name) does not occupy any autorouting memory. On the other hand, a supply layer that has been created with the aid of one or more polygons consumes as much routing memory as any other signal layer.

Space is also required for dynamic data, in addition to the static memory requirement. The dynamic data require in a very rough estimate about 10% up to 100 % (in some cases even more!) of the static value. This depends heavily on the layout.

Total memory requirement (rough approximation):

```
    static memory x (1.1..2,0) [bytes]
```

This much RAM should be free before starting the Autorouter. If this is insufficient, the Autorouter must store data on the hard disk. This lengthens the routing time enormously, and should be avoided at all costs. Short accesses to the hard disk are normal, since the job file on the hard disk is regularly updated.

Try to choose the coarsest possible routing grid. This saves memory space and routing time!

Layer

If you want to design a double-sided board, then select Top and Bottom as route layers. You should only use the Bottom layer for a single-sided board. In the case of inner layers, it is helpful to use the layers from the outside to the inside, i.e. first 2 and 15 and so on.
Inner layers are converted to supply layers if they are renamed to $name, where *name* is a valid signal name. These layers are not routed. Supply layers with more than one signal can be implemented with polygons. These layers are treated as normal signal layers.

In the case of boards that are so complex that it is not certain whether they can be wired on two sides, it is helpful to define them as multilayer boards, and to set very high costs for the inner layers. This will cause the Autorouter to avoid the inner layers and to place as many connections as possible in the outer layers. It can, however, make use of an inner layer when necessary.

These settings are made in the Autorouter menu (see page 148).

The autorouter shows the message *Unreachable smd in layer...*, if a layer that contains smds is not active. Clicking *OK* starts the autorouter nevertheless. If you want to change the autorouter setup click *Cancel*.

Preferred Directions

On the two outside layers the preferred directions are normally set to 90 degrees from each other. For the inner layers it may be useful to choose 45 and 135 degrees to cover diagonal connections. Before setting the preferred direction it is well worth examining the board (based on the airwires) to see if one direction offers advantages for a certain side of the board. This is particularly likely to be the case for SMD boards.

Please also follow the preferred direction when pre-placing tracks. The defaults are vertical for the Top (red) and horizontal for the Bottom (blue).

Experience has shown that small boards containing mainly SMD components are best routed without any preferred direction at all (set * in the Autorouter setup). The router then reaches a usable result much faster.

Single sided boards should be routed without a preferred direction.

Restricted Areas for the Autorouter

If the Autorouter is not supposed to route tracks or place vias within certain areas, you can define restricted areas by using the commands RECT, CIRCLE, and POLYGON in the layers 41, *tRestrict*, 42, *bRestrict,* and 43, *vRestrict.*

tRestrict: Restricted areas for the Top layer.
bRestrict: Restricted areas for the Bottom layer.
vRestrict: Restricted areas for vias.

Such restricted areas can already be defined in a device package (around, for instance, the fixing holes for a connector, or for a flat-mounted transistor under which there should not be any tracks).

Wires drawn in layer 20, *Dimension*, are boundary lines for the Autorouter. Tracks cannot be laid beyond this boundary.
Typical application: board boundaries.

An area drawn in layer 20 can also be used as a restricted region for all signals. It should, however, be noted that this area should be deleted before sending the board for manufacture, since layer 20 is usually output during the generation of manufacturing data.

Cost Factors and Other Control Parameters

The default values for the cost factors are chosen on the basis of our experience in such a way as to give the best results.

The control parameters such as *mnRipupLevel*, *mnRipupSteps* etc. have also been set to yield the best results according to our experience.

We want to emphasize, that we recommend working with the default values. If you nevertheless do want to experiment with these parameters, please consider the description of the cost factors in the following section. In the case of many parameters even small alterations can have large effects.

7.5 How the Cost Factors Influence the Routing Process

Values between 0..99 are possible for each cost factor (cfXXX), but the full range is not useful for all parameters. Sensible values are therefore given with each parameter. The control parameters (mnXXX) accept values in the range 0..9999. Reasonable figures are also provided under each parameter.

cfBase.xx: 0..20

Base costs for one step on the corresponding layer. Recommendation: outside layers (Top, Bottom) always 0, inside layers greater than 0.

cfVia: 0..99

Controls the use of vias. A low value produces many vias but also allows the preferred direction to be followed. A high value tries to avoid vias and thus violates the preferred direction. Recommendation: low value for the routing pass, high value for the optimization.

cfNonPref: 0..10

Controls following of the preferred direction. A low value allows tracks to be routed against the preferred direction, while a high value forces them into the preferred direction.
If *cfNonPref* is set to 99, track sections can only be placed in the preferred direction. Only select this value if you are certain that this behavior is really wanted.

cfChangeDir: 0..25

Controls how often the direction is changed. A low value means many bends are allowed within a track. A high value produces virtually straight tracks.

cfOrthStep, cfDiagStep

Implements the rule that the hypotenuse of a right-angled triangle is shorter than the sum of the other two sides. The default values are 2 and 3. That means that the costs for the route using the two other sides are 2+2, as against 3 for the hypotenuse.
This parameter should only be altered with great care!

cfExtdStep: 0..30

Controls the avoidance of track sections which run at an angle of 45 degrees to the preferred direction, and which would divide the board into two sections. A low value means that such sections are allowed while a high value tries to avoid them. In combination with the parameter *mnExtdStep* you can control the length of these tracks. If *mnExtdStep* = 0, each grid step at 45 degrees to the preferred direction causes costs that are defined in parameter *cfExtdStep*. Choosing for example *mnExtdStep* = 5 allows a track to run five steps at 45 degrees without any additional costs. Each further step causes costs defined in *cfExtdStep*.
In this way, 90 degree bends can be given 45 degree corners. Settings like *cfExtdStep* = 99 and *mnExtdStep* = 0 should avoid tracks with 45 degree

angles.
This parameter is only relevant to layers which have a preferred direction. Recommendation: use a lower value for the routing pass, and a higher value for the optimization.

cfBonusStep, cfMalusStep: 1..3

Strengthens the differentiation between *preferred* (*bonus*) and *bad* (*malus*) areas in the layout. With high values, the router differentiates strongly between *good* and *bad* areas. When low values are used, the influence of this factor is reduced. See also *cfPadImpact, cfSmdImpact*.

cfPadImpact, cfSmdImpact: 0..10

Pads and SMDs produce *good* and *bad* sections or areas around them in which the Autorouter likes (or does not like) to place tracks. The *good* areas are in the preferred direction (if defined), the *bad* ones perpendicular to it. This means that tracks which run in the preferred direction are routed away from the pad/SMD. With high values the track will run as far as possible in the preferred direction, but if the value is low it may leave the preferred direction quite soon.
It may be worth selecting a somewhat higher value for *cfSmdImpact* for densely populated SMD boards.

cfBusImpact: 0..10

Controls whether the ideal line is followed for bus connections (see also *cfPadImpact*). A high value ensures that the direct line between start and end point is followed. Only important for bus routing.

cfHugging: 0..5

Controls the hugging of parallel tracks. A high value allows for a strong hugging (tracks are very close to each other), a low value allows for a more generous distribution. Recommendation: higher value for routing, lower value for the optimization.

cfAvoid 0..10

During the ripup, areas are avoided from which tracks were removed. A high value means strong avoidance.
Not relevant to the optimization passes.

cfPolygon 0..30

Every step within a polygon is associated with this value. A low value allows routing inside a polygon. The probability that the polygon is broken into several pieces is higher. A higher value causes the Autorouter to make as few connections as possible inside a polygon.

mnVia 0..30

Controls the maximum number of vias that can be used in creating a connecting track.

mnSegments 0..9999

Determines the maximum number of wire pieces in one connecting track.

mnExtdSteps 0..9999

Specifies the number of steps that are allowed at 45 degrees to the preferred direction without incurring the value of *cfExtdStep*.
See also *cfExtdStep*.

7.6 Number of Ripup/Retry Attempts

Due to the structure of the Autorouter there are some parameters which influence the ripup/retry mechanism. They are set in such a way that they offer a good compromise between time demand and routing result. The user should therefore only carefully change the values for *mnRipupLevel*, *mnRipupSteps* and *mnRipupTotal* when needed.

As a rule, high parameter values allow for many ripups but result in increased computing times.

To understand the meaning of the parameters you need to know how the router works.

To begin with the tracks are routed one after the other until no other path can be found. As soon as this situation occurs, the router removes up to the maximum number of already routed tracks (this number has been defined with *mnRipupLevel*) to route the new track. If there are eight tracks in the way, for example, it can only route the new track if *mnRipupLevel* is at least eight.

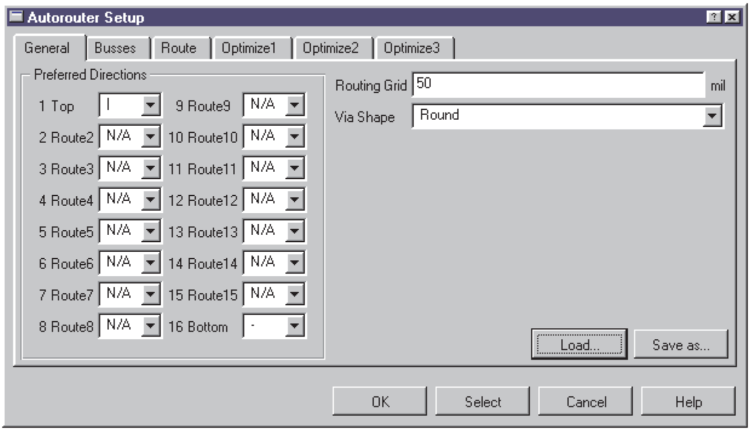
After routing the new track, the router tries to reroute all the tracks which were removed. It may happen that a new ripup sequence must be started to reroute one of these tracks. The router is then two ripup sequences away from the position at which, because of a track which could not be routed, it started the whole process. Each of the removed tracks which cannot be rerouted starts a new ripup sequence. The maximum number of such sequences is defined with the *mnRipupSteps* parameter.

The parameter *mnRipupTotal* defines how many tracks can be removed simultaneously. This value may be exceeded in certain cases.

If one of these values is exceeded, the router interrupts the ripup process and reestablishes the status which was valid at the first track which could not be routed. This track is considered as unroutable, and the router continues with the next track.

7.7 The Autorouter Menu

When running the Autorouter with the AUTO command, the setup menu appears first. All the necessary settings are made there.



Preferred Directions

1 Top	|	9 Route9	N/A
2 Route2	N/A	10 Route10	N/A
3 Route3	N/A	11 Route11	N/A
4 Route4	N/A	12 Route12	N/A
5 Route5	N/A	13 Route13	N/A
6 Route6	N/A	14 Route14	N/A
7 Route7	N/A	15 Route15	N/A
8 Route8	N/A	16 Bottom	-

Routing Grid 50 mil

Via Shape Round

Autorouter setup: General settings (*General* tab)

This is where you specify the layers that may be used for routing and which preferred directions apply. Click in the appropriate combo box with the mouse, and select the desired value.

Setting the preferred directions:

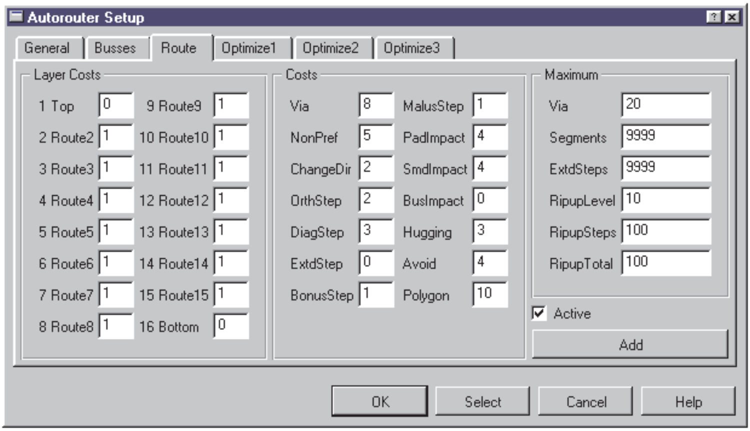
`-`	`horizontal`
`|`	`vertical`
`/`	`diagonal at 45°`

```
\   diagonal at 135°
*   none
N/A  Layer not active
```

Define a suitable *Routing grid* and select the shape of the through-plated holes (*Via shape*).

You may use the *Load...* and *Save as....* buttons to load a different parameter set from an Autorouter control file (**.ctl*) or to save the current settings for further projects.

You now make the settings for the individual routing steps. Click on the corresponding tab and make any settings you may want.



Layer Costs

1 Top	0	9 Route9	1
2 Route2	1	10 Route10	1
3 Route3	1	11 Route11	1
4 Route4	1	12 Route12	1
5 Route5	1	13 Route13	1
6 Route6	1	14 Route14	1
7 Route7	1	15 Route15	1
8 Route8	1	16 Bottom	0

Costs

Via	8	MalusStep	1
NonPref	5	PadImpact	4
ChangeDir	2	SmdImpact	4
OrthStep	2	BusImpact	0
DiagStep	3	Hugging	3
ExtdStep	0	Avoid	4
BonusStep	1	Polygon	10

Maximum

Via	20
Segments	9999
ExtdSteps	9999
RipupLevel	10
RipupSteps	100
RipupTotal	100

☑ Active

Autorouter setup: Settings for the Route pass

The parameters in the *Layer costs*, *Costs* and *Maximum* groups can be different for each pass.

The *Active* check box specifies whether this step should be executed or not.

Additional optimization passes can be inserted with the *Add* button.

Clicking on the *Select* button allows certain signals to be selected for autorouting. Select these with a mouse click, or enter their names on the command line. Then start the Autorouter with a click on the traffic-light icon in the action toolbar. It is, alternatively, possible to enter the signals on the command line while the AUTO command is active. Examples:

```
VCC GND ;
```

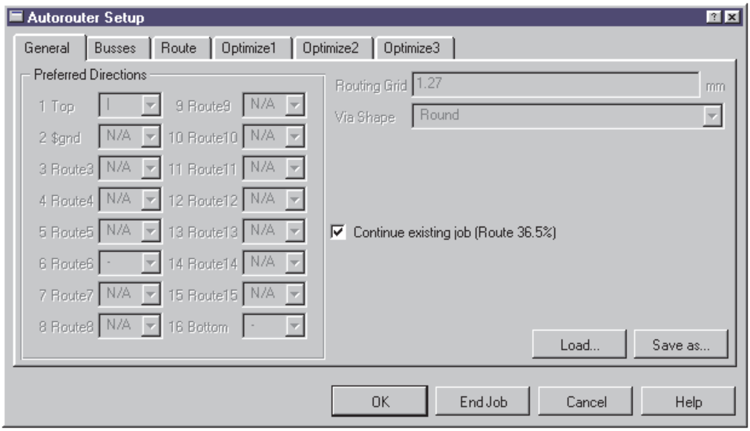
The signals VCC and GND are routed. The semicolon at the end of the line starts the Autorouter immediately. It is alternatively possible to click on the traffic-light icon.

```
! VCC GND ;
```

All signals except VCC and GND are routed.

The Autorouter is started for all the signals that have not yet been laid by clicking on the *OK* button.

The *Cancel* menu item interrupts the AUTO command without storing any changes.



Preferred Directions

1 Top	|	9 Route9	N/A
2 $gnd	N/A	10 Route10	N/A
3 Route3	N/A	11 Route11	N/A
4 Route4	N/A	12 Route12	N/A
5 Route5	N/A	13 Route13	N/A
6 Route6	-	14 Route14	N/A
7 Route7	N/A	15 Route15	N/A
8 Route8	N/A	16 Bottom	-

Routing Grid 1.27 mm

Via Shape Round

☑ Continue existing job (Route 36.5%)

Autorouter setup: Restarting an interrupted job

Do not at first make any changes to the parameters if you want to restart an interrupted routing job. Use the *Continue existing job* check box to decide whether you want to continue with an existing job, or whether you want to choose new settings for the remaining signals.

End job ends the autorouting job and loads the previous routing result.

7.8 Routing Multi-Layer Boards

There are two different ways of implementing supply layers.

- Supply layers with a $-sign at the start of the layer name
- Supply layers with polygons

Supply Layers

The definition of supply layers has already been described in the previous section under *Power Supply Layers with One Signal* on page 111.

The following points must be considered when autorouting:

- The layer name specifies the signal that is carried by this layer. The layer $VCC, for example, only carries the VCC signal.
- The layer is represented with an inverse display.

- The layer is not active during autorouting (setting *N/A* in the Autorouter setup, *General* tab).
- After routing, you should draw an isolation wire around the layout. This prevents short-circuits at the edges of the board.
- No other signals or polygons may be drawn!

The Autorouter can't use supply layers in combination with blind and buried vias. Use inner layers with polygons instead!

Polygons as Supply Layers

It is possible with polygons to create supply layers that contain more than one supply voltage, and a few individual wires as well. Please note the instructions on page 121, *Ground Planes and Supply Layers with Several Signals.* These are not the kind of supply layer identified by a $ in the name, but are in fact ordinary layers.

- Define the polygons before running the Autorouter.
- Give the appropriate signal names to the polygons.
- Select the preferred directions and base costs (*cfBase*) for the layer in the Autorouter setup. A higher value of *cfBase* for the polygon layer causes the Autorouter to avoid these layers more strongly.
- After routing, check that the polygon still connects all the signal points. It is possible that the polygon was divided as a signal was laid. RATSNEST recomputes polygons, and issues the message *Ratsnest: Nothing to do!*, if everything is in order.

7.9 Backup and Interruption of Routing

As, with complex layouts, the routing process may take several hours, a backup is carried out at intervals (approx. every 10 minutes). The file *name.job* always contains the last status of the job. If the job is interrupted for any reason (power failure etc.) the computer time invested so far is not lost, since you can recall the status saved in *name.job*. Load your board file in the Layout Editor, and then enter:

```
AUTO;
```

Answer the prompt as to whether the Autorouter should recall (*Continue existing job?*) with *Yes*. The Autorouter will then continue from the position at which the job was last saved (a maximum of 10 minutes may be lost).

If the autorouting is interrupted via the stop icon, the *name.job* file also remains intact and can be recalled. This may be useful when you have started a complex job on a slow computer and want to continue with it on a fast computer as soon as one is available.

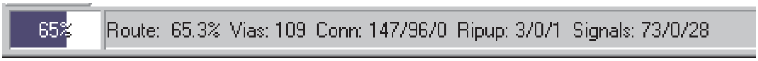
Please note that changing the parameters before recalling will not influence the job, since it will have been saved together with the parameters which were valid at the time of the initial Autorouter start.

When the Autorouter has finished, the routed board is saved as *name.b$$*. You can rename it to *name.brd* and use it, for instance, if a power failure occurred after the autorouting run and you could not save the board file. This file is deleted automatically after the board has been saved.

7.10 Information for the User

Status Display

During the routing, the Autorouter displays information on the actual routing result in the status bar.



65% Route: 65.3% Vias: 109 Conn: 147/96/0 Ripup: 3/0/1 Signals: 73/0/28

The displayed values have the following meaning:

```
Route:   result % (hitherto maximum, best data
Vias:    Number of vias
Conn.:   Connections total/found/not routable
Ripup:   No. of ripups/cur. RipupLevel/cur. RipupTotal
Signals: Signals found/signals handled/signals prepared
```

Connections means 2-point connections.

The significance of the various RIPUP options:

Number of ripups:
This indicates the number of connections that have already been routed during the foregoing routing procedure that have been (can be) removed in order to be able to route new signals.

Current RipupLevel:
This indicates the number of connections that have been removed or converted in airwires in order to lay the track for the current signal.

Current RipupTotal:
After a signal's routes have been ripped up it can be broken down into a large number of two-point connections. These connections are then routed again. This variable indicates the number of such two-point connections still to be routed.

Log file

For each routing pass the Autorouter generates a file called *name.pro*, containing useful information.

Example:

```
EAGLE AutoRouter Statistics:

Job           : d:/eagle4/test-design/democpu.brd

Start at      :   15.43.18  (24.07.2000)
End at        :   16.17.08  (24.07.2000)
Elapsed time  :   00.33.48

Signals       :    84   RoutingGrid: 10 mil  Layers: 4
Connections   :   238   predefined:  0 ( 0 Vias )

Router memory :   1121760

Passname:       Busses    Route   Optimize1 Optimize2 Optimize3 Optimize4

Time per pass: 00.00.21 00.08.44  00.06.32  00.06.15  00.06.01  00.05.55
Number of Ripups: 0        32         0         0         0         0
max. Level:       0         1         0         0         0         0
max. Total:       0        31         0         0         0         0

Routed:          16       238       238       238       238       238
Vias:             0       338       178       140       134       128
Resolution:     6.7 %   100.0 %   100.0 %   100.0 %   100.0 %   100.0 %

Final:        100.0% finished
```

7.11 Parameters of a Control File

We see here how the individual parameters in an Autorouter control file (*name.ctl*) are used.

Parameter	Default	Meaning
RoutingGrid	= 50Mil	Grid used by the Autorouter for tracks and via-holes
		Cost factors for...
cfVia	= 8	...Vias
cfNonPref	= 5	...Not using preferred direction
cfChangeDir	= 2	...Changing direction
cfOrthStep	= 2	...0 or 90 deg. step
cfDiagStep	= 3	...45 or 135 deg. step
cfExtdStep	= 30	...Deviation 45 deg. against preferred direction
cfBonusStep	= 1	...Step in bonus area
cfMalusStep	= 1	...Step in handicap area
cfPadImpact	= 4	...Pad influence on surrounding area
cfSmdImpact	= 4	...SMD influence on surrounding area
cfBusImpact	= 4	...Leaving ideal bus direction
cfHugging	= 3	...Wire hugging
cfAvoid	= 4	...Previously used areas during ripup
cfPolygon	= 10	...Avoiding polygons
cfBase.1	= 0	Basic costs for a step in the given layer
cfBase.2	= 1	
...		
cfBase.15	= 1	
cfBase.16	= 0	

```
                            Maximum number of...
mnVias        =   20        ...Vias per connection
mnSegments    = 9999        ...Wire segments per connection
mnExtdSteps   = 9999        ...Steps 45 deg. against preferred direction
mnRipupLevel  =  100        ...Ripups per connection
mnRipupSteps  =  300        ...Ripup sequences per connection
mnRipupTotal  =  200        ...Ripups at the same time

                            Track parameters for...
tpViaShape    =  Round      ...Via shape (round or octagon)

PrefDir.1       =  |        Preferred direction in the given layer
PrefDir.2       =  0        Symbols:  0 - / | \ *
...                         0 : Layer not used for routing
PrefDir.15      =  0        * : No preferred direction
PrefDir.16      =  -        - : X is preferred direction
                            | : Y is preferred direction
                            / : 45 deg. is preferred direction
                            \ : 135 deg. is preferred direction
```

7.12 Practical Tips

This section presents you with some tips that have, over a period of time, been found useful when working with the Autorouter.

Look on these examples as signposts suggesting ways in which a board can be routed. None of these suggestions guarantee success.

General

The layer costs (*cfLayer*) should increase from the outer to the inner layers or be the same for all layers. It is unfavorable to use lower values in the inner layers than in the outer layers. This could increase the needed routing memory enormously.

The Autorouter can't layout wires as arcs.

The Autorouter can’t set micro vias!

Single-Sided Boards

There are two procedures, depending on the kind of layout:

In the simplest case, only layer 16, *Bottom*, is active. No preferred direction is defined. Select a suitable grid and run the Autorouter.

If the layout is rather more complex, it may be possible to achieve a usable result with special parameter settings. Please take a look at the project named *singlesided*, which can be found in the *eagle/projects/examples* directory. This example project comes with various control files (**.ctl), which* are optimized for singlesided routing.

The Autorouter may use the Top layer as well. The tracks laid there will be realized as wire bridges on the board. In layer 41, *tRestrict*, you can define restricted areas around the components and in regions where wire bridges are not allowed.

Feel free to experiment with the parameter settings for your layout.

SMD Boards With Supply Layers

The following procedure has been found effective:

The supply signals are routed first. In general, a short track is wanted from a SMD component to a via that connects to the inner layer.

Before altering the parameters, save the current (default) values in an Autorouter control file (CTL file). Click on the button *Save as..* in the *General* tab of the Autorouter setup window and input any name, for example, *standard.ctl*.

Now switch off the bus router and all the optimization passes in the Autorouter setup. Only the routing remains active. Alter the following cost factors:

```
cfVia = 0      Vias are wanted
mnVia = 1      Max. 1 via per connection
cfBase.1/16 = 30..99   Short tracks
                       in Top/Bottom
mnSegments  = 2..8     short tracks
```

Start the Autorouter, using the *Select* button, and choose the signals to be routed. After the routing pass it is possible, if appropriate, to optimize the result manually.

The rest of the connections are routed after this. Use AUTO to open the Autorouter setup menu, and load the previous stored control parameters with the *Load..* button (*standard.ctl*). Adjust the values to any special wishes you may have, and start the Autorouter.

This procedure is identical for both types of supply layer (automatically generated supply layer, or a supply layer generated with polygons).

What can be done if not all signals are routed?

If this happens, check your settings.

- Has a sufficiently fine routing grid been selected?
- Have the track widths got appropriate dimensions?
- Can the vias have smaller diameters?
- Have the minimum clearances been optimally chosen?

If it is either impossible or unreasonable to optimize these values any further, an attempt to achieve a higher level of routing may be made by increasing the ripup level. Observe the notes in the section on the *Number of Ripup/Retry Attempts* on page 147.

Chapter 8

Component Design Explained through Examples

When developing circuits with EAGLE, components are fetched from libraries and placed into the schematic or, if the Schematic Editor is not being used, into the layout. All the component information is then saved in the schematic or board file. The libraries are no longer needed for continued work with the data. So when you want to pass your schematic to a third party to have a layout made from it, you do not also have to supply the libraries. An alteration in a library has no effect on a schematic or board.

The most important procedures for designing components (devices) and working with libraries are explained from page 53 on. Please read this paragraph before you continue to read the current chapter!

Some practical examples follow, from which the effective application of the relevant commands and parameters will be seen. First we will take the example of a resistor and go through the whole process of designing a simple component.
The second example provides a full description of the definition of a complex component, including various package variants and technologies. After that we shall discuss the special features which have to be taken into account with more complicated components.

Starting at page 205 hints concerning library and device management can be found. How to create my own library? How to copy elements from one library into another?

First attempts at editing packages, symbols, or devices may result in the need to delete various library elements. To do this, use the REMOVE command (see page 208).

8.1 Definition of a Simple Resistor

First open a new library in the EAGLE Control Panel using the *File/New/Library* menu.
Alternatively you can type the command

```
OPEN
```

on the command line of the Schematic or Layout Editor windows. Then enter a library name in the file dialog. The library window opens.

Resistor Package

Define a New Package

Select the package editing mode via the icon in the action toolbar, and enter the package name *R-10* in the *New* field. Answer the question *Create new package 'R-10'?* with *Yes*. Later when creating a new symbol and a new device you will again have to answer the corresponding questions with *Yes*.

Set the Grid

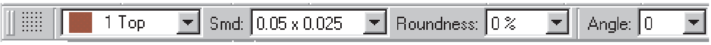
Use the GRID command to set an appropriate grid size for the pad placement. 0.05 inch (i.e. 50 mil) is usual for standard components with lead wires.

Solder Pads

For a resistor with lead-wires, select PAD, and set the pad shape and the drill diameter in the parameter toolbar. The default value for the pad diameter is *auto* (respectively 0). This value should be kept. The actual diameter is specified by the Design Rules for the layout. Then place two pads at the desired distance. The origin of the drawing will later be the identifying point with which a component is selected. For this reason it should be somewhere near the center of the device.

You should not draw any objects in layer 17, Pads, or 18, Vias! They will not be recognized, nor by the DRC, neither by polygons drawn in the layout, and can lead to short circuits!

For a SMD resistor, select SMD, and set the pad dimensions in the parameter toolbar. You can either select one of the offered values, or directly type the length and breadth into the entry field.



1 Top Smd: 0.05 x 0.025 Roundness: 0 % Angle: 0

SMD command: Parameter toolbar

All properties can be altered after placement using the CHANGE command or by typing the command directly on the command line.

Select *Top* as the layer, even if the component will later be placed on the underside of the board. SMD components are located on the other side of a board using the MIRROR command. This moves the elements in all the *t*..-layers into the corresponding *b*..-layers.

Place the two SMD pads (which in EAGLE are just called SMDs) at the desired distance. It may be necessary first of all to alter the grid setting to a suitable value. The SMD can be rotated with the right mouse button before it is placed.

The parameter ROUNDNESS specifies whether the corners of the SMDs are to be rounded. By default this value is set to 0% (no rounding). This value is usually kept, since the final roundness of SMDs is specified in the Design Rules. The help system provides you with more information about this parameter.

Angle determines the rotation of the SMD pad.

The INFO command provides you here with a quick summary of the current properties of a SMD or pad.

Pad Name

You can now enter the names, such as *1* and *2*, for the pads or SMDs using the NAME command.

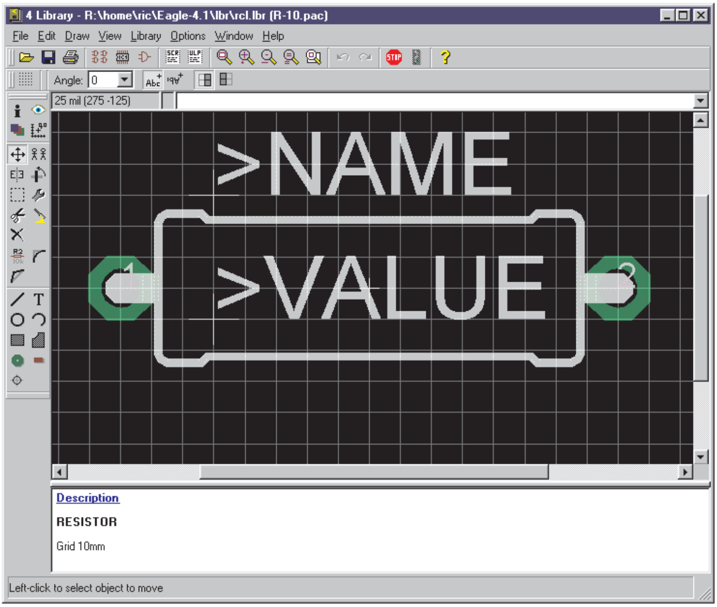
Silkscreen and Documentation Print

Now use the WIRE command to draw the silkscreen symbol in layer 21, *tPlace*. This layer contains what will be printed on the board. It is up to you how much detail you give to the symbol. Set a finer grid size if it helps.

Take the information provided in *library.txt* as a guideline for the design of components.. The line thickness for the silk screen is usually 0.01 inch (0.254 mm).

You can also use the ARC, CIRCLE, RECT and POLYGON commands to draw the symbols for the silkscreen.

Layer 51, *tDocu*, is not used to print onto the board itself, but is a supplement to the graphical presentation which might be used for printed documentation. Care must be taken in layer 21, *tPlace*, not to cover any areas that are to be soldered. A more realistic appearance can be given, however, in the *tDocu* layer, which is not subject to this limitation. In the example of the resistor, the symbol can be drawn in layer 21, *tPlace*, but the wires, which go over the pads, are drawn in layer 51, *tDocu*.


4 Library - R:\home\ric\Eagle-4.1\lbr\rcl.lbr (R-10.pac)
File Edit Draw View Library Options Window Help
Angle: 0
>NAME
>VALUE
1
2
Description

25 mil (275 -125)

RESISTOR

Grid 10mm

Left-click to select object to move

The Package Editor

Labeling

With the TEXT command you place the texts >NAME (in layer 25, *tNames*) and >VALUE (in layer 27, *tValues*) in those places where in the board the actual name and the actual value are to appear. 0.07 inch for the text height (size) and 10% for the ratio (relationship of stroke width to text height, which can only be set, using CHANGE, for vector fonts) are recommended.

SMASH and MOVE can be used later to change the position of this text relative to the package symbol on the board.

In the case of ICs, for instance, the value corresponds to what will later be the device name (e.g. *74LS00*).

When working with only the Layout Editor, the value is specified in the board.

Restricted area for components

In layer 39, *tKeepout*, you should create a restricted area over the whole component (RECT command). This allows the DRC to check whether components on your board are too close or even overlapping.

Description

Finally, you click on the *Description* box. Text can then be entered in the lower part of the window which then opens. Rich Text format can be used. This is a syntax somewhat similar to HTML, which permits formatting of the text. You will find detailed information in the help system under *Rich Text*.

Example:

```
<b>R-10</b>
<p>
Resistor 10 mm grid.
```

Keywords from this text can be searched for from the ADD dialog in the layout.

Do not forget to save the library from time to time!

Note

The CHANGE command can be used at a later stage to alter object properties such as the stroke thickness, text height, pad shape, or the layer in which the object is located.

If you want to change the properties of several objects at one go, define a group with the GROUP command, click the CHANGE command, select the parameter and the value, and click on the drawing surface with the right mouse button.

Example: Use GROUP to define a group that contains both pads, then select CHANGE and SHAPE/SQUARE. Click on the drawing surface with the right mouse button. The shape of both pads changes.

Resistor Symbol

Define a New Symbol

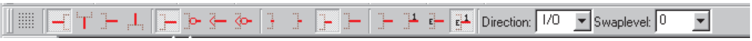
Select the symbol editing mode, and enter the symbol name *R* in the *New* field. This name only has a meaning internal to the program, and does not appear in the schematic.

Set the Grid

Now check that 0.1 inch is set as the grid size. The pins in the symbol **must** be placed on this grid, since this is what EAGLE expects.

Place the Pins

Select the PIN command. You can now set the properties of these pins in the parameter toolbar, before placing them with the left mouse button. All these properties can be changed at a later stage with the CHANGE command. Groups can again be defined (GROUP) whose properties can then be altered with CHANGE and the right mouse button. See also page 161.



Direction: I/O

Swaplevel: 0

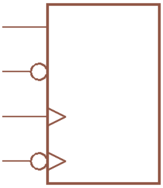
Pin command - Parameter toolbar

Orientation

Set the direction of the pins (*Orientation* parameter) using the four left-hand icons in the parameter toolbar or, more conveniently, by rotating with the right mouse button.

Function

The function parameter is set with the next four icons on the parameter toolbar. This specifies whether the symbol is to be shown with an inversion circle (Dot), with a clock symbol (Clk), with both (DotClk) or simply as a stroke (None). The diagram illustrates the four representations on one package.



Pin Functions

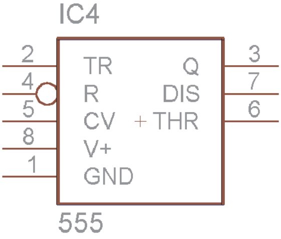
Length

The next four icons on the parameter toolbar permit setting of the pin length (0, 0.1 inch, 0.2 inch, 0.3 inch). The 0 setting is used if no pin-line is to be visible, or if, as in the resistor symbol, a pin shorter than 0.1 inch is desired. In that case the pin is to be drawn with the WIRE command as a line in layer 94, *Symbols*.

The SHOW command can be used to check whether a net is connected to a pin in the schematic diagram. The pin line and the net are displayed more brightly if they are connected. If a pin with length 0 is used, or if it was drawn as a line with the WIRE command, it cannot be displayed brightly.

Visible

The next four icons in the parameter toolbar specify whether the pins are to be labeled with pin names, pad names, both or neither. The diagram illustrates an example in which pin names are shown inside and pad-names outside. The location of the label relative to the pin is fixed. The text height is also fixed (at 60 mil).



Pin labeling

Direction

The Direction parameter specifies the logical direction of the signal flow:

NC	Not connected
In	Input
Out	Output
I/O	Input/output
OC	Open Collector or Open Drain
Hiz	High impedance output
Pas	Passive (resistors, etc.)
Pwr	Power pin (power supply input)
Sup	Power supply output for ground and power supply symbols

The electrical rule check is based on these parameters. It will flag, for example, if two pins whose direction is *out* are connected. Please note that the ERC can only offer warnings. You must interpret the messages yourself.

The *Pwr* and *Sup* directions are used for the automatic connection of supply voltages (see page 192).

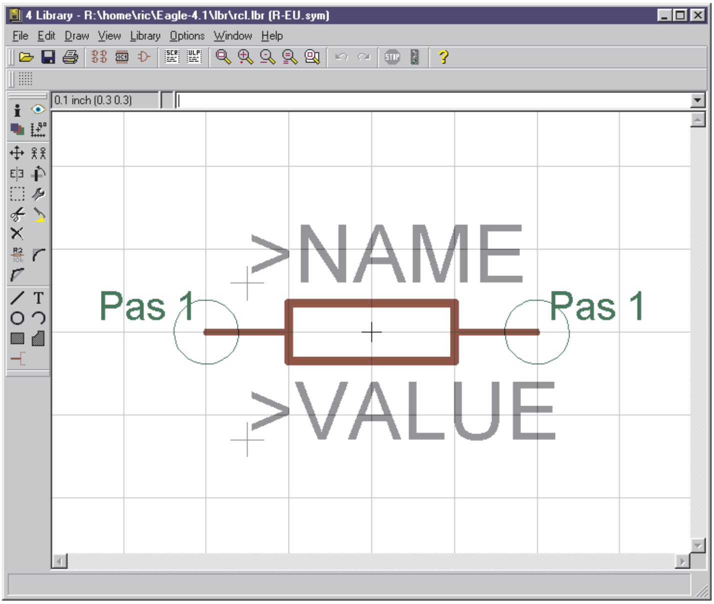
Swaplevel

The swaplevel is a number between 0 and 255. The number 0 means that the pin cannot be exchanged for another pin in the same gate. Any number bigger than 0 means that pins can be exchanged for other pins which have the same swaplevel and are defined within the same symbol. The pins can be swapped in the schematic or in the board with the PINSWAP command.

The two pins of a resistor can have the same swaplevel (e.g. 1), since they are interchangeable.

If the layer 93, *Pins*, is being displayed, the connection points on nets are shown with green circles. The *Direction* and *Swaplevel* parameters moreover (here *Pas* and *1*) are displayed in this layer.

The connections of a diode, for instance, cannot be exchanged, and are therefore given swaplevel 0.


4 Library - R:\home\ric\Eagle-4.1\lbr\rcl.lbr (R-EU.sym)
File Edit Draw View Library Options Window Help
0.1 inch (0.3 0.3)
>NAME
>VALUE
Pas 1
Pas 1

The Symbol Editor

Pin Names

The NAME command allows you to name pins after they have been placed. The automatic name allocation, as described on page 73 also operates.

Schematic Symbol

The schematic symbol is drawn in the symbols layer using WIRE and the other drawing commands. Place the texts >*NAME* and >*VALUE* in layers 95, *Names*, and 96, *Values* (TEXT command). Place them where the name and value of the component are to appear in the schematic.

Precise placement of the text can be achieved by setting the grid finer, which can even be done while the TEXT command is active. Afterwards, however, set the again grid to 0.1 inches.

Resistor Device

Define a New Device

Create the new device *R-10* with this icon. When you later use the ADD command to fetch the component into the schematic, you will select it by using this name. It is only a coincidence that in this case the name of the package and the name of the device are the same.

So enter the name *R* on the *New* line. The Device Editor opens after the confirming question *Create new device 'R'?*.

Selecting, Naming and Configuring Symbols

The previously defined resistor symbol is fetched into the device with the ADD command.
If a device consists of several schematic symbols which can be placed independently of one another in the circuit (in EAGLE these are known as *gates*), then each gate is to be individually brought into the schematic with the ADD command.

Set an addlevel of *Next* and a swaplevel of *0* in the parameter toolbar, and then place the gate near the origin. (There are further explanations about addlevel in the following sections.)

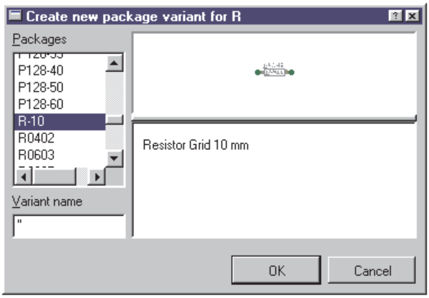
The swaplevel of a gate behaves very much like the swaplevel of a pin. The value of 0 means that the gate cannot be exchanged for another gate in the device. A value greater than 0 means that the gate can be swapped within the schematic for another gate in the same device and having the same swaplevel. The command required for this is GATESWAP.
Only one gate exists in this example; the swaplevel remains 0.

You can change the name of the gate or gates with the NAME command. The name is unimportant for a device with only one gate, since it does not appear in the schematic.
Keep the automatically generated name!

In the case of devices with several gates, the name of the particular gate is added to the name of the device.
Example:
The gates are called A, B, C and D, and the name of the component in the schematic is IC1, so the names which appear are IC1A, IC1B, IC1C and IC1D.

Selecting the Package

Now click on the *New* button at the lower right of the Device Editor window. Choose the R-10 package from the selection window, and enter a name for the version. If only one package version is used, it is usual to use two single quote marks (") for the name of the package version. It is, however, quite possible to assign a particular name.



Create new package variant for R

Packages: R-10

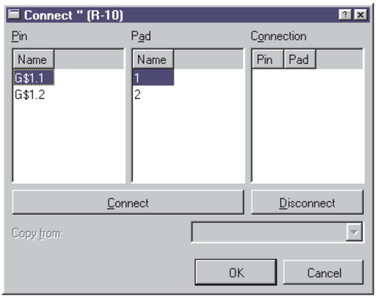
Variant name: "

Resistor Grid 10 mm

The package selection

Connect – Connections Between Pins and Pads

With the CONNECT command you specify which pins are taken to which package pads.



Connect " (R-10)

Pin: Name	Pad: Name	Connection: Pin	Connection: Pad
G$1.1	1		
G$1.2	2		

The CONNECT window

The resistor gate in this example is automatically identified as *G$1*, for which reason the pins *G$1.1* and *G$1.2* of this gate appear in the *Pin* column.
The two connections of the housing are listed in the *Pad* column. Mark a pin and the associated pad, and click on *Connect*.
If you want to undo a connection that you have made, mark it in the *Connection* column and click *Disconnect*.
Clicking on a column's header bar changes the sorting sequence.

Finish the CONNECT command by clicking on *OK*.

Define Prefix

The PREFIX command is used to specify a prefix for a name. The name itself will initially be automatically allocated in the schematic. For a resistor this would, naturally enough, be *R*. The resistors will then be identified as R1, R2, R3 etc..

The names can be altered at any time with the NAME command

Value

On: You are allowed to change the value in the schematic (for example for resistors). Without a value the part will not be specified exactly.

Off: The value will be generated from the device name and includes technology and package variant (e. g. 74LS00N), if available. Also recommended for supply symbols.

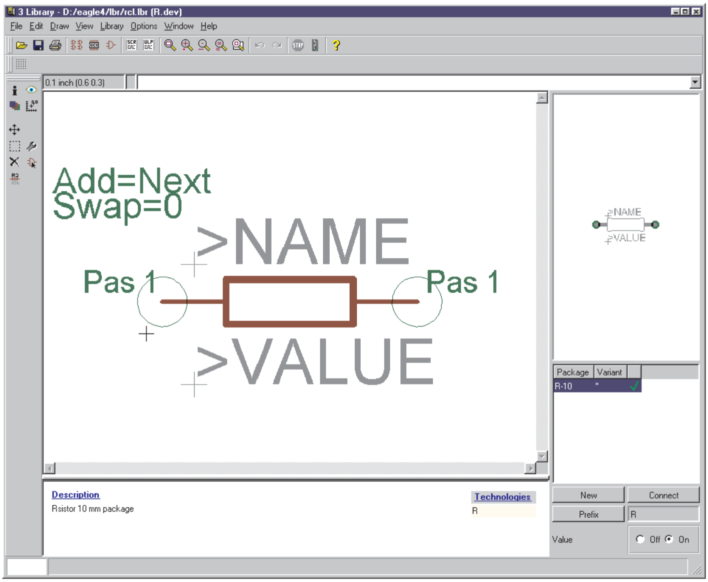
Description

Click on the *Description* in the description box. You can enter a description of the component here. The search facility of the ADD command in the schematic diagram will search through this text.

You can use the Rich Text format, as in the package description. You will find notes about this in the help system under the keyword *Rich Text*.

It can look like this:

```
<b>R-10</b>
<p>
Resistor 10mm package
```



The Device Editor: Fully defined resistor

Save

This completes definition of the resistor, and it can be fetched into the schematic diagram. If you have not already saved the library, please do it at this stage!

Use

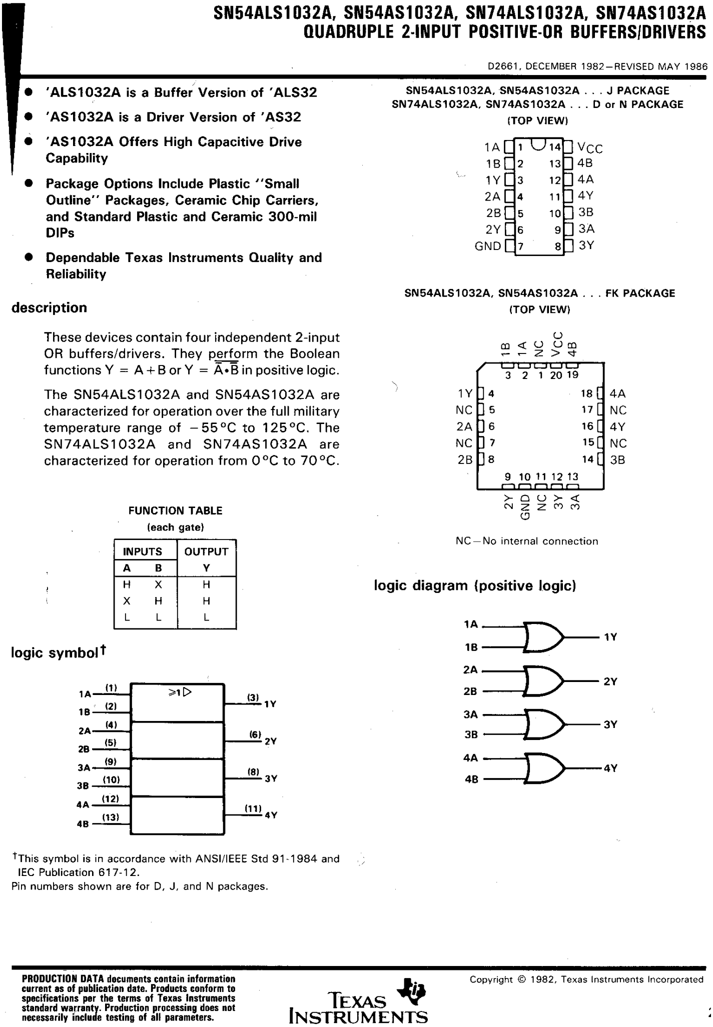
The newly created library has to be made available for the schematic or layout with the help of the USE command.
This command has to be used in the Schematic or Layout Editor. It is also possible to mark a library as *in Use* in the Control Panel's tree view. See help for details. Now the library will be recognized by the ADD command and its search function.

8.2 Defining a Complex Device

In this section we use the example of a TTL chip (541032) to define a library element that is to be used in two different packages (pin-leaded and SMD). It is a quad OR gate. The schematic diagram symbol is to be defined in such a way that the individual OR gates can be placed one after another. The power supply pins are not initially visible in the schematic diagram, but can be fetched into the diagram if needed.

The definition proceeds in the following steps:

- Creating a new library
- Drawing the pin-leaded housing (DIL-14)
- Creating the SMD housing (LCC-20)
- Defining the logic symbol
- Creating the power supply symbol
- Associating the packages and symbols to form a Device set



SN54ALS1032A, SN54AS1032A, SN74ALS1032A, SN74AS1032A QUADRUPLE 2-INPUT POSITIVE-OR BUFFERS/DRIVERS

D2661, DECEMBER 1982—REVISED MAY 1986

- 'ALS1032A is a Buffer Version of 'ALS32
- 'AS1032A is a Driver Version of 'AS32
- 'AS1032A Offers High Capacitive Drive Capability
- Package Options Include Plastic "Small Outline" Packages, Ceramic Chip Carriers, and Standard Plastic and Ceramic 300-mil DIPs
- Dependable Texas Instruments Quality and Reliability

description

These devices contain four independent 2-input OR buffers/drivers. They perform the Boolean functions Y = A + B or Y = $\overline{\overline{A} \cdot \overline{B}}$ in positive logic.

The SN54ALS1032A and SN54AS1032A are characterized for operation over the full military temperature range of −55 °C to 125 °C. The SN74ALS1032A and SN74AS1032A are characterized for operation from 0 °C to 70 °C.

SN54ALS1032A, SN54AS1032A . . . J PACKAGE
SN74ALS1032A, SN74AS1032A . . . D or N PACKAGE
(TOP VIEW)

SN54ALS1032A, SN54AS1032A . . . FK PACKAGE
(TOP VIEW)

NC—No internal connection

FUNCTION TABLE
(each gate)

INPUTS		OUTPUT
A	B	Y
H	X	H
X	H	H
L	L	L

logic diagram (positive logic)

logic symbol†

†This symbol is in accordance with ANSI/IEEE Std 91-1984 and IEC Publication 617-12.
Pin numbers shown are for D, J, and N packages.

PRODUCTION DATA documents contain information current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

TEXAS INSTRUMENTS

Data sheet for the 541032

All the data for this component has been extracted from a data book published by Texas Instruments, whom we thank for permission to reproduce it.

Creating a New Library

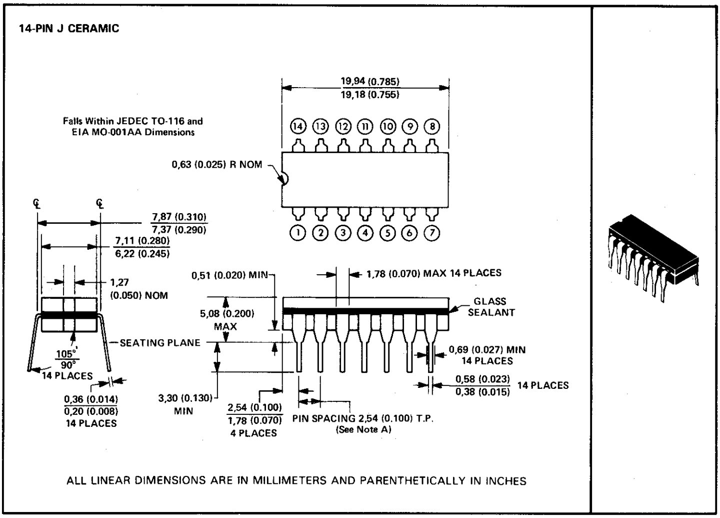
Click on the *File/New/Library* menu in the EAGLE Control Panel. The Library Editor window appears, containing a new library, *untitled.lbr*.

It is, of course, also possible to expand an existing library. In that case you would use *File/Open/Library* to select the library you want, or you would click on the *Libraries* entry in the Control Panel's tree view, selecting the desired library with a click of the right mouse button. This will open a context menu, one of whose options is *Open*. The Library Editor is opened.

Drawing the Pin-Leaded Package

The component is manufactured in a pin-leaded package. This is a DIL-14 housing with a pin spacing of 2.54 mm (0.1 inch) and a width of 7.62 mm (0.3 inch).

If there is a suitable package in another library, it can be copied into the current library. A new definition would not be necessary.



NOTE A: Each pin centerline is located within 0,25 (0.010) of its true longitudinal position.

DIL-14 data sheet

Click on the *Edit a package* icon in the action toolbar, and enter the name of the package in the *New* box of the *Edit* menu, which is *DIL-14* in our present example. Click on *OK*, and confirm the question *Create new package 'DIL-14'?* by answering *Yes*.

The Package Editor window now opens.

Set the Grid

First set the appropriate grid (50 mil in this case) using the GRID command, and let the grid lines be visible.
The grid can easily be shown and hidden with the F6 function key.

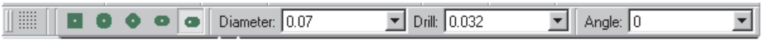
Place Pads

Use the PAD command, and place the solder pads in accordance with the specifications on the data sheet. The pads should be arranged in such a way that the coordinate origin is located somewhere near the center of the package.

Each pad can have individual properties such as *Shape*, *Diameter*, and *Drill* hole diameter. Available shapes are: *Square*, *Round*, *Octagon*, *Long*, and *Offset* (Long with offset drill).

Select the desired pad shape and specify the hole diameter.

The pad diameter usually is defined with the standard value *auto* (respectively 0), since the size is finally determined in the layout by means of the Design Rules, *Restring* tab. The pad appears in the library with the default value of 55 mil.
You may, however, assign an individual value. If, for instance, you specify 70 mil, the consequence is that the diameter of the pad on the board cannot be less than 70 mil (independent of the calculated value of the Design Rules). You select this value when the PAD command is active (i.e. the pad is attached to the mouse cursor) using the parameter toolbar. It is also possible to specify the drill hole diameter and the pad shape.



Diameter: 0.07 Drill: 0.032 Angle: 0

The parameter toolbar when the PAD command is active

The properties of pads that have already been placed can be altered at a later stage by means of the CHANGE command. Click on the CHANGE icon and select the property and the appropriate value. Then click on the pads whose properties are to be altered. CHANGE can also be applied to groups (using the GROUP command). After the property has been selected, click inside the group with the right mouse button.

As soon as a pad has been placed, EAGLE automatically generates solder stop symbols in layers 29 and 30, *t/bStop*. The dimensions of the solder stop symbols is specified in the Design Rules, *Mask* tab, *Stop* parameter.

Pads can be marked with special flags (*First*, *Stop*, *Thermals*). They can be altered with CHANGE subsequently. Giving one pad of a package the *First* flag (CHANGE FIRST ON) allows to define a special shape for it in the Design Rules, *Shapes* tab, option *First*, in order to mark it as the number '1' pad of the package.

Setting the *Thermals* flag *off* prevents generating a thermal symbol in a copper area.
CHANGE STOP OFF prevents automatic solder stop mask generation for a pad.

Pad Name

EAGLE automatically assigns pad names, *P$1, P$2, P$3* etc., as placement proceeds. Assign the names in accordance with the information in the data book.

The names can be checked easily by clicking the *Options/Set/Misc* menu and choosing the *Display pad names* option. All pad names are displayed after refreshing the screen (F2).
Alternatively type in the command line:

```
SET PAD ON
```

To hide the pad names again:

```
SET PAD OFF
```

The following procedure is recommended for components that have a large number of sequentially numbered pads:

Select the PAD command, type in the name of the first pad, e.g. '1', and place the pads in sequence. The single quote marks must be typed on the command line. See also the section on *Names and Automatic Naming* on page 73.

Draw the Silk Screen Symbol

A simple silk screen symbol that is to be visible on the board is drawn in layer 21, *tPlace*. Use the commands WIRE, CIRCLE, and ARC. Ensure that it does not cover soldered areas, since this can cause problems when the boards come to be soldered. If necessary, use the GRID command to set a finer grid or use the *Alt* key for the alternative grid (see GRID command). The standard width (CHANGE WIDTH) for lines in the screen print is 10 mil (0.254 mm), and should not be made thinner.

It is also possible to create an additional and rather better-looking silk screen for documentation purposes in layer 51, *tDocu*. This may indeed cover soldered areas, since it is not output along with the manufacturing data.

Package Name and Package Value

The labeling now follows. Use the TEXT command and write

```
>NAME
```

in layer 25, *tNames*, for the name placeholder, and

```
>VALUE
```

in layer 27, *tValues*, as the placeholder for the value, and place this at a suitable location. We use proportional font with a text height of 70 mil as default.

If you want to have texts upside down by a package rotation of 180°, you have to use the *Spin* flag (see help function for TEXT command).

The texts can be relocated at a later stage using SMASH and MOVE.

Areas Forbidden to Components

▨ ╱ In layer 39, *tKeepout*, you should create a restricted area over the whole component using the RECT command or draw a frame around the package with WIRE. This allows the DRC to check whether components on your board are too close or even overlapping.

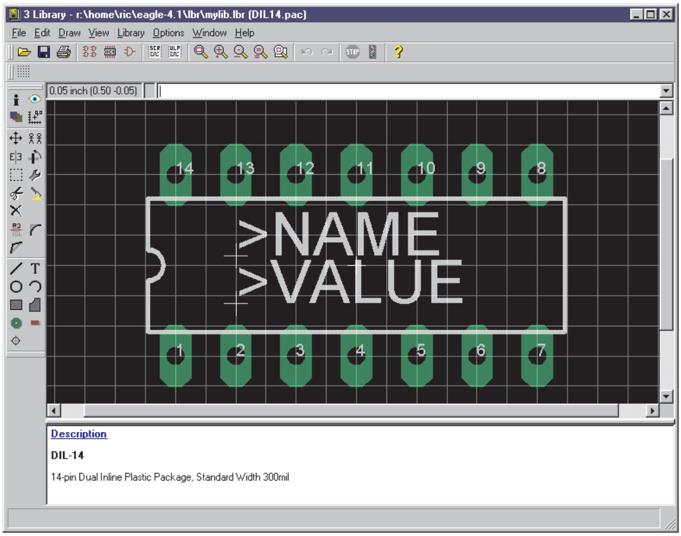
Description

Click on *Description* in the description box. A window opens in whose lower part it is possible to enter text, while the formatted appearance of the description is displayed in the upper part (*Headline*). The text can be entered in Rich Text format. This format works with a subset of HTML tags that allow the text to be formatted. You will find detailed information in the help system under *Rich Text*.

The descriptive text for our DIL-14 might look like this:

```
<b>DIL-14</b>
<p>
14-Pin Dual Inline Plastic Package, Standard
Width 300 mil
```

It is also possible to add, for instance, the reference data book,
the e-mail address of the source or other information here. The search facility in the Layout Editor's ADD dialog also looks in this text for keywords.


3 Library - r:\home\ric\eagle-4.1\lbr\mylib.lbr (DIL14.pac)
File Edit Draw View Library Options Window Help
0.05 inch (0.50 -0.05)
14
13
12
11
10
9
8
>NAME
>VALUE
1
2
3
4
5
6
7

Description

DIL-14

14-pin Dual Inline Plastic Package, Standard Width 300mil

Package Editor with DIL-14

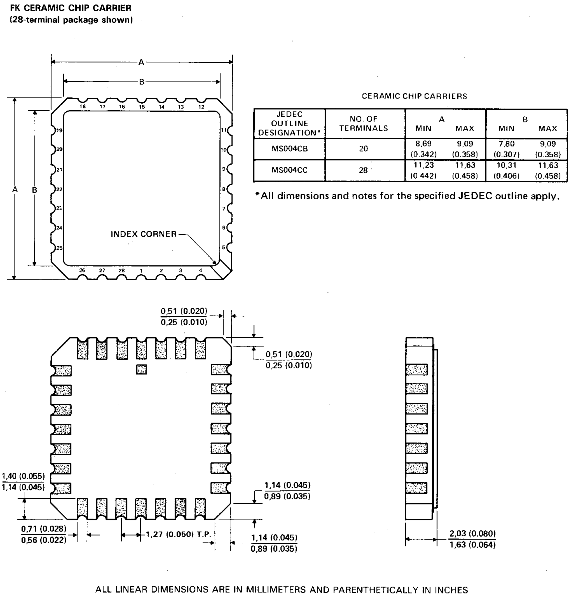
Save

At this stage if not before the library should be saved under its own name (e.g. *my_lib.lbr*).

Defining the SMD Package

The second type of housing for this component may be seen in the following scale drawing.

The size of the soldering areas is to be 0.8 mm x 2.0 mm. The SMD 1, at 0.8 mm x 3.4 mm, is larger.



FK CERAMIC CHIP CARRIER
(28-terminal package shown)

CERAMIC CHIP CARRIERS

JEDEC OUTLINE DESIGNATION*	NO. OF TERMINALS	A		B	
		MIN	MAX	MIN	MAX
MS004CB	20	8,69 (0.342)	9,09 (0.358)	7,80 (0.307)	9,09 (0.358)
MS004CC	28	11,23 (0.442)	11,63 (0.458)	10,31 (0.406)	11,63 (0.458)

*All dimensions and notes for the specified JEDEC outline apply.

ALL LINEAR DIMENSIONS ARE IN MILLIMETERS AND PARENTHETICALLY IN INCHES

SMD package, FK version.

Click again on the *Edit a package* icon, and enter the name of the package in the *New* box in the edit menu. The package is to be called LCC-20. Click on *OK* and confirm the question *Create new package 'LCC-20'?* by answering *Yes*.

Set the Grid

Adjust the grid to 0.635 mm (0.025 inch), and let the grid lines be visible. It is useful to define an alternative grid of 0.05 mm for designing this package.

Placing SMD Solder Pads

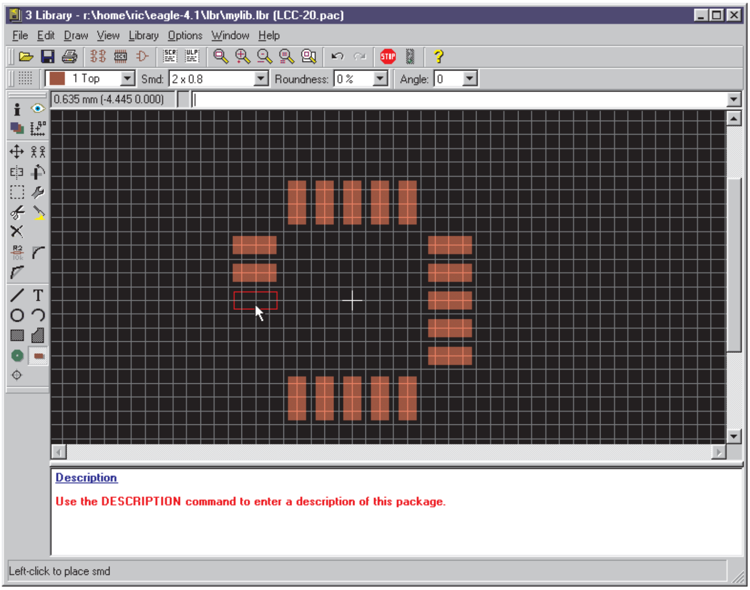
SMD devices are generally defined on the top of the board; SMDs are therefore always in layer 1, *Top*.
If you do want to have components on the solder side, the item is if needed reflected on the board with the MIRROR command. See also the section on page 199.

Begin by placing 5 SMDs at a distance of 1,27 mm from each other in two horizontal rows close to the coordinate origin. Since the value 0.8 x 2.0 is not contained in the SMD menu, this must be entered as 0.8 2.0, either on the command line or in the SMD box on the parameter toolbar.

Click therefore in the SMD icon, and type

```
0.8 2 ←
```

on the command line. Create two vertical rows as well. The SMDs can be rotated in 90 degree increments with the right mouse button.



3 Library - r:\home\ric\eagle-4.1\lbr\mylib.lbr (LCC-20.pac)

1 Top
Smd: 2 x 0.8
Roundness: 0 %
Angle: 0

0.635 mm (-4.445 0.000)

Description

Use the DESCRIPTION command to enter a description of this package.

Left-click to place smd

Placing the SMDs

The *Roundness* parameter (CHANGE command) specifies whether curves should be given to the corners of the solder pads. The default value is 0%, which means that there is no rounding.
See also the section on page 103.
If a square SMD is selected, and if *Roundness* is defined as 100%, the result is a round SMD, as is needed when creating ball grid array housings (BGA). *Roundness* is usually chosen to be 0% when a package is being defined. A general value can be specified in the Design Rules if slightly rounded solder pads are preferred.

Drag the 4 SMD rows into the correct position. Therfore use the finer alternative grid of 0.05 mm by pressing the *Alt* key. The commands GROUP and MOVE, followed by a right mouse click on the marked group can be used to drag the marked group into the correct position.

The size of the central SMDs in the upper row can be altered with the CHANGE SMD command. Since the value 0.8 x 3.4 is not contained in the menu as standard, type

```
    change smd 0.8 3.4 ←
```

onto the command line, then click on the SMD. Drag it with MOVE so that it is located at the correct position.

The INFO command is helpful for checking the position and properties of the solder pads.

When a SMD is placed (in the Top layer), symbols for solder stop and solder cream are automatically created in layer 29, *tStop*, and layer 31, *tCream*, respectively.
If the component in the layout is mirrored onto the bottom side, these are changed to the layers with the corresponding functions, namely 30, *bStop* and 32, *bCream*.

SMDs can have special flags (*Stop*, *Cream*, *Thermals*) that can be modified with the CHANGE command.
Setting the *Thermals* flag *off* avoids a thermal symbol for the SMD copper areas.
CHANGE STOP OFF or CHANGE CREAM OFF prevents EAGLE from generating a solder stop mask or a cream frame for the SMD automatically. See also help function.

SMD Names

If no names are visible in the SMD pads, click the *Options/Set/Misc* menu and activate the *Display pad names* option.

Alternatively you can type the following onto the command line:

```
    set pad_names on ←
```

All the names will be visible once the screen is redrawn (WINDOW command).

Use the NAME command to adjust the names to match the specifications of the data sheet.

It is alternatively possible to assign names as the SMDs are being placed, if the component has a large number of pads with sequential numbers. Select the SMD command, type in the name of the first SMD, e.g. '1', and place the pads in the correct sequence. The single quote marks must be entered on the command line. See also the section on *Names and Automatic Naming* on page 73.

You can also combine several statements on the command line, for example:

```
    smd 0.8 2 '1' ←
```

A SMD of 0.8 mm x 2.0 mm and with the name 1 is now attached to the mouse cursor.

Draw the Silk Screen

First set the grid to a suitable value such as 0.254 mm (10 mil). Draw the silk screen print in layer 21, *tPlace*. Note that it must not cover soldered areas, as this can cause problems when the boards come to be soldered.

Do not use a line width below the minimum of 0.2 mm. The standard value is 0.254 mm (10 mil).

It is also possible to create an additional, more detailed, silk screen for documentation purposes in layer 51, *tdocu*. This may indeed cover soldered areas, since it is not output along with the manufacturing data.

Package Name and Package Value

The labeling now follows. Use the TEXT command and write

```
>NAME
```

in layer 25, *tNames*, for the name placeholder, and

```
>VALUE
```

in layer 27, *tValues*, as the placeholder for the value, and place this at a suitable location. The texts can be separated and relocated at a later stage using SMASH and MOVE.

Area Forbidden to Components

In layer 39, *tKeepout*, you should create a forbidden area over the whole component (RECT command) or draw a frame around the package with the WIRE command. This allows the DRC to check whether components on your board are too close, or even overlapping.

Locating Point

As soon as you have finished drawing the package, please check where the coordinate origin is located. It should be somewhere near the middle of the package. If necessary, use GRID to choose a suitable grid (e.g. 0.635 mm), and shift the whole package with GROUP and MOVE.
First make sure that all the layers are made visible (DISPLAY ALL). That is the only way to be sure that all the elements have indeed been moved.

Description

Then click on *Description* in the description box. You can insert a detailed description of this package form here. The Rich Text format can be used. This format is described in the program's help system under *Rich Text* .

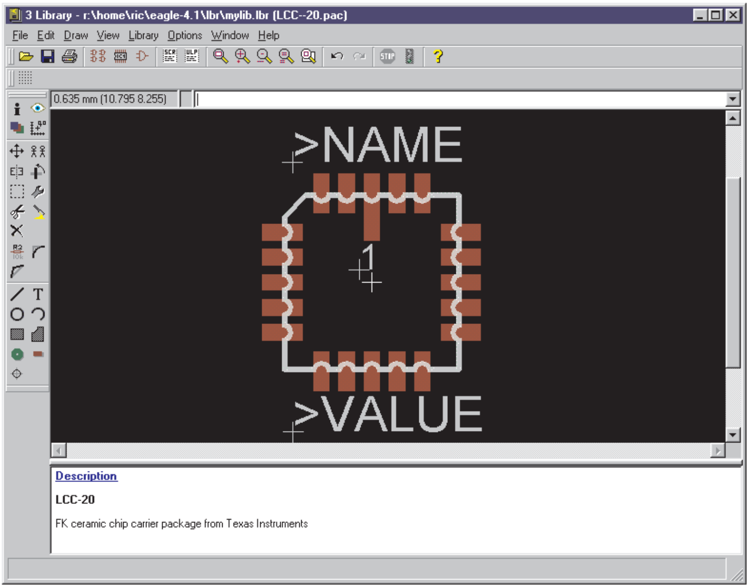
The entry of the LCC-20 in Rich Text format could look like this:

```
<b>LCC-20</b>
<p>
FK ceramic chip carrier package from Texas
Instruments.
```

The ADD dialog in the Layout Editor can search for this description or for keywords within it.

Save

Please do not forget to save the library from time to time!


3 Library - r:\home\ric\eagle-4.1\lbr\mylib.lbr (LCC--20.pac)
File Edit Draw View Library Options Window Help
0.635 mm (10.795 8.255)
>NAME
1
>VALUE

Description

LCC-20

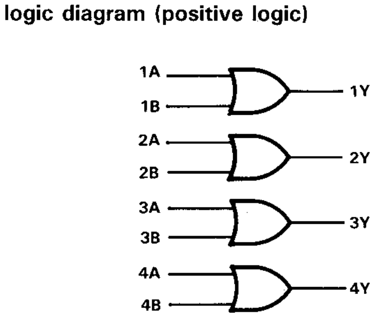
FK ceramic chip carrier package from Texas Instruments

The fully defined LCC-20

Supposed you found a package that is exactly the one you need, simply copy it into your current library. More information about this on page 205.

Defining the Logic Symbol for the Schematic Diagram

Our device contains four OR gates, each having two inputs and one output. We first create an OR symbol.



Logical appearance of the 541032.

Click on the *Edit a symbol* icon. Enter a name for the symbol on the *New* line, such as *2-input_positive_or*, and click *OK*. Confirm the question *Create new symbol '2-input_positive_or'?* by answering *Yes*. You now have the Symbol Editor window in front of you.

Check the Grid

Check that the grid is set to the default value of 0.1 inch. Please try to use only this grid, at least when placing the pins. It is essential that pins and net lines are located on the same grid. Otherwise there will not be any electrical connection between the net and the pin!

Place the Pins

Select the PIN command, and place 3 pins. The pin properties can be changed by means of the parameter toolbar as long as the pin is attached to the mouse cursor and has not been placed. If a pin has already been placed, its properties can be altered at a later stage with the CHANGE command. A number of pins can be handled at the same time with the GROUP and CHANGE commands followed by a click with the right mouse button. The parameters *Orientation*, *Function*, *Length*, *Visible*, *Direction* and *Swaplevel* have been thoroughly described when the example of the resistor symbol was examined (see p. 162).
The coordinate origin should be somewhere near the center of the symbol, and, if possible, not directly under a pin connection point. This makes it easy to select in the schematic diagram.

Pin Name

You assign pin names with the NAME command. In our symbol the two input pins are named *A* and *B*, and the output pin is named *Y*.

Draw the Symbol

Use the WIRE command to draw the symbol in layer 94, *Symbols*. The standard line thickness for the Symbol Editor is 10 mil. You may also choose any other line thickness.

Placeholders for NAME and VALUE

For the component labeling, use the TEXT command in the schematic diagram to write

```
>NAME
```

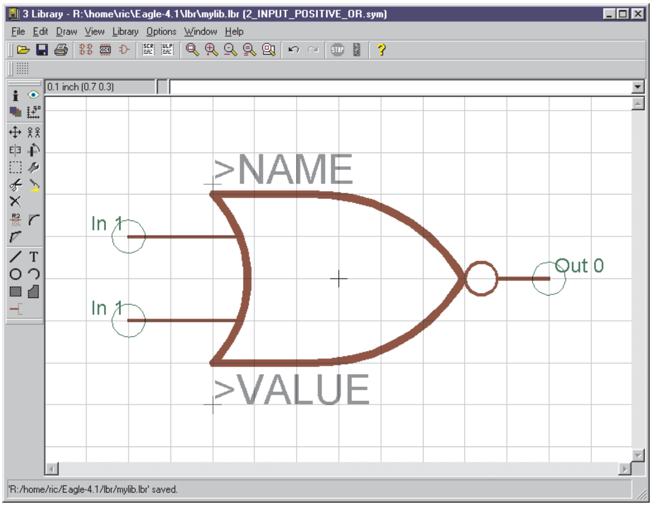
in layer 95, *Names* and

```
>VALUE
```

in layer 96, *Values*. Place the two texts at a suitable location. It is possible to move the texts again in the schematic diagram after using SMASH to separate it. The symbol should now have the appearance shown in the following diagram.

Save

This is a good moment to save the work that you have done so far.



3 Library - R:\home\ric\Eagle-4.1\lbr\mylib.lbr (2_INPUT_POSITIVE_OR.sym)

0.1 inch (0.7 0.3)

'R:/home/ric/Eagle-4.1/lbr/mylib.lbr' saved.

The Symbol Editor: Logic symbol (American representation)

Supposed you found a symbol that is exactly the one you need use GROUP, CUT, and PASTE to copy it into the current library. See also page 206.

Defining a Power Supply Symbol

Two pins are needed for the supply voltage. These are kept in a separate symbol, since they will not initially be visible in the schematic diagram.

Click on the *Edit a symbol* icon. Enter a name for the symbol on the *new* line, such as *VCC-GND*, and click *OK*. Confirm the question *Create new symbol 'VCC-GND'?* with *Yes*.

Check the Grid

First check that the grid is set to the default value of 0.1 inch. Only ever use this grid when placing pins!

Place the Pins

Fetch and place two pins with the PIN command. The coordinate origin should be somewhere near the center of the symbol.

Both pins are given PWR as their direction. To do this, click with the mouse on CHANGE, select the *Direction* option, and choose *PWR*. Now click on the two pins to assign this property.

The green pin label is updated, and now shows *Pwr 0*. It is only visible when layer 93, *Pins*, is active!

Pin Name

You use the NAME command to give the two pins the names of the signals that they are to carry. In this case, these are GND and VCC.

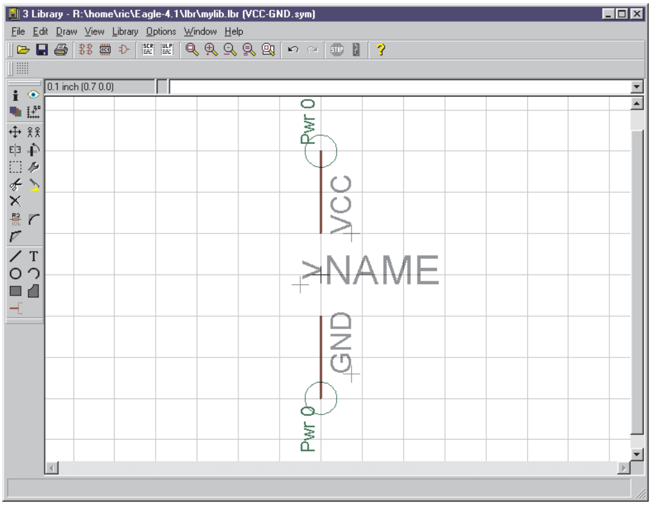
For reasons of appearance, the pin property *Visible* is set to *Pad* in the symbol shown below, and the pin label has been placed on layer 95, *Names*, using TEXT.

Placeholders for NAME and VALUE

For the component labeling, use the TEXT command in the schematic diagram to write

```
>NAME
```

in layer 95, *Names*. Place the text at a suitable location. No placeholder is necessary for value here.


3 Library - R:\home\ric\Eagle-4.1\lbr\mylib.lbr (VCC-GND.sym)
File Edit Draw View Library Options Window Help
0.1 inch (0.7 0.0)
Pwr 0
VCC
>NAME
GND
Pwr 0

The Symbol Editor: Supply Symbol

Associating the Packages and Symbols to Form a Device Set

We now come to the final step, the definition of the Device set. A Device set is an association of symbols and package variants to form real components

A Device set consists of several devices, which use the same symbols for the schematic but different technologies or package variants.

Defining a Device set or a device consists essentially of the following steps:

- Select symbol(s), name them and specify properties
- Assign package(s) or specify variants
- Specify the assignment of pins to pads using the CONNECT command
- Define technologies (if desired/necessary)
- State prefix and value
- Describe the device

Click on the *Edit a device* icon. Enter the name for the device on the *new* line.
In our example this is a 541032A. This device is to be used in two different technologies, as the 54**AS**1032A and as the 54**ALS**1032A. A * is

used as a placeholder at a suitable location in the device name to represent the different technologies. Enter, therefore, the name 54*1032A, and confirm the question *Create new device '54*1032A'*? with *Yes*.

The Device Editor window opens.

Select Symbols

First use ADD to fetch the symbols that belong to this device. A window opens in which all the symbols available in the current library are displayed. Double-click on the *2-input_positive_or* symbol and place it four times.
Click again on the ADD icon, and select the *'VCC-GND'* symbol from the list. Place this too onto the drawing area.

Naming the Gates

A symbol that is used in a device is known as a gate. They are automatically given generated names (G$1, G$2 etc.). This name is not usually shown on the schematic diagram.

It is nevertheless helpful to assign individual gate names when components are composed of a number of gates. To distinguish the individual OR gates, you use the NAME command to alter the gate names. Assign the names *A*, *B*, *C* and *D*, and name the power supply gate *P*.

Specify the Addlevels and Swaplevels

The addlevel can be used to specify how the gates are placed in the schematic diagram by the ADD command. You can see the current addlevel for each gate written above left in layer 93, *Pins*.

Assign the addlevel *Next* for gates A to D, and the addlevel *Request* to the power supply gate. Do this by clicking on the CHANGE icon, selecting the *Addlevel* entry, and then selecting the desired value for a gate. Then click on the gate that you want to change.

This means that as soon as the first OR gate has been placed on the schematic diagram, the next one is attached to the mouse cursor. All 4 gates can be placed one after another. The power gate does not automatically appear. You can, however, fetch it into the schematic diagram if necessary, by making use of the INVOKE command.

The parameter ADDLEVEL is described in full detail in the section entitled *More About the Addlevel Parameter* on page 195.

The swaplevel determines whether a device's gates can be swapped within the schematic diagram. The value that is currently set is – like the addlevel – displayed above left in layer 93, *Pins*, for each gate. The default value is 0, meaning that the gates may not be exchanged. The swaplevel can have a value from 0 to 255. Gates with the same swaplevel can be exchanged with one another.

Our device consists of four identical gates that may be swapped. Click on CHANGE, select the *Swaplevel* entry, and enter the value *1*. Click on the four OR gates. The information text in layer 93, *Pins*, changes correspondingly.

Choosing the Package Variants

In the Device Editor window, click on the *New* button at the lower right. A window opens that displays the packages defined in this library. Select the *DIL-14* package and give the version name *J*. Click on *OK*. Repeat this procedure, select the *LCC-20*, and give the version name *FK*.

In the list on the right you will now see the chosen package variants, with a simple representation of the selected package above it.
Clicking on a package variant entry with the right mouse button will open a context menu. This allows variants to be deleted, renamed or newly created, technologies to be defined, or the CONNECT command to be called.

Both entries are marked by a yellow symbol with an exclamation mark. This means that the assignment of pins and pads has not yet been (fully) carried out.

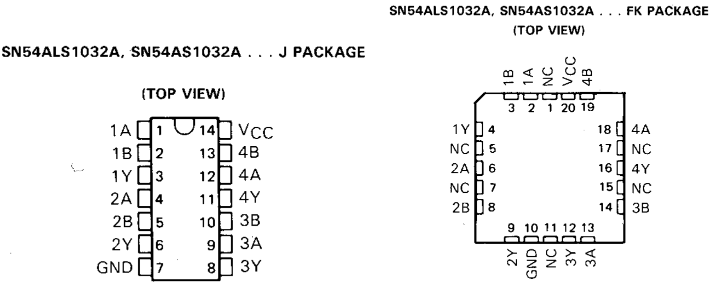
Supposed you don't find the appropriate package variant in the current library you may use packages from another library. Use the PACKAGE command to copy the package into the current variant and to define a new variant.
Example:

```
    PACKAGE DIL14@d:\eagle-4.1\lbr\dil.lbr J
```

This command copies the package named DIL14 *from* dil.lbr *into the current library. Simultaneously the variant* J *is generated for the device. See also page 199.*

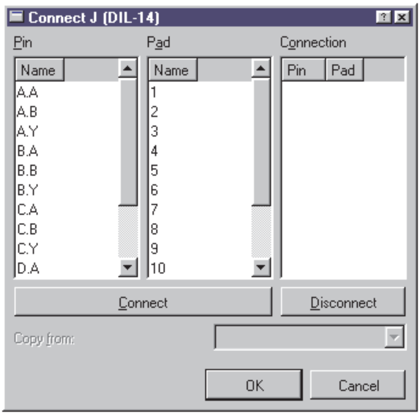
The Connect Command

This must be the most important step in the library definition. CONNECT assigns each pin to a pad. The way in which nets in the schematic diagram are converted into signal lines in the layout is defined here. Each net at a pin creates a signal line at a pad. The pin assignment for the 541032 is specified in the data sheet. Check the connects in the library with care. Errors that may pass unnoticed here can make the layout useless.



The pin assignment for the packages

Select the *J* version from the package list and click on the CONNECT button. The connect window opens.



Pin Name	Pad Name
A.A	1
A.B	2
A.Y	3
B.A	4
B.B	5
B.Y	6
C.A	7
C.B	8
C.Y	9
D.A	10

CONNECT dialog

The list of pins is on the left, and the pads are in the center. Click on a pin-entry, and select the associated pad. Both entries are now marked. You join them with the connect button. This pair now appears on the right, in the *Connection* column. Join each pin to its pad in accordance with the data sheet. Finish the definition by clicking on *OK*.

Define the connections for the second package version, *FK*, in the same way. Select the version, and click on the *Connect* button. The usual dialog appears in the connect window. Proceed exactly as described above. Please note that six pads are not connected in this version. They are left over in the *Pad* column. Finish the process by clicking on *OK*.

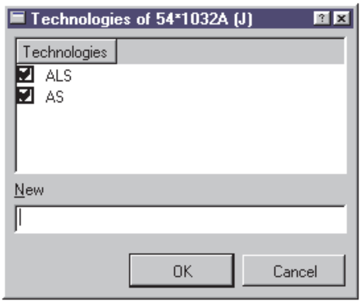
There is now a green tick to the right of both package variants, and this indicates that connection is complete. This is only true when every pin is connected to a pad.

One pin has to be connected to exactly one pad! It is not possible to connect several pins with a common pad! A device may contain more pads than pins, but not the other way around!
Pins with direction NC (not connected) must be connected to a pad, as well!

Defining Technologies

As noted above, the 541032 is to be used in two different technologies – AS and ALS –. By including a * as a placeholder in the device name we have already taken the first step towards this. In the schematic diagram the code for the chosen technology will appear instead of the *. The data sheet shows that both technologies are available in both packages.

Select the *J* package from the list on the right of the Device Editor window. Then click on *Technologies* in the description box. The technologies window opens. Define the technology in the *New* line, and confirm the entry with *OK*. When the entry has been completed, the *AS* and *ALS* entries are activated with a tick.



Technologies of 54*1032A (J)

Technologies

☑ ALS
☑ AS

New

Technologies for package version J

Close the window by clicking on *OK* again.

Select the FK version from the package list. Click on *Technologies* in the description box again. You will now see that AS and ALS are available as selections in the technologies window. Activate both of these by clicking in the small box to the left, so that a tick is displayed. Finish the definition by clicking on *OK*.

The technologies available for the selected package version are now listed in the description area of the Device Editor.

Specifying the Prefix

The prefix of the device name is defined simply by clicking on the *Prefix* button. *IC* is to be entered in this example.

Value

The setting of *value* determines whether the VALUE command can be used to alter the value of the device in the schematic diagram and in the layout. The default setting is *off*, so that alteration is not permitted. Since that appears to be appropriate here too, value is left *off*.

The value of the device corresponds to the device name in this case.

Description

Click on the *Description* in the description box. You can enter a description of the device in the window which now opens. Use typical terms that you might apply for a keyword search. The search facility of the ADD command in the schematic diagram will also search through this text.

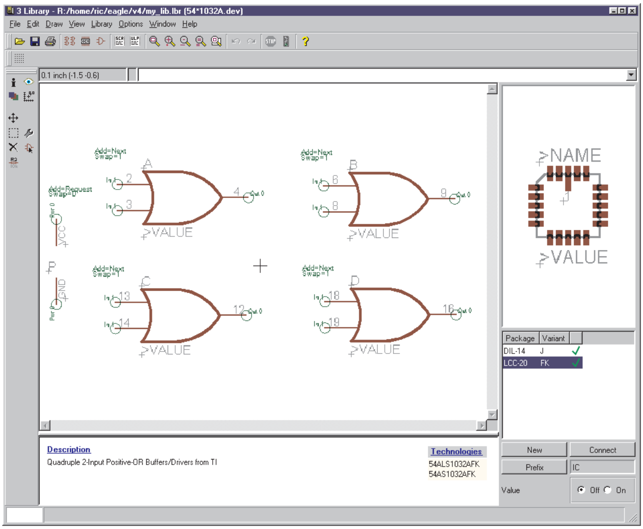
You can use Rich Text format. This is described in the help system under the keyword *Rich Text*.

It can look like this:

```
<b>541032A</b>
<p>
Quadruple 2-Input Positive-OR Buffers/Drivers
from TI.
```

Save

This completes definition of the Device set. If you have not already saved the library, please do it at this stage!


3 Library - R:/home/ric/eagle/v4/my_lib.lbr (54*1032A.dev)
File Edit Draw View Library Options Window Help
0.1 inch (-1.5 -0.6)
Add=Next
Swap=1
A
B
C
D
Add=Request
Swap=0
VCC
GND
P
>VALUE
>NAME
New
Connect

Description

Quadruple 2-Input Positive-OR Buffers/Drivers from TI

Technologies

54ALS1032AFK
54AS1032AFK

Package	Variant	
DIL-14	J	✓
LCC-20	FK	✓

Prefix IC

Value ● Off ○ On

Device-Editor: 54*1032A.dev

8.3 Supply Voltages

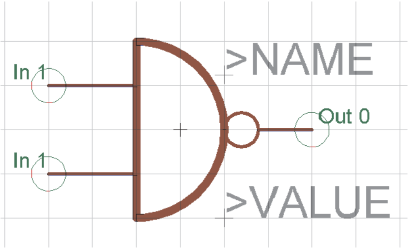
Component Power Supply Pins

The components' supply pins are to be given the direction *Pwr* in the symbol definition. The pin name determines the name of the supply signal. Pins whose direction is *Pwr* and which have the same name are automatically wired together (even when no net line is shown explicitly). Whether the pins are visible in the schematic diagram or are fetched by means of a hidden symbol is also not relevant.

Invisible Supply Pins

We do not want as a rule to draw the supply connections for logic components or operational amplifiers in the schematic. In such a case a specific symbol containing the supply connections is defined. This can be demonstrated with the example of a 7400 TTL component:

You first define a NAND gate with the name *7400*, and the following properties in the Symbol Editor:

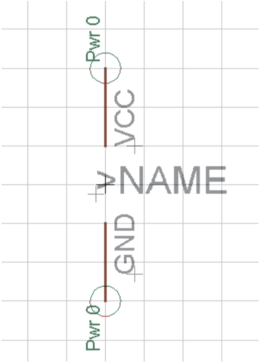


NAND symbol 7400 (European representation)

The two input pins are called *I0* and *I1* and are defined as having direction *In*, swaplevel *1,* visible *Pin* and function *None*.

The output pin is called *O* and is defined with direction *Out*, swaplevel *0*, visible *Pin*, and function *Dot*.

Now define the supply gate with the name *PWRN*, and the following properties:



Power gate

The two pins are called *GND* and *VCC*. They are defined with direction *Pwr*, swaplevel *0*, function *None*, and visible *Pad*.

Now create the *7400* device in the Device Editor:

Specify the package with PACKAGE (which must already be present in the library) and use PREFIX to specify the name prefix as *IC*.

Use the ADD command to place the *7400* symbol four times, with addlevel being set to *Next* and swaplevel to *1*. Then label the gates as A, B, C and D with the NAME command.

The addlevel of *Next* means that as these gates are placed into the schematic, they will be used in that sequence, i.e., the sequence in which they were fetched into the device.

Then place the *PWRN* symbol once, using addlevel *Request* and swaplevel *0*. Name this gate *P*.

Addlevel *Request* specifies two things:

- The supply gate will only be fetched into the schematic if requested, i.e. with the INVOKE command. The ADD command will only be able to place NAND gates.
- The supply gate will not be included when names are allocated to the schematic. Whereas an IC with two next-gates appears in the schematic as something like IC1A and IC1B, an IC with one *Next*-gate and one *Request*-gate will only be identified as IC1.

So use the CONNECT command to define the housing pads to which the supply pins are connected.

8.4 Supply Symbols

Supply symbols, such as might be used in the schematic for ground or VCC, are defined as devices without a package. They are needed for the automatic wiring of supply nets (see page 94).

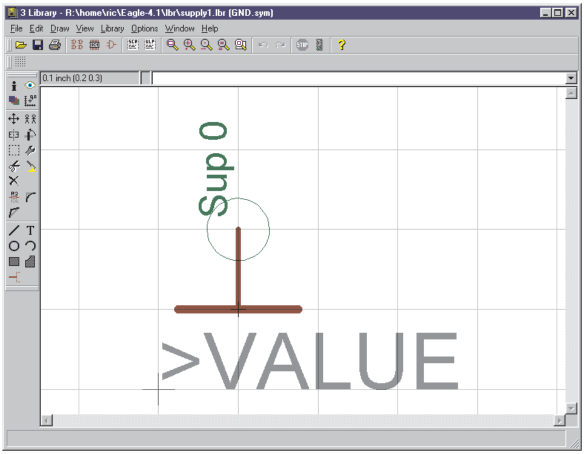
The following diagram shows a GND symbol as it is defined in one of the supplied EAGLE libraries.

Note that when defining your own supply symbols, the pin and the device name need to agree.

The pin is defined with direction *sup* and has the name *GND*. This specifies that the device containing this symbol is responsible for the automatic wiring of the GND signal. The text variable for the value (>VALUE) is chosen for the labeling. The device also receives the name *GND*. Thus the label GND appears in the schematic, since by default EAGLE uses the device name for the value.

It is very important that the labeling reproduces the pin names, since otherwise the user will not know which signal is automatically connected.

The pin parameter *Visible* is therefore set to *off*, since otherwise the placing, orientation and size of the pin name would no longer be freely selectable. Directly labeling with the text *GND* would also have been possible here. With the chosen solution however, the symbol can be used in various devices (such as for DGND etc.).


3 Library - R:\home\ric\Eagle-4.1\lbr\supply1.lbr (GND.sym)
File Edit Draw View Library Options Window Help
0.1 inch (0.2 0.3)
Sup 0
>VALUE

Supply symbol for GND

As has been explained above, the device receives the name of the pin that is used in the symbol. The corresponding device is defined with addlevel *Next*. If you set value to *off* you can be sure that the labeling is not accidentally changed. On the other hand, you have more flexibility with value set to *on*. You can alter the label if, for instance, you have a second ground potential. You must, however, then create explicit nets for the second ground.

Quick guide to define a Supply Symbol:

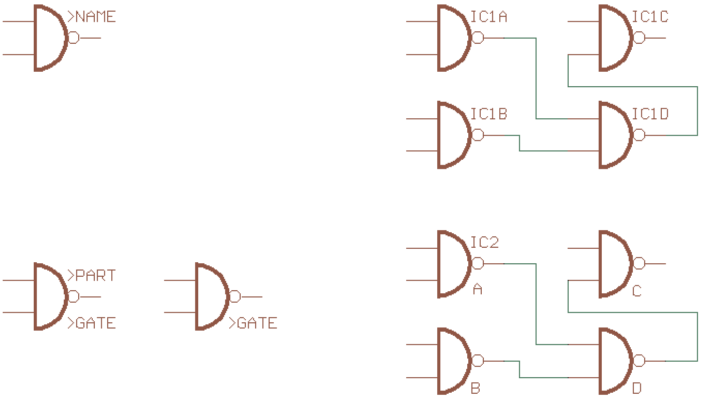
- Create a new symbol in the library
- Place the pin, with direction Supply
- Pin name corresponds to the signal name
- Value placeholder

- Create a new device
- Device name is signal name
- Package assignment not necessary

8.5 Labeling of Schematic Symbols

The two text variables >*NAME* and >*VALUE* are available for labeling packages and schematic symbols. Their use has already been illustrated. There are two further methods that can be used in the schematic: >PART and >*GATE.*

The following diagram illustrates their use, in contrast to >*NAME*. The symbol definition on the left, the appearance in the schematic diagram on the right.



Labeling of a schematic symbol

In the first case all the symbols are labeled with >NAME. In the second case, the symbol of the first gate is labeled with >PART and >GATE, the other three with >GATE only.

8.6 Pins with the Same Names

If you want to define components having several pins of the same name, then proceed as follows. Let us suppose that three pins are all to be called GND. During the symbol definition the pins are given the names GND@1, GND@2 and GND@3. Only the characters in front of the "@" are visible in the schematic, and the pins are treated there as if they were all called GND. However these pins are not necessarily internally connected.

8.7 More about the Addlevel Parameter

The addlevel of the gates that have been fetched determines the manner in which these gates are fetched into the schematic, and under what conditions it can be deleted from the schematic.

Summary

Next: For all gates that should be fetched in sequence (e.g. the NAND gates of a 7400). This is also a good option for devices with a single gate. The ADD command first takes unused next-gates from components which exist on the current sheet before "opening" a new component.

Must: For gates which must be present if some other gate from the component is present. Typical example: the coil of a relay. *Must*-gates cannot be deleted before all the other gates from that component have been deleted.

Can: For gates which are only used as required. In a relay the contacts may be defined with addlevel *Can*. In such a case the individual contacts can be specifically fetched with INVOKE, and can later be deleted with DELETE.

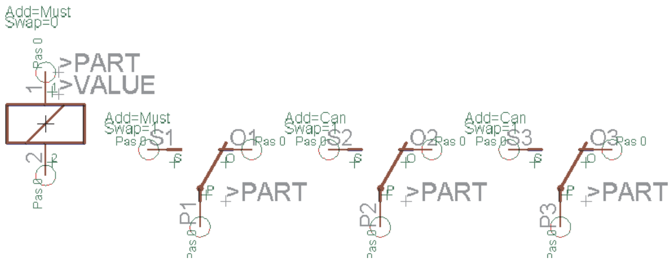
Always: For gates which as a general rule will be used in the schematic as soon as the component is used at all. Example: contacts from a multi-contact relay, of which a few are occasionally left unused. These contacts can be removed with DELETE, provided that they were defined with addlevel *Always*.

Request: Only for components' supply gates. The difference from *Can* is that they are not counted in name allocation. So a device with a *Next*-gate and a *Request*-gate will not be named IC1A and IC1B.

Relay: Coil and First Contact must be Placed

A relay with three contacts is to be designed, of which typically only the first contact will be used.

Define the coil and one contact as their own symbols. In the device, give the coil and the first contact the addlevel *Must*. All the other contacts are given the addlevel *Can*.



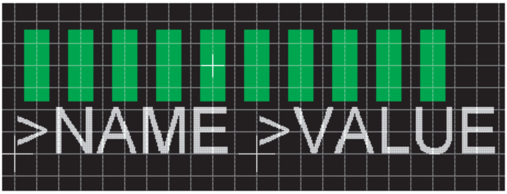
Relay with one coil and three contacts

If the relay is fetched into the schematic with the ADD command, the coil and the first contact are placed. If another contact is to be placed, this can be done with the INVOKE command. The coil cannot be deleted on its own. It disappears when all the contacts have been deleted (beginning with those defined with addlevel *Can*).

Connector: Some Connection Surfaces can be Omitted

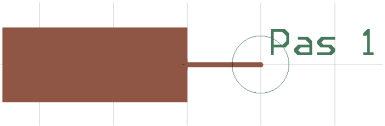
A pcb connector is to be designed in which normally all the contact areas are present. In some cases it may be desirable for some of the contact areas to be omitted.

Define a package with 10 SMDs as contact areas, giving the SMDs the names 1 to 10.



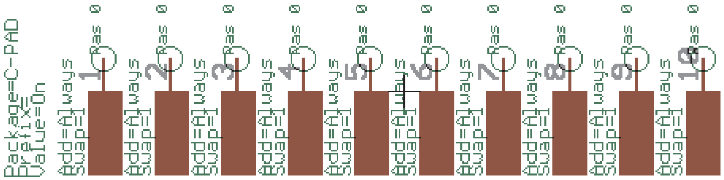
Package of a circuit board connector

Now define a symbol representing one contact area. Set visible to *Pad*, so that the names 1 to 10, defined in the package, appear in the schematic.



Connector symbol for the schematic diagram

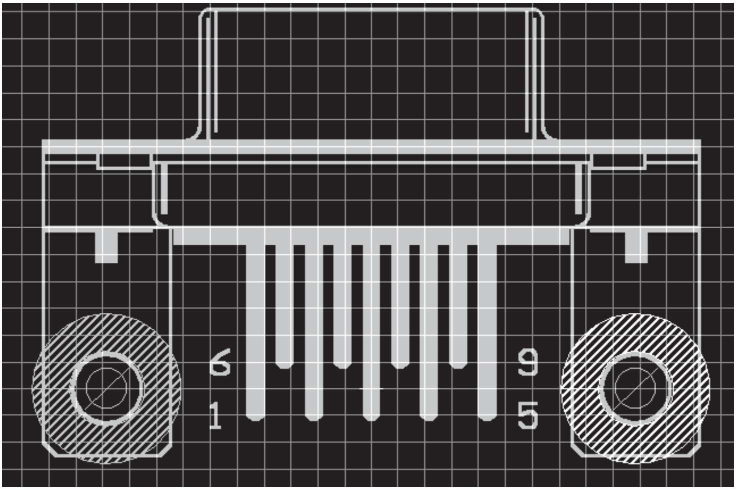
Then fetch the symbol ten times into a newly created device, setting the addlevel in each case to *Always*, and use the CONNECT command to create the connections between the SMDs and the pins. When you now fetch this device into a schematic, all the connections appear as soon as it is placed. Individual connections can be removed with DELETE.



After ADD, all the connections are visible in the schematic

Connector with Fixing Hole and Forbidden Area

A connector is to be defined having fixing holes. On the solder side (bottom), the Autorouter must avoid bringing tracks closer to the holes than a certain distance.



Fixing holes with restricted areas

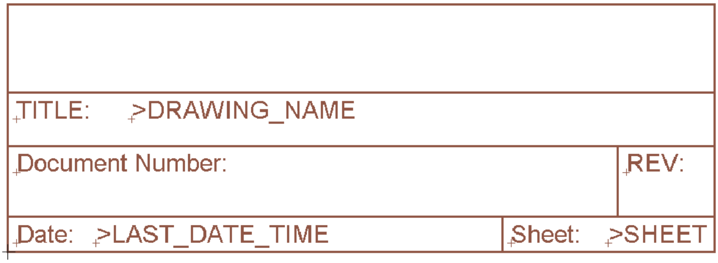
The drill holes are placed, with the desired diameter, on the package using the HOLE command. The drilling diameter can be retrospectively changed with CHANGE DRILL.

The forbidden area for the Autorouter is defined in layer 42, *bRestrict*, using the CIRCLE command. For reasons of representational clarity the circle is shown here with a non-zero width. Circles whose width is 0 are filled. In this case it has no effect on the Autorouter, since it may not route within the circle in either case. These forbidden areas are also taken into account by a polygon in layer 16, *Bottom*.

8.8 Drawing Frames

It may be true that drawing frames are not components, but they can be defined for schematics as devices with neither packages nor pins. Such devices in EAGLE's *frames* library contain a symbol consisting merely of a frame of the appropriate size, and a documentation field, which is also defined as a symbol.

In both cases the identification point is located at the bottom left, so that neither the frame nor the documentation field will be unintentionally selected while working within the drawing area.



Text variables in the documentation field

The text variables *>DRAWING_NAME*, *>LAST_DATE_TIME* and *>SHEET* are contained, as well as some fixed text. The drawing's file name, date and time of the last change appear at these points together with the sheet number in the schematic (e.g., *2/3* = sheet 2 of 3).

In addition, the variable *>PLOT_DATE_TIME* is available. It contains the date and time of the last printout.

All of these text variables can be placed on the schematic, and (with the exception of *>SHEET*) on the board.

The frame is defined in the device with addlevel *Next*, and the documentation field with addlevel *Must*. This means that the documentation field cannot be deleted as long as the frame is present.

There are frames defined as packages available for the Layout Editor which can be placed even if there is a consistent schematic/layout pair. These frames don't have any electrical significance because they are defined without pads or smds.

8.9 Components on the Solder Side

SMD components (and leaded ones too) can be placed on the top or bottom layers of a board. For this reason EAGLE makes a set of predefined layers available which are related to the top side (*Top*, *tPlace*, *tOrigins*, *tNames*, *tValues* etc.) and another set of layers related to the bottom side (*Bottom*, *bPlace* etc.).

SMD components are always defined in the layers associated with the top.

In the board, a component of this sort is moved to the opposite side with the MIRROR command. Therfore click the component with the mouse or enter the component's name in the command line. This causes objects in the *Top* layer to be reflected into the *Bottom* layer, while all the elements in the *t..* layers are reflected into the corresponding *b..* layers.

If one of the commands ADD, COPY, MOVE, or PASTE is active the component can be mirrored by clicking the middle mouse button.

8.10 Creating New Package Variants

Most components are manufacured in various package variants. Supposed you do not find the appropriate package for a certain device in one of the libraries, it is very easy to define a new package.

To describe this procedure clearly we want to come back again to our example device *541032A* from paragraph 8.2.
Please notify the explanations concerning this topic, in particular if the appropriate package already exists in the current library beginning with page 186.

The third package variant to be designed here only serves as an example for practice and does not meet the specifications of the manufacturer!

Package from Another Library

In the most favorable case you can use an already existing package from another library. The easiest way to define the new package variant is to use the PACKAGE command directly in the Device Editor.

After searching all libraries, either in the *Libraries* branch of the tree view in the Control Panel or with the help of the search function of the ADD command in the Layout Editor (only there, not in the Schematic, one can search for packages), it is possible to Drag&Drop the package from the Control Panel directly into the library. As an alternative you could use the PACKAGE command.

Open the library (here: *my_lib.lbr* from paragraph 8.2) that contains the device you want to define the new package variant for. For example, by the menu *File/Open/Library* of the Control Panel.

Click the *Edit-a-Device* icon and select the device *54*1032A* from the menu. The Device Editor opens.

Defining the Package Variant

The new variant should be named *Test*. The package must have a minimum of 14 pads because both gates together have 14 pins. As an example, we take the *SO14* package from the *smd-ipc.lbr* library.

If Control Panel and Library Editor window are arranged side by side, select the SO14 package and Drag&Drop it into the opened Library Editor window. After releasing the mouse button you will be asked for the new package variant name. Enter it and confirm it by clicking *OK*. The new variant is now shown in the package list.

It is also possible to define the package variant in the Device Editor directly with the PACKAGE command.

Type in the command line:

```
PACKAGE SO14@smd-ipc.lbr TEST
```

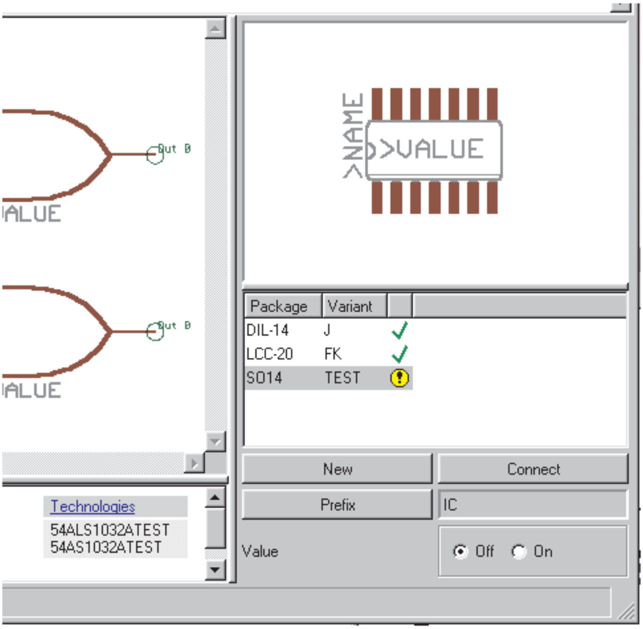
Or include the path (if necessary):

```
PAC SO14@d:\eagle-4.1\lbr\smd-ipc.lbr TEST
```

If the path contains spaces include the path name in single quotes, for example:

```
PAC 'SO14@\With Spaces\smd-ipc.lbr' TEST
```

Now on the lower right of the Device Editor window a new entry for the package *SO14* and the variant name *TEST* appears.
On the left a black exclamation mark on yellow ground is shown which indicates that there are no connections between pins and pads defined yet.



Package	Variant	
DIL-14	J	✓
LCC-20	FK	✓
SO14	TEST	!

Technologies
54ALS1032ATEST
54AS1032ATEST

Prefix: IC

Value: ● Off ○ On

Device Editor: List of Package Variants

The PACKAGE command copies the complete package definition into the current library and makes available the new variant with the given name for the device.

If you decide to erase a newly defined variant you can do this with the UNDO function (as far as possible) or by using the context menu of the package entry (right mouse click, Delete entry).

Connect Command

Click the *Connect* button now. The *Connect* window opens (see image one page 187). Connect pins with pads by clicking on the pin and pad entries belonging together as described in paragraph 8.2.
It is also possible to adapt the pin/pad connections from an already existing package variant. In our example the assignment does not differ from the DIL14 package. Therefore select the entry *DIL14* from the *Copy from:* combo box.
After clicking *OK* the CONNECT command is finished.

Defining Technologies

The device *54*1032A* is available in two technologies (ALS and AS). These still have to be set up for the new package variant.

Select the package variant *Test* from the list on the lower right of the Devcice Editor window. A click on *Technologies* in the description field opens a window. Click the *New* button and set up technology ALS with a following click on *OK*, and AS again with a following click on *OK*. Both entries are shown with a tag now. A further click on the *OK* button closes the window again.

Save

The definition of the package variant is finished. Now it is time to save the library.

Using a Modified Package from Another Library

If there is no appropriate but a similar package available in another library you should copy the package in the current library first, then modify it, and use it afterwards as new variant for the device.

Copying the Package

We want to use a package named *SOP14* from the *smd-ipc.lbr* library here. This package should get a new name, *MYSOP14*, in the library *my_lib.lbr*.

Using the COPY command

Type in the command line of the Library Editor window (it does not matter which editor mode is active) the following:

```
    COPY SOP14@smd-ipc.lbr MYSOP14
```

Or with the whole path:

```
    COPY SOP14@d:\eagle4.1\lbr\smd-ipc.lbr MYSOP14
```

If the path contains spaces use single quotes for it, for example:

```
    COPY 'SOP14@\P A T H\smd-ipc.lbr' MYSOP14
```

The Package Editor window opens and the package can be modified as needed.

In the Control Panel

First of all open a Library editor window with the library that should contain the new package (*File/Open/Library*). It is not necessary to select a certain editing mode. Now switch to the Control Panel (e. g. *Window* menu) and expand the *Libraries* branch of the tree view.

Choose the library which contains the requested package and select it. On the right half of the Control Panel a preview of the package is visible now.

If the Control Panel and the Library Editor window are arranged in a way that both windows are visible you can move the packge into the library window by keeping the left mouse button pressed. After releasing the mouse button (Drag&Drop) the Library Editor will be in the package editing mode. The package is shown there.

Alternatively you could use a right mouse click to open the context menu of the package entry in the tree view. Select *Copy to Library* now. The Library Editor needs not to be visible.

Now the package can be modified. The package name is adopted from the source library. To change the package name use the RENAME command.

Don't forget to save the library.

Defining the Variant

We want to define a further variant for our example device. Switch to the Device editing mode, for example, by the menu *Libraries/Device*. The *Edit* window opens. Select the entry *54*1032A*. Click *OK* to open the editor window.

Use the *New* button to define a new variant. Select the package *MYSOP14* in the selection dialog and enter, for example, *TEST2* as *variant name*. After clicking OK a new entry is shown in the package list.

To complete the definition execute the CONNECT command and define Technologies (as described in the previous paragraph) now.

8.11 Defining Packages in Any Rotation

Components can be defined in any rotation with a resolution of 0.1 degrees in the Package Editor. Usually the package is definied in normal position first and rotated afterwards as a whole. The definition of packages has been already explained in this chapter. Here we only want to elaborate on the rotation of packages.

Packages can be defined in any rotation! Schematic symbols can be rotated in 90-degrees steps only!

Rotating a Package as a Whole

To come back to the example of this chapter, please open the library *my_lib.lbr* and edit the package *LCC-20*.

Display all layers with `DISPLAY ALL` to make sure you have all objects rotated. Now use GROUP and draw a frame around everything.

Use the ROTATE command to rotate the group:
Now click with the left mouse into the *Angle* box of the paramter toolbar and type in the requested angle. Then use a right mouse click into the group to define the rotation point.
The package is shown now in the given angle.

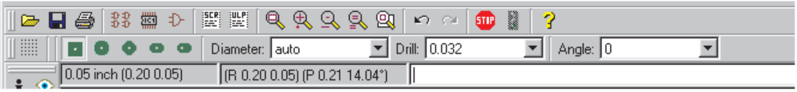
Alternatively you can work with the command line:

```
ROTATE R22.5 (> 0 0)
```

rotates, for example, the previously selected group 22.5° further around the point (0 0). The > sign (right angle bracket) within the parenthesis for coordinates causes the rotation of the whole group (as a right mouse click on the point (0 0) would do).

Packages with Radial Pad Arrangement

It is possible to work with polar coordinates to place pads or SMDs in a radial arrangement. Set a suitable reference point, for example, in the center of the package with the MARK command first. The command line shows now additional information about the cursor position.



Diameter: auto
Drill: 0.032
Angle: 0

0.05 inch (0.20 0.05)
(R 0.20 0.05) (P 0.21 14.04°)

Package Editor: Relative and Polar Coordinates Display

Values marked with an `R` are relative values referring to the previously set reference point. The leading `P` indicates polar values referring to the reference point.

Example:
Three pads are to be placed on the circumference of a circle with a radius of 50 mm. The center of the part is at position (0 0).

```
GRID MM;
MARK (0 0);
PAD '1' (P 50 0);
PAD '2' (P 50 120);
PAD '3' (P 50 240);
```

Depending on the used pad shape it may be useful to place the pads rotated (for example for *Long* pads or SMDs).

It is possible to enter the angle directly in the parameter toolbar or in the command line while the PAD or SMD command is active.

Example:

```
GRID MM ;
MARK (0 0);
PAD '2' LONG R120 (P 50 120) ;
```

8.12 Library and Part Management

Copying of Library Elements

Within a Library

If you want to use a **symbol** or a **package** which already exists in a related manner for a device definition you can copy it within the library with the commands GROUP, CUT, and PASTE. Afterwards it can be modified as requested.

The following sections explain every single step with the help of an example package taken from *linear.lbr*.

Open Library

Use the menu *File/Open/Library* in the Control Panel to open the library *linear.lbr* or select the entry *Open* from its context menu of the tree view's expanded *Libraries* branch.

Edit Existing Element

Open the *Edit* window with *Library/Package* and select the package *DIL08*. After clicking *OK* it is shown in the Package Editor window.

Use DISPLAY to show all layers.

Draw a frame around all objects to be copied with GROUP.

Now click the CUT icon followed by a left mouse click into the group. Alternatively you may enter a reference point in the command line for CUT, as for example:

```
CUT (0 0) ;
```

Define New Element

Click the *Edit-a-package* icon in the action toolbar. Enter the name *DIL08-TEST* in the *New* field of the *Edit* window and confirm with *OK*.

Click the PASTE icon followed by a click at the drawing's reference point. The package will be placed.
Or use again the command line:

```
PASTE (0 0) ;
```

Now it can be modified as requested.

The given coordinates for CUT and PASTE can be used to move a group by a certain value in the coordinates system. This may be valuable for elements that have been drawn in the wrong grid.

This procedure can be applied to symbols too!
Devices can't be copied within a library!

From One Library into Another

Devices

If there is a proper device or Device set that you want to use in your current library you can copy it in two different ways.

In the Control Panel:
Move (with Drag&Drop) the requested Device set from the Control Panel's tree view into the opened Library Editor window. The complete Device set with symbol(s) and package(s) will be copied and newly defined.
As an alternative you could use the entry *Copy to Library* in the context menu of the device entry.

With the COPY command:
Type, for example,

```
    COPY 75130@751xx.lbr
```

or with the whole path

```
    COPY 75130@d:\eagle-4.1\lbr\751xx.lbr
```

in the command line, the Device *75130* from library *751xx.lbr* is copied into the currently opened library.

If the path contains spaces use single quotes for it, for example:

```
    COPY '75130@d:\P A T H\751xx.lbr'
```

If the Device should be stored in the current library under a new name simply enter it, like here:

```
    COPY 75130@751xx.lbr 75130NEW
```

Symbols

As symbols are not shown in the tree view of the Control Panel it is not possible to copy them from there into the Library Editor.
Symbols are copied from one library into another the same way as one would do it within the same library. Therefore we use the commands GROUP, CUT, and PASTE (see previous paragraph).

The only difference to copying a symbol within a library is that you have to open the other library (*File/Open*) after CUT and click into the group. Then you define a new symbol that may have the same or another name as before.

In short:

- OPEN the *source* library and EDIT the Symbol
- DISPLAY all layers
- Select all elements with GROUP
- Use CUT and click into the group to set a reference point
- OPEN your *target* library with *File/Open*
- EDIT a new Symbol
- Use PASTE to place it
- Save library

Packages

The procedure to copy packages is nearly the same as to copy devices.

Either move (with Drag&Drop) the requested package from the Control Panel's tree view into the opened Library Editor window. The complete package will be copied and newly defined in the current library. As an alternative you could use the entry *Copy to Library* in the context menu of the package entry.

Or use the COPY command. Type, for example,

```
COPY DIL16@751xx.lbr
```

in the command line, the package DIL16 from library *751xx.lbr* is copied into the currently opened library. If the library is not in the current working directory you have to enter the whole path, as for example, in:

```
COPY DIL16@\eagle-4.1\mylbr\751xx.lbr
```

If the path contains spaces use single quotes for it:

```
COPY 'DIL16@D:\P A T H\mylbr\751xx.lbr'
```

If the package should be stored in the current library under a new name simply enter it directly in the command line:

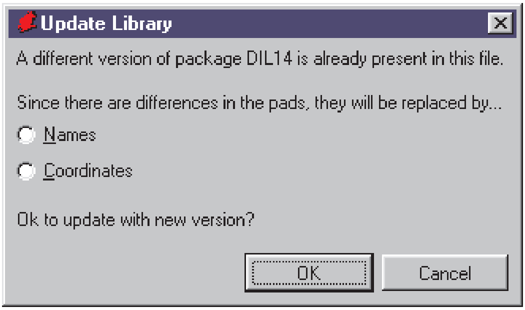
```
COPY DIL16@\eagle-4.1\mylbr\751xx.lbr DIL16NEW
```

The package is stored with the new name DIL16NEW now.

If you want to copy a package that already exists with the same name in the target library the package will be simply replaced.

If the package is already used in a device and the position or the name of one or more pads/SMDs changes, EAGLE prompts a message in which mode the pads/SMDs are to be replaced. This procedure can also be cancelled. The package remains unchanged then.

If the enumeration and position of the pads are unchanged but the order is, EAGLE will ask you for the appropriate update mode. Depending on your selection the pin/pad connections of the device may change (see CONNECT command).



Update Library

A different version of package DIL14 is already present in this file.

Since there are differences in the pads, they will be replaced by...

○ Names

○ Coordinates

Ok to update with new version?

Requesting the Update Mode

Composition of Your own Libraries

The previously mentioned methods to copy library elements make it very easy to compose your own libraries with selected contents.

Provided the Control Panel and the Library Editor window are arranged in a manner that both are visible on the screen at the same time, it is very easy to make user-defined libraries while browsing through the library contents in the Control Panel. Simply use Drag&Drop or the context menu *Copy to Library* of the current device or package.

Removing and Renaming Library Elements

Devices, symbols, and packages can be removed from a library with the REMOVE command. Defining a new library element can't be cancelled by UNDO.

Example:
You would like to remove the package named DIL16.
Open the menu *Library/Remove...*. A dialog window opens where you can enter the name of the element to be deleted.

This can be done also at the command line:

```
REMOVE DIL16;
```

Packages and symbols can be removed only if they are not used in one of the library's devices. In this case the message *Package is in use!* or *Symbol is in use!* appears. Remove the corresponding device first or delete the particular package or symbol in the device (set).

Would you like to change the name of an element in your library? Then use the RENAME command.
Switch to the Package editing mode so that the element that should be renamed is shown first and open the menu *Library/Rename*. A dialog window opens where you can enter the new name of the element.

This can also be done at the command line:

```
RENAME DIL16 DIL-16;
```

The package DIL16 gets the new name DIL-16.

The device, symbol, or package name may also be given with its extension (*.dev*, *.sym*, *.pac*), for example:

```
REMOVE DIL16.PAC
```

In this case it is not necessary to switch to the related editing mode before.

Update Packages in Libraries

As already mentioned in the section *Copying of Library Elements* it is possible to copy packages from one library into another one. An already existing package is replaced in that case.
Each library contains packages which are needed for device definitions. In many libraries identical types of packages can be found. To keep them uniform over all libraries it is possible to replace all packages of a library with those of another library with the help of the UPDATE command. An existing package with the proper name will be replaced by the current definition.

If you have, for example, special requirements for packages you could define them in a custom-built package or SMD library. The UPDATE command could transfer them to other libraries.

Therefore open the library to be updated und select *Library/Update...*. Now select the library which you want to take the packages from.

Having finished the update EAGLE reports in the status bar:
Update: finished - library modified!
If there was nothing to replace: *Update: finished - nothing to do.*

It is also possible to use the commad line for this procedure.

If you want to update your library with packages from, for example, *dil.lbr*, type:

```
UPDATE dil.lbr
```

To transfer packages from different libraries, type in one after another:

```
UPDATE dil.lbr smd.lbr special.lbr
```

To update a single package, type in the package name:

```
UPDATE SO14@smd
```

The extension *.lbr* is not necessary. You may also use the whole library path. See page 207 for further information.

Chapter 9
Preparing the Manufacturing Data

There are pcb firms who need only the EAGLE board file in order to manufacture films or prototypes. You will find links to such firms on our Internet pages.

If however your board maker is not set up to process EAGLE board files directly, you will have to supply him with a set of files. You generate these with the aid of the CAM Processor.

You obtain a parts list if, for instance, you load your schematic diagram into the editor and execute the User Language program *bom.ulp* with RUN. See also page 230.

Additional useful ULPs are available on CadSoft's Internet pages and on the current EAGLE-CD-ROM. They can be used, for example, for the generation of glue masks, for the calculation of milling contours, or for the output of data for automatic mounting and testing equipment.

9.1 Data for Board Manufacture

Gerber Format

Nowadays most pcb manufacturers use Gerber photoplotters with variable aperture wheels. In contrast to photoplotters with fixed aperture wheels, no agreement about the aperture table is required between you and the film maker.

Please inquire which format is desired. The simplest way is the RS-274X format, in which the aperture table is integrated into the output file. The generation of the data is just as easy as it is for PostScript or any other printer. Use the GERBER_RS274X driver for this. There is a predefined job file, *gerb274x.cam,* for easy generation with the CAM Processor. This job can be modified so suit your own requirements. Please read the following chapters describing how to modify or extend jobs with, for example, *gerber.cam*.

RS-274D, or subsets of it, is the commonest format, which can be generated with the CAM Processor. Therefore choose the devices GERBER and GERBERAUTO. In this case a file with the associated aperture table must be supplied, in addition to the files with the plot data. All the further explanations in this section are based on this format.

Drill Data

The generation of drill data is very similar to the generation of photoplot data. The aim is to create files which are sent to the board manufacturer. The appropriate device drivers are SM1000, SM3000, and EXCELLON resp. EXCELLON_RACK. A drill configuration has to be defined before an output can be generated. The manufacturer must know this configuration, otherwise he cannot drill the proper holes. It is therefore advisable to send the drill *rack* configuration file along with the drill *data* file(s) to the manufacturer.

Since EAGLE 4.11r2 the EXCELLON device writes the rack file information into the drill data file automatically. So you do not have to care about the rack file as it was in previous versions with the EXCELLON device.
The device EXCELLON_RACK works with a separate rack file as it is with SM1000 and SM3000 (and was with EXCELLON in earlier EAGLE versions).

EAGLE generates drill holes for the objects *pads*, *vias*, and *holes*. The appropriate data for pads and vias (plated-through holes) are generated if layer 44, *Drills*, is active during the CAM Processor output session. The data for holes (with no contact between different layers) are generated if layer 45, *Holes*, is active. Both layers must be active if all holes should be drilled together. If they should be drilled separately, either the Holes or the Drills layer must be active, and two separate files created.

A tolerance can be specified for drills. In such cases please make sure that your Design Rules are maintained.

The *Rotate* option given in the configuration is useful if the board appears in the Layout Editor in landscape format.

Data for Milling Machines

Data used by milling machines are normally drawn in layer 46, *Milling*. This is where, for example, the contours of board cut-outs are drawn. The output is made via the CAM Processor in Gerber or HPGL format. Agree the required format with your board manufacturer beforehand.

You can generate outline data for milling a prototype board with the help of the User Language programs.

A simple example ist the *outlines.ulp*. Start it with the RUN command.

Start the ULP with the RUN command.

Select the layer for which outline data will be generated, define the diameter of the milling tool, and select the output file format (script file format or HPGL, further formats will follow) in the ULP's dialog window. The script file containing the outline data can be imported into EAGLE with the SCRIPT command. In this case use the CAM Processor to generate manufacturing data.

The ULP executes the following steps:

- Drawing a polygon with the name _OUTLINES_ in the selected layer over the whole board area
- Setting the properties of the polygon:
  Rank = 6
  Width = diameter of milling tool
- Computing the polygon; the filling defines what has to be milled out
- Generating output data
- Deleting the polygon in the layout

Further information can be found in the help function, *Outline data*.

There is a further User Language program *mill-outlines.ulp* that generates outline data. This one offers, compared to *outlines.ulp* some more configuration parameters.

Data for Component Insertion Machines

EAGLE includes some ULPs which create data for various component insertion machines and in-circuit testers. In the *eagle/ulp* directory you will find various programs of this type (e.g. *mount_smd.ulp*, *fabmaster.ulp*, *unidat.ulp*).

To start an User Language Program use the RUN command in the Layout Editor.

9.2 Which Files does the Board Maker Need?

Files Generated with the CAM Processor

The following list contains the files which should be generated for the four-layer board demo.brd. In this example Layer 2 is a normal inner layer with various signals. Layer 3 is a supply layer for the GND signal, which the user has renamed to $*GND* in accordance with EAGLE's naming convention for supply layers. The filenames are suggestions. You are, of course, free to use unequivocal names of your own definition.

The options given in the list are recommendations, which in some cases can be changed.

File	**Active layers**	**Comments/recommended options**
demo.cmp	Top Via Pad	Component side. Options: pos. coord., optimize, fill pads.
demo.ly2	Via Pad	Multilayer inner layer. Options: pos. coord., optimize, fill pads.
demo.ly3	$GND	Multilayer supply layer. Is automatically output inverse. Options: pos. coord., optimize.
demo.sol	Bottom Vias Pads	Solder side. Options: pos. coord., optimize, fill pads.
demo.plc	tPlace Dimension tName	Component side silkscreen. Options: pos. coord., optimize.
demo.pls	bPlace Dimension bName	Solder side silkscreen. Only needed if there are components on the underside Options: pos. coord., optimize.
demo.stc	tStop	Component side solder mask. Options: pos. coord., optimize.
demo.sts	bStop	Solder side solder mask. Options: pos. coord., optimize.
demo.crc	tCream	Component side cream frame. Options: pos. coord., optimize.
demo.crs	bCream	Solder side cream frame. Only required if there are SMD components on the bottom side Options: pos. coord., optimize.
demo.drd	Drills Holes	Drilling data for NC drilling machines. Options: (rotate), pos. coord., optimize.
demo.drd.xxxx	Drills Holes	Drill data for multilayer boards for distinction of via lengths of blind and buried vias. Optiones: (rotate), pos. coord., optimize.

Additional Information for the Board Manufacturer

You must also supply the service company with the configuration files for apertures (*name.whl*) and drills (*name.drl*). You can also include the board file *name.brd*. This can help to avoid problems with what can become time-consuming questions.

In general, alignment marks (which are for example defined in layer 49, *Reference*) or an information field (in, for example, layer 48, *Documentation*) can be included.

A text file, such as *demo.txt*, should contain instructions for the board manufacturer.

9.3 Rules that Save Time and Money

- Each layer should without fail be uniquely identified (e.g. CS for Component Side).
- For cost reasons you should, if at all possible, avoid track that narrows to below 8 mil.
- Only angles should be drawn at the corners to delimit the board. Closed borders can lead to manufacturing difficulties.
- You should always leave at least 2 mm (about 80 mil) around the edge of the board free of copper. In the case of supply layers on multilayer boards, which are plotted inverse, you do this by drawing a wire around the edge of the board.
- As already mentioned, text in copper layers ought to be written in vector font. So you can really be sure that the text on your board looks the same as it does in the Layout Editor window.
- For the sake of completeness we want to point out here again that all questions concerning layer setup, layer thickness, and drill diameter for multilayer boards with blind, buried, or micro vias have to be pre-examined.

9.4 Generating the Data with Ready-Made CAM Jobs

The CAM Processor provides a job mechanism with the aid of which the creation of the output data for a board can be automated.

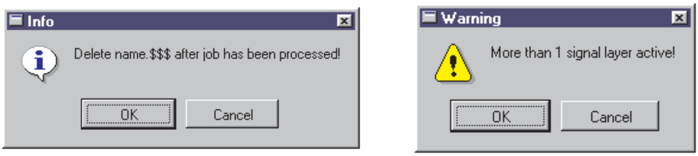
The Control Panel's tree view (*CAM Jobs* branch) lists all jobs and shows a brief description.

Gerber.cam Job for Two-Layer Boards

The *gerber.cam* job uses the GERBERAUTO and GERBER drivers. These generate files in the RS-274D format. It is set up for two-layer boards which are to receive a solder stop mask on the component and on the soldering side, and is to receive a silkscreen print.

Load the job by double-clicking on the *gerber.cam* entry in the tree view of the Control Panel.

In the first step an aperture table *name.whl* is automatically generated. Two messages appear, which you confirm with OK.



Delete name.$$$ after job has been processed!

More than 1 signal layer active!

Gerber job messages

The first message is generated by the entry in the *Prompt* field, and reminds you to delete the temporary files created when generating the aperture table when the job is done.

The second message advises you that more than one signal layer is active at the same time. Normally only one signal layer is active while output is generated. However, when generating the wheel, all the layers need to be active at the same time in order to form a common aperture table for all Gerber output.

Data for the following layers is subsequently output:

name.cmp	component side
name.sol	solder side
name.plc	Silkscreen
name.stc	Solder stop mask, component side
name.sts	Solder stop mask, solder side

If other layers are also to be generated, e.g. silkscreen for the bottom side, or a solder cream mask, the Gerber job can be extended as required.

Extending the job is discussed later in this section.

Job rs274x.cam

This job can be used as an alternative to *gerber.cam* if the circuit board manufacturer uses the *Extended Gerber* format. In contrast to *gerber.cam*, a separate aperture table is not created. The various Gerber files are simply output one after another. Each file contains aperture definition and Gerber data.

Drill Data

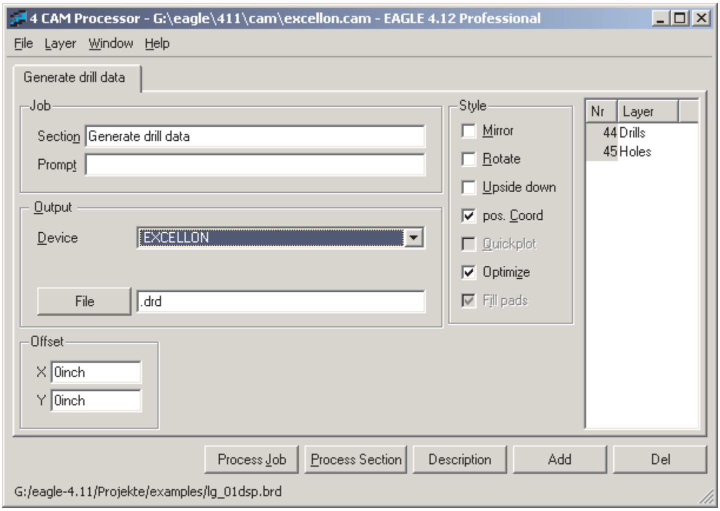
The CAM Processor offers the usual output devices for drill data, like Excellon, SM1000, and SM3000.

Job excellon.cam

The easiest way to generate drill data is to use the pre-defined CAM job *excellon.cam*. Used with the CAM Processor a file that contains drill data and the respective drill table will be generated automatically.

Proceed as follows:

- Start the CAM Processor (for example *File/CAM Processor*).
- Load the board into the CAM Processor (*File/Open/Board*) if it was not loaded automatically during the start sequence.
- Load the predefined job *excellon.cam*, e.g. with *File/Open/Job*.
- Execute the job (*Process Job*).
- Send the files *name.drl* and *name.drd* (for layouts with blind and buried vias *name.drd.xxxx*) and perhaps the info file *name.dri* as well to your boardhouse.



4 CAM Processor - G:\eagle\411\cam\excellon.cam - EAGLE 4.12 Professional

Job
Section: Generate drill data
Prompt:

Output
Device: EXCELLON
File: .drd

Offset
X: 0inch
Y: 0inch

Style
☐ Mirror
☐ Rotate
☐ Upside down
☑ pos. Coord
☐ Quickplot
☑ Optimize
☑ Fill pads

Nr	Layer
44	Drills
45	Holes

G:/eagle-4.11/Projekte/examples/lg_01dsp.brd

CAM Processor: Generate drill data with the excellon.cam job

The EXCELLON device combines drill table and drill coordinates in one common file. The default unit for the drill table is inch.
If the drill table would be preferred in millimeters the device definition can be changed in the file eagle.def. More details concerning this can be found in the section *Units for aperture and drill tables*.

Further output devices

If you prefer drill data in another data format like Sieb&Meyer 1000 or 3000 (SM1000, SM3000) or to work with the Excellon format that EAGLE used in its previous versions with a separate rack file (now device EXCELLON_RACK) a drill table is required first. This is created from the Layout Editor by means of a User Language program. The drill data is then output with the CAM Processor.

You proceed as follows:

- Load the board into the Layout Editor.
- Execute the *drillcfg.ulp* User Language program with the RUN command. It generates the *boardname.drl* drill configuration file.
- Start the CAM Processor (e.g. *File/CAM Processor*).
- Load the board into the CAM Processor (*File/Open/Board*) if it was not loaded automatically during the start sequence.
- Load the predefined job *excellon.cam*, e.g. with *File/Open/Job*.
- Check the parameters and change the output device to EXCELLON_RACK, SM1000, or SM3000. A tolerance of ± 2.5% makes sense to avoid problems due to internal unit conversions. Only the layers 44, *Drills,* and 45, *Holes*, may be selected. No other layers! You can display the selected layers by clicking *Layer/Show selected*.
- After changing the output device the CAM Processor demands for a *Rack*. Type in the file extension *.drl*. The CAM Processor now looks for the previously generated rack file in the same directory as the board file is.
- Save the job file via *File/Save job* with a new name.
- Execute the job (*Process Job*).
- Send the files *name.drl* and *name.drd* (perhaps *name.dri* as well) to your boardhouse.

Drill Data for Multilayer Boards With Blind and BuriedVias

The CAM Processor generates one drill data file for each via length for a layout that uses blind and buried vias.

The file extension *.drd* is expanded by the via length specification. If there are, for example, vias from layer 1 to 2, the output file extension will be *.drd.0102*.

The layer specification can be moved to another position with the help of the wildcard *%L*. Writing, for example, in the *File* box of the CAM Processor *.%L.drd* results in an output file with the file extension *.0102.drd*.

Pads and trough vias will be written into the output file with extension *.drd.0116*. If you have placed holes (HOLE command) in the layout and the *Holes* layer is active for output, the CAM Processor writes these values also in the same file with extension *.drd.0116*.

Your boardhouse finally gets the common rack file *name.drl* which is generated by *drillcfg.ulp*, provided you did not use the EXCELLON device which combines drill table and drill coordinates in a common file, and all files *name.drd.xxxx* containing the drill coordinates for each via length. Maybe also the common info file *name.dri*.

Drill configuration file (Rack)

This file is usually created by means of a ULP (*drillcfg.ulp*). It is of course also possible to use fixed drill table with the aid of a text editor. Then enter the table's path and filename in the CAM Processor's *rack* field. The rack file usually has *drl* as its extender.

```
T01   0.010
T02   0.016
T03   0.032
T04   0.040
T05   0.050
T06   0.070
```

Example of a drill configuration file:

All dimensions are given here in inches. It is also possible to enter the values with their unit, e.g. 0.010in or 0.8mm. Comments in drill configuration files are identified by a semicolon, which may stand at the start of a line or be preceded by a space.

Drill Info File

The file *name.dri* contains the missing drill diameters and other important information. It is written into the same directory as the output file.

9.5 Set Output Parameters

This section describes the setting of the parameters for the output of a drawing or a file, which will then be started with the button *Process job* or *Process section*. The parameters for a section, as described below, are set in the same way.

Load the schematic or board file from the CAM Processor's *File/Open* menu, and set the following parameters:

- Select the driver for the desired output device in the *Device* combo box.
- Enter the output data in the *File* field.
  If output is to go directly to a plotter, enter the name of the print queue that is connected to the corresponding computer interface in UNC notation, e.g.
  \\Servername\Plottername.
  If, under *File, Wheel*, or *Rack*, you only give the dot and the file extension, the entry will be expanded with the name of the schematic or layout file that is loaded.
  Example: *.cmp* is expanded to *boardname.cmp*.
  See also the section on the *Names of the Output Files*.
- Select the output layers by clicking the appropriate layer number.
  Click the menu *Layer/Deselect all* to first uncheck all the layers.
  Layer/show selected displays only the currently selected layers.
- Set device-specific parameters (aperture wheel etc.).
  Selecting the Gerber device driver, for example, invokes the *Wheel* field. Here you have to input the path to your aperture list. If you want to generate drill data (devices: EXCELLON_RACK, SM1000, SM3000), EAGLE asks for a drill table in the *Rack* field (see also page 212, *Drill data*).
- Set flag options.
  Mirror:
  Mirrors the output
  Rotate:
  Rotates the output by 90 degrees.
  Upside down:
  Rotates the output by 180 degrees. When combined with *Rotate*, the drawing is rotated by a total of 270 degrees.
  Pos. Coords.:
  Offsets the output so that negative coordinates are eliminated and the drawing is referenced to the origin of the output device. This is advisable for devices which generate error messages if negative coordinates are detected. The default is *on*.
  Quickplot:
  Draft output which shows only the outlines of objects. This option is not available for all devices.
  Optimize:
  Activates the optimization of the drawing sequence for plotters.

Fill Pads:
Pads will be filled. This option is on for generating manufacturing data.
Devices of the *generic* type (see eagle.def), for example PostScript, allow to deactivate this option. The drill holes for pads will be visible on the output.

- Select sheet (for schematics only):
  Use the *Sheet* combo box as far as the schematic consists of more than one page.

9.6 Names of the Output Files

The extensions for the output files given in the table on page 214 are only suggestions. If you are defining a job for Gerber data, which generates all the files with these extensions, the information file (**.gpi*) will be overwritten with each succeeding section. If you want to supply information files to your photoplotting service, proceed as follows:

For output, enter either *name.xx#* or *.xx#*.

Here, *name* stands for any name, and *xx* stands for any characters (that are allowed in file names).

The output files generated will look like this:

name.xx# results in *name.xxx*, the info file will be named *name.xxi*,
.xx# results in *boardname.xxx*, the info file will be named *boardname.xxi*.

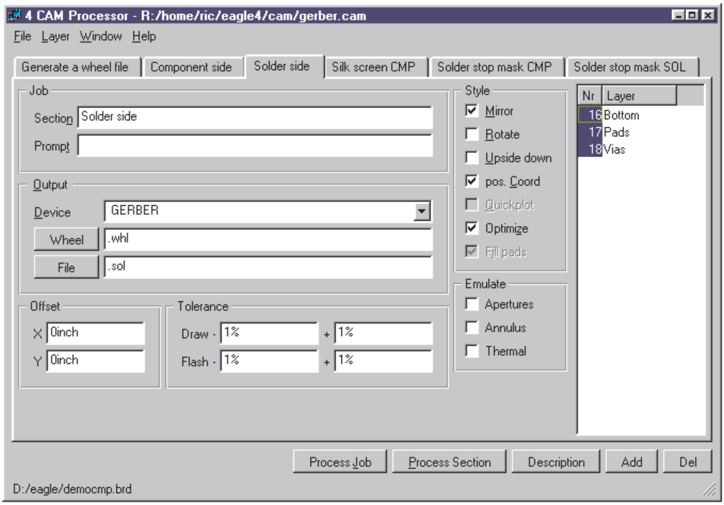
The files will be written into the same directory as the board file is.

Example:
The board *myboard.brd* is loaded. *.cp#* is entered into the *File* field.
The output file is called *myboard.cpx*, and the information file is called *myboard.cpi*.

The file extension can have any length. The # character must occupy the last place of the extension.

Please ensure when defining a job that the extensions of the output file for each section are unique and therefore distinguishable.

If you want to output a file on a particular drive, place the drive identifier or, if appropriate, the path in front of the filename ending. For instance, under Windows, *d:\.cmp* would place the file *boardname.cmp* in the root directory of drive D. This also applies to the Linux version, so that, for example, */dev/hdc2/.cmp*, would place the file on drive *hdc2*.



Job
Section: Solder side
Prompt:

Output
Device: GERBER
Wheel: .whl
File: .sol

Offset
X: 0inch
Y: 0inch

Tolerance
Draw - 1% + 1%
Flash - 1% + 1%

Style
☑ Mirror
□ Rotate
□ Upside down
☑ pos. Coord
□ Quickplot
☑ Optimize
☑ Fill pads

Emulate
□ Apertures
□ Annulus
□ Thermal

Nr	Layer
16	Bottom
17	Pads
18	Vias

D:/eagle/democmp.brd

CAM Processor: *Solder side* section of the *gerber.cam* job.

The diagram shows the ending *.whl* in the *Wheel* line for the aperture table. This means that the file *democmp.whl* is looked for. The output (*File*) receives the name *democmp.sol*. The flag options *Optimize* and *Pos. coords.* are active.

A tolerance of ± 1% is permitted for the aperture selection. This is necessary in order to compensate for small rounding errors (of around 0.00005 inch) that can arise during the conversion from mm to inch values as the aperture table is generated.

The status line below shows which layout is loaded.

9.7 Automating the Output with CAM Processor Jobs

Defining a Job

A job consists of one or more sections. A section is a group of settings, as described above in the *Set Output Parameters* chapter, which defines the output of *one* file. In this way you can use a job to generate all the files that are necessary for a project.

You define a job as follows:

- Start the CAM Processor. No job is loaded at first, unless there is a file called *eagle.cam* in the *cam* directory.
- Click on the *Add* button and enter a name for the *section* which is to be defined.
  If, for example, you assign the section name *Wheel: Generate Aper-*

ture File, only the title *Wheel* is visible as a tab name. The additional description can be read in the section line. The colon ends the title in the tab.

- Make a complete set of parameter settings. That means you have to select the *Device*, to activate the appropriate layers, to select the output, to set the scale factor if needed, and to check the flag options (*Mirror, Rotate, Upside Down...*).
- In the *prompt* field enter a message which is to appear on the screen before execution, should you so wish.
- Define further sections in the same way using different names. Very important: First use *Add* to create a new section, then set the parameters.
- Delete a section if needed by clicking the *Del* button.
- Save all the sections of a job under a name of your choice (with *File/Save job..*).

You can load a job through the tree view of the Control Panel (the *CAM jobs* entry) or via the *File/Open/Job* menu in the CAM Processor.
To generate the output for a particular board, the file must be loaded via the CAM Processor's file menu (*File/Open/Board*).

All the sections of the job will be executed if you click the *Process job* button.
A specific section will be executed if you click the *Process section* button.

The *Description* button allows a description of the CAM job that will be displayed in the Control Panel.

Extending gerber.cam Job for Multilayer Boards

The *gerber.cam* job can be used as the basis of the job for multilayer boards. It must simply be extended for the additional layers.

Example: You want to output the files for a board with SMD components on the top and bottom sides, a supply layer $*GND* in Layer2, and another inner layer with a polygon *VCC* in Layer 15 (which is renamed to VCC).

You need silkscreen prints for the upper and lower sides, solder stop masks, and a mask for the solder cream for both sides.

Before you start to change the CAM job you should save the job under a new name through the *File/Save job as..* menu.

The CAM job then contains the following sections:

1. Creating the aperture table with the GERBERAUTO driver.
All the layers which will be needed in later sections must be activated here.

2. Section for the component side (already contained in GERBER.CAM). You create the output files:

   name.cmp Layers: Top, Pads, Vias

3. Solder side (already contained in *gerber.cam*):

   name.sol Layers: Bottom, Pads, Vias

4. Silkscreen print, component side (already contained in *gerber.cam*):

   name.plc Layers: tPlace, Dimension, tNames

5. Silkscreen print, solder side (already contained in *gerber.cam*):

   name.pls Layers: bPlace, Dimension, bNames

6. Supply layer $GND (new in *gerber.cam*):

   name.ly2 Layers: $GND

7. VCC inner layer (new in *gerber.cam*):

   name.l15 Layers: VCC, Pads, Vias

8. Solder stop mask, component side (already contained in *gerber.cam*):

   name.stc Layers: tStop

9. Solder stop mask, solder side (already contained in *gerber.cam*):

   name.sts Layers: bStop

10. Solder cream mask, component side (new in *gerber.cam*):

   name.crc Layers: tCream

11. Solder cream mask, solder side (new in *gerber.cam*):

   name.crs Layers: bCream

Check once more whether all the necessary layers for the creation of the aperture table are active in the first section. The output file generated in the first section cannot be used. For this reason, the file *name.$$$* should be deleted.

Layers that refer to the bottom side of your layout will be generated mirrored in all predefined job files. If your board house recommends non-mirrored files, please deactivate this flag option for each section.

If you want to generate output files for supply layers with one signal (here $GND), you must only select the inner signal layer. No pads, no vias!

For other inner layers that may contain polygons you have to select additional pads and vias layers (as you would also do for the top or bottom side).

If the Layer Setup defines blind and buried vias, the combination of one signal layer and the Vias layer shows (according to the via length) only those vias that belong to this signal layer. If there is only the via layer active (no signal layer) the CAM Processor will output all vias of the layout.

9.8 Gerber Files for Photoplotters with Fixed Aperture Wheels

This section goes into more detail on the definition of the aperture table. Some circuit board manufacturers may perhaps still be using a Gerber plotter that works with a fixed aperture wheel. In such a case it is necessary to adapt the aperture table to the restricted facilities of the Gerber plotter. Files for Gerber photoplotters with fixed aperture wheels are generated with the GERBER driver. It is essential to confer with your photoplot service bureau in good time, so as to adjust EAGLE to the available apertures.

There are various types of apertures. They differ in size and shape. The most common are circle, octagon, square, thermal and annulus symbols. The drawing aperture used for tracks is normally the round aperture.

You must specify the aperture configuration before you can generate files for a fixed aperture wheel photoplotter.

To do this, enter the configuration file for apertures *name.whl* e.g. with the EAGLE text editor, and then load this file into the CAM Processor by clicking the *Wheel* button after selecting the GERBER device driver (see also the *Set Output Parameters* section, page 220).

The CAM Processor normally chooses the aperture from the configuration file appropriately for each object. For each line, a draw aperture must be present with the appropriate diameter. For each symbol (e.g. pad) an aperture with the correct shape and size is required. If this is not available, no output file will be produced.

Info File

The apertures which have not been found are then listed in the file *name.gpi*, where *name* here stands for the name chosen for the output file.

You can then change your board in such a way that the existing apertures can be used. After generating plot or drill files you should always check the related info files.

Aperture Emulation

If objects exist in a drawing which are not compatible with the available aperture sizes, it is possible to emulate the desired dimensions by selecting the *Emulate Apertures* option. The CAM Processor then selects smaller apertures to emulate dimensions which are not matched by aperture sizes. Emulation results in longer plot times and higher costs, so it should be avoided whenever possible.

Thermal or annulus apertures are only emulated with draw apertures if the *Emulate Thermal* or *Emulate Annulus* options, respectively, are activated.

The file *name.gpi* indicates the need for emulation and which apertures must be emulated.

Arcs with flat caps (CHANGE CAP FLAT) will always be emulated and drawn with short small wires (draw aperture). Arcs with round caps won't be emulated.

Aperture Tolerances

If you enter tolerances for draw and/or flash apertures, then if the aperture with the exact value is not available the CAM Processor will use apertures within the tolerances. Please bear in mind that your Design Rules might not be observed as a result of allowing tolerances!

Refer to file *name.gpi* for tolerances used.

Defining the Aperture Configuration

The CAM Processor distinguishes DRAW apertures from FLASH apertures. The first type is used to draw objects (e.g. tracks). The second type is used to generate symbols (e.g. pads) by a light flash. Only if draw apertures are defined can the plotter draw lines. Therefore, if the plotter doesn’t distinguish between draw and flash apertures, you must additionally define round or octagonal apertures as draw apertures.

The following apertures are available:

Name	**Dimension**
Draw	diameter
Round	diameter
Square	length
Octagon	diameter
Rectangle	length-X x width-Y
Oval	diameter-X x diameter-Y
Annulus	outside diameter x inside diam.
Thermal	outside diameter x inside diam.

Use of Aperture Shapes in the CAM Processor:

Draw	draws wires and emulates apertures
Round	draws round pads and vias
Square	draws square pads, SMDs and vias
Octagon	draws octagonal pads and vias with the same X- and Y-dimensions
Rectangle	draws rectangles and SMDs

```
Oval       draws octagonal pads with
           different X- and Y-dimensions
Annulus    draws isolation rings in a supply
           layer
Thermal    draws connections in a supply
           layer
```

Aperture configuration file example:

```
D001    annulus   0.004 x 0.000
D002    annulus   0.005 x 0.000
D017    annulus   0.063 x 0.000
D020    round     0.004
D033    round     0.059
D040    square    0.004
D052    square    0.059
D054    thermal   0.090 x 0.060
D057    thermal   0.120 x 0.080
D105    oval      0.090 x 0.030
D100    rectangle 0.060 x 0.075
D101    rectangle 0.075 x 0.060
D110    draw      0.004
D111    draw      0.005
```

All values default to inches, unless a unit is added (e.g. 0.010in or 0.8mm). Comments are marked with semicolons at the beginning of a line, or with a semicolon following a blank character.

9.9 Device Driver in File eagle.def

Creating Your Own Device Driver

Output device drivers are defined in the *eagle.def* text file. Here you will find all the information that is needed for the creation of your own device driver. The best way is to copy the block for an output device of the same general category, and then alter the parameters where necessary.

Please use a text editor that does not introduce any control codes into the file.

Here is an example for a Gerber(auto) device that generates data in mm units:

```
[GERBER_MM33]

Type     = PhotoPlotter
Long     = "Gerber photoplotter"
Init     = "G01*\nX000000Y000000D02*\n"
Reset    = "X000000Y000000D02*\nM02*\n"
ResX     = 25400
ResY     = 25400
Wheel    = ""
Move     = "X%06dY%06dD02*\n"     ; (x, y)
Draw     = "X%06dY%06dD01*\n"     ; (x, y)
Flash    = "X%06dY%06dD03*\n"     ; (x, y)
Units    = mm
Decimals = 4
Aperture = "%s*\n"                ; (Aperture code)
Info     = "Plotfile Info:\n"\
           "\n"\
           " Coordinate Format : 3.3\n"\
           " Coordinate Units  : 1/1000mm\n"\
           " Data Mode         : Absolute\n"\
           " Zero Suppression  : None\n"\
           " End Of Block      : *\n"\
           "\n"


[GERBERAUTO_MM33]

@GERBER_MM33
Long          = "With automatic wheel file generation"
Wheel         = "" ; avoids message!
AutoAperture  = "D%d" ; (Aperture number)
FirstAperture = 10
MaxApertureSize = 2.0
```

Units in the Aperture and Drill Table

When automatically generated with the GERBERAUTO driver, the aperture table contains values in inches. This is also the case for the drill table which is automatically written into the drill data file with the output device EXCELLON.
If your pcb manufacturer insists on mm units for aperture sizes and drill diameters, you can achieve this by altering the GERBER or GERBERAUTO respectively the EXCELLON driver.

In order to do this, use a text editor (one that does not introduce any control codes) to edit the *eagle.def* file. Look for the line

```
[GERBER]
```

or

```
[GERBERAUTO]
```

and at the end of that section add/edit the lines:

```
Units = or Decimals =
```

Example:

```
Units =    mm
Decimals = 4
```

To avoid problems arising from rounding errors during the conversion, a tolerance of ± 1% should be allowed for draw and flash apertures.

To change the drill table units there, look for the line:

```
[EXCELLON]
```

and change:

```
Units = Inch  to  Units = mm
```

9.10 Film Generation Using PostScript Files

Whereas, until a few years ago, Gerber files almost exclusively had to be generated for the manufacture of professional pcb films, there are today PostScript raster image recorders which offer a high-quality alternative which is simple to handle and economical.

With the *PS* driver, the CAM Processor generates files in PostScript format. These can be processed directly by appropriate service companies (most of which operate in the print industry).

For PostScript recorders the *Width* and *Height* parameters should be set to very high values (e.g. 100 x 100 inches), so that the drawing is not spread over several pages.

Films and drawings that relate to the bottom side are usually output in mirrored form (*Mirror* option in the CAM Processor or in the PRINT dialog).

The *EPS* driver generates Encapsulated PostScript files, and these can be further processed with DTP programs.

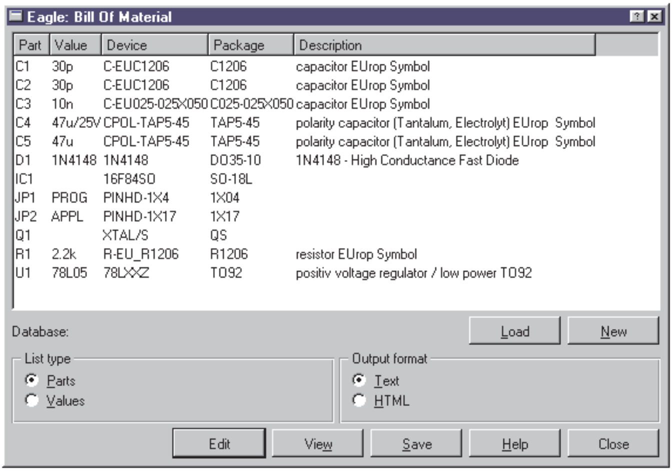
9.11 Documentation

Many documentation items can be generated with the aid of User Language programs. You can find a description of a ULP in the *User Language programs* branch of the tree view in the Control Panel, or at the start of a ULP file itself. In that case, examine the ULP with a text editor.

Note also the wide range of programs that are made available on our web server.

Parts List

The parts list can be created by a number of User Language programs. One very convenient program for the generation of the parts list is *bom.ulp*. Start it from the schematic diagram editor, using the RUN command. The *Bill Of Material* window with the parts summary opens first.



Part	Value	Device	Package	Description
C1	30p	C-EUC1206	C1206	capacitor EUrop Symbol
C2	30p	C-EUC1206	C1206	capacitor EUrop Symbol
C3	10n	C-EU025-025X050	C025-025X050	capacitor EUrop Symbol
C4	47u/25V	CPOL-TAP5-45	TAP5-45	polarity capacitor (Tantalum, Electrolyt) EUrop Symbol
C5	47u	CPOL-TAP5-45	TAP5-45	polarity capacitor (Tantalum, Electrolyt) EUrop Symbol
D1	1N4148	1N4148	DO35-10	1N4148 - High Conductance Fast Diode
IC1		16F84SO	SO-18L	
JP1	PROG	PINHD-1X4	1X04	
JP2	APPL	PINHD-1X17	1X17	
Q1		XTAL/S	QS	
R1	2.2k	R-EU_R1206	R1206	resistor EUrop Symbol
U1	78L05	78LXXZ	TO92	positiv voltage regulator / low power TO92

List type
● Parts
○ Values

Output format
● Text
○ HTML

bom.ulp: Dialog window

It is possible to import additional information from a database into the parts list (*Load*), or to create a new database with its own properties such as manufacturer, stores number, material number or price (*New*).

You can obtain further details about the current version of the ULP by clicking the help button.

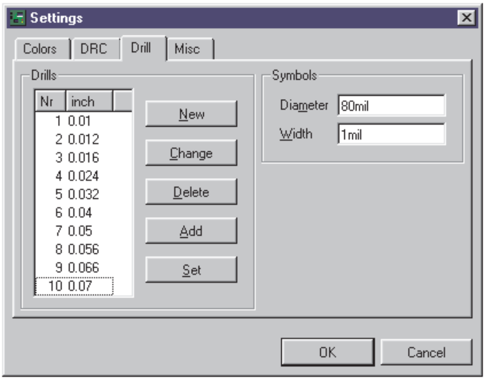
A simple parts list can also be created by means of the EXPORT command (*Partlist* option).

Drill Plan

A drill plan can be printed which enables you to check the drill holes and their diameters. It shows an individual symbol for each diameter. EAGLE uses 19 different symbols: 18 of them are assigned a certain diameter; one (Ø) appears if no symbol has been defined for the diameter of this hole.

The diameter symbols appear in layer 44, *Drills*, at the positions where pads or vias are placed, and in layer 45, *Holes*, at the positions where holes are placed. These layers must therefore be active when the drill plan is generated.

The relation between diameters and symbols is defined via the *Options/Set/Drill* dialog of the Layout Editor



Nr	inch
1	0.01
2	0.012
3	0.016
4	0.024
5	0.032
6	0.04
7	0.05
8	0.056
9	0.066
10	0.07

Diameter: 80mil

Width: 1mil

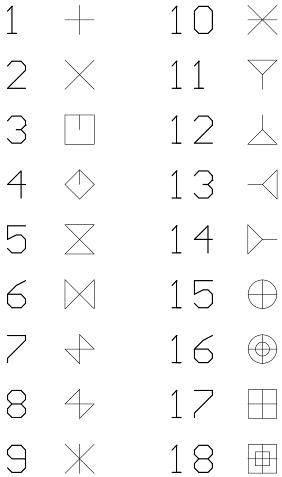
.

Configuration of the drill symbols

The buttons *New*, *Change*, *Delete* and *Add* can be used to create a new table, to modify certain entries, delete them or to add new ones. The *Set* button extracts all the hole diameters from the layout and automatically assigns them a drill symbol number.

The values of *Diameter* and *Width* determine the diameter and line thickness of the drill symbol on the screen and in a printout.

The image shows that drill symbol 1 is assigned to a drill diameter of 0.01 inch. In the following image you can see the related symbol drawn in layer 44, *Drills*, or 45, *Holes*, looks like. The symbol number 1 looks like a plus character (+).



Assignment of the drill symbols

The dill symbol assignment is stored in the user-specific file *eaglerc.usr* (*.eaglerc* for Linux).

Appendix

A. Layers and their Usage

In Layout and Package Editor

1	Top	Tracks, top side
2	Route2	Inner layer (signal or supply)
3	Route3	Inner layer (signal or supply)
4	Route4	Inner layer (signal or supply)
5	Route5	Inner layer (signal or supply)
6	Route6	Inner layer (signal or supply)
7	Route7	Inner layer (signal or supply)
8	Route8	Inner layer (signal or supply)
9	Route9	Inner layer (signal or supply)
10	Route10	Inner layer (signal or supply)
11	Route11	Inner layer (signal or supply)
12	Route12	Inner layer (signal or supply)
13	Route13	Inner layer (signal or supply)
14	Route14	Inner layer (signal or supply)
15	Route15	Inner layer (signal or supply)
16	Bottom	Tracks, bottom side
17	Pads	Pads (through-hole)
18	Vias	Vias (through all layers)
19	Unrouted	Airlines (rubber bands)
20	Dimension	Board outlines (circles for holes) *)
21	tPlace	Silk screen, top side
22	bPlace	Silk screen, bottom side
23	tOrigins	Origins, top side (generated autom.)
24	bOrigins	Origins, bottom side (generated autom.)
25	tNames	Service print, top side (component NAME)
26	bNames	Service print, bottom s. (component NAME)
27	tValues	Component VALUE, top side
28	bValues	Component VALUE, bottom side
29	tStop	Solder stop mask, top side (gen. autom.)
30	bStop	Solder stop mask, bottom side (gen. autom.)
31	tCream	Solder cream, top side
32	bCream	Solder cream, bottom side
33	tFinish	Finish, top side
34	bFinish	Finish, bottom side

35	tGlue	Glue mask, top side
36	bGlue	Glue mask, bottom side
37	tTest	Test and adjustment information, top side
38	bTest	Test and adjustment inf., bottom side
39	tKeepout	Restricted areas for components, top side
40	bKeepout	Restricted areas for components, bottom s.
41	tRestrict	Restricted areas for tracks, top side
42	bRestrict	Restricted areas for tracks, bottom side
43	vRestrict	Restricted areas for via-holes
44	Drills	Conducting through-holes
45	Holes	Non-conducting holes
46	Milling	Milling
47	Measures	Measures
48	Document	Documentation
49	Reference	Reference marks
51	tDocu	Detailed top screen print
52	bDocu	Detailed bottom screen print

In Schematic, Symbol, and Device Editor

91	Nets	Nets
92	Busses	Busses
93	Pins	Buses
93	Pins	Connection points for component symbols with additional information
94	Symbols	Shapes of component symbols
95	Names	Names of component symbols
96	Values	Values/component types

*) Holes generate circles with their diameter in this layer. They are used to place restrictions on the Autorouter.

Layers can be used with their names or their numbers. Names can be changed with the LAYER command or in the DISPLAY menu. The functions of the special layers remain.

If you want to create your own layers, please use layer numbers above 100. Use the DISPLAY menu to create new layers (*New* button) or type the LAYER command on the command line. If you want to create, for example, layer 200, *Remarks*, type in:

```
LAYER 200 Remarks
```

To set up color and fill style of this layer use the DISPLAY command.

B. EAGLE Files

EAGLE uses the following file types:

Name	**Type of file**
*.brd	Layout
*.sch	Schematic
*.lbr	Library
*.ulp	User Language Program
*.scr	Script file
*.txt	Text file (also other suffixes)
*.dru	Design Rules
*.ctl	Control parameter for the Autorouter
*.pro	Autorouter protocol file
*.job	Autorouter job
*.b$$	Backup file of brd after finishing the Autorouter
*.cam	CAM Processor job
*.erc	Error file from electrical rule check
*.b#x	Backup file of BRD (x = 1..9)
*.s#x	Backup file of SCH (x = 1..9)
*.l#x	Backup file of LBR (x = 1..9)
*.b##	Automatic backup file of BRD
*.s##	Automatic backup file of SCH
*.l##	Automatic backup file of LBR

EAGLE for Linux only creates and recognizes lower case characters in file endings!

C. EAGLE Options at a Glance

In order to output manufacturing data, for instance, with the CAM Processor, EAGLE can be started directly from a terminal window under Linux or from a console window under Windows.

Since Windows programs give up their connection to the console they have been started from, you can use the file *eaglecon.exe* (located in the *demo/bin* subdirectory on the CD-ROM) if you want to run the CAM Processor from a batch file.
This version of EAGLE is exactly the same as the *eagle.exe*, except that it doesn't disconnect from the console.
Type *eaglecon -?* for a list of CAM Processor options.

The following options are permitted:

```
-Dxxx  Draw tolerance (0.1 = 10%)
-Exxx  Drill tolerance (0.1 = 10%)
-Fxxx  Flash tolerance (0.1 = 10%)
-N+    Suppress message prompts
-O+    Optimize pen movement
-Pxxx  plotter Pen (layer=pen)
-Rxxx  drill Rack file
-Sxxx  Script file
-Wxxx  aperture Wheel file
-X-    eXecute CAM Processor
-a-    emulate Annulus
-c+    positive Coordinates
-dxxx  Device (-d? for list)
-e-    Emulate apertures
-f+    Fill pads
-hxxx  page Height (inch)
-m-    Mirror output
-oxxx  Output filename/channel
-pxxx  Pen diameter (mm)
-q-    Quick plot
-r-    Rotate output 90 degrees
-sxxx  Scale factor
-t-    emulate Thermal
-vxxx  pen Velocity
-u-    rotate output 180 degrees
-wxxx  page Width (inch)
-xxxx  offset X (inch)
-yxxx  offset Y (inch)
```

Where:

```
xxx Further data, e.g. file name as with -W  or a
    decimal number as with -s.
    Examples:
     -W /home/user/eagle/project/aperture.whl
     -s 1.25
-  Default for option is off
+  Default for option is on
   Example:
    -e   Aperture Emulation on
    -e+  ditto
    -e-  Aperture Emulation off

Flag options (e.g. -e) can be used without repeating the
'-' character:

    -eatm   Aperture emulation on, annulus and thermal
            emulation on, mirror output

    -ea-t+  Aperture emulation on, NO annulus
            emulation, thermal emulation on

Defining tolerance values:

If there is no sign, the value applies to either direc-
tion,
+  signifies a positive tolerance,
-  a negative tolerance.

-D0.10           adjusts the draw tolerance to ±10%
-D+0.1 -D-0.05   adjusts the draw tolerance
                 to +10% and -5%
```

Notes on the individual options:

-D **Draw Tolerance (0.1 = 10%):**
Default: 0

-E **Drill Tolerance (0.1 = 10%):**
Default: 0

-F **Flash Tolerance (0.1 = 10%):**
Default: 0

-N **Supress messages:**
This option supresses warnings or other information in the console window (DOS box, Linux console).
Thus CAM jobs run without interruption.
Default: off

-O **Route-Optimizing**:
With this option the route-optimizing for the plotter can be turned on and off.
Default: on

-P **Plotter Pen (layer=pen):**
If you use a color pen plotter, you can determine which layer is to be drawn in which color.
Example: -P1=0 -P15=1

-R **Drill Rack File:**
With this option you define the path to a file with the drill configuration table.

-S **Script File:**
When opening the editor window, EAGLE executes the *eagle.scr* file. This option allows a different name or directory to be selected for the script file.
The script file is not read by the CAM Processor.

-W **Aperture Wheel File:**
This option defines the path to the wheel file which should be used

-X **Calls command line version of the CAM Processor**

-a **Annulus Symbol Emulation:**
Default: off

-c **Positive Coordinates:**
If this option is set the CAM Processor creates data without negative coordinates. The drawing is moved to the zero-coordinates. This option can be turned off with the option -c-. Please be careful with this option, especially if you use mirrored and rotated drawings, because negative coordinates normally cause problems.
Default: on

-d **Device:**
This option determines the output driver. The following command displays a list of the available drivers:

```
eagle -d?
```

-e **Emulate Apertures:**
If this option is selected, apertures that do not exist are emulated with smaller apertures. If this option is turned off, no apertures are emulated. Thermal or annulus apertures are not emulated either
Default: off

-f **Fill Pads:**
This option can only work with generic devices like Postscript.
Default: on for all devices

-h **Page Height (inch):**
Printable region in the y-direction (in inches). The Y direction is the direction in which the paper is transported. See also the -w parameter.

-m **Mirror Output:**
Default: off.

-o **Output File Name**

-p **Pen Diameter [mm]:**
EAGLE uses the Pen-diameter measurement to calculate the number of lines required when areas are to be filled.
Default: 0

-q **Quick Plot:**
Generates a draft or fast output, which only prints the frames of the objects.
Default: off

-r **Rotate Output:**
Rotates the output by 90 degrees.
Default: off

-s **Scale Factor:**
Those devices which cannot change their scale-factor (in the menu of the CAM Processor), have a scale factor of 1.
Default: 1

-t **Emulate Thermals:**
Works only in combination with -e+.
Default: off

-u **Rotate Output by 180 degrees:**
In combination with -r+ one can rotate by 270 degrees.
Default: off

-v **Pen Velocity in cm/s:**
This option is for pen plotters supporting different speeds. To select a plotter's default speed, use a value of 0.
Default: 0

-w **Page Width (inch):**
Printable area in x direction. See also -h.

-x **Offset X (Inch):**
This option can be used to move the origin of the drawing.
Default: 0

-y **Offset Y (Inch):**
Default: 0

Example for starting eaglecon.exe:

```
eaglecon -X -dgerber -Waperture.whl -oboard.sol
-D-0.1 name.brd  pad via bottom
```

This generates a Gerber file board.sol which contains data of the layers Pads, Vias, and Bottom. Draw apertures may have a negative tolerance up to 10%.

All parameters have to be written in a common line!

D. Configuration of the Text Menu

With the help of a script file (e.g. *menu.scr*) you can configure your own text menu.

```
# Command Menu Setup
#
# This is an example that shows how to set up a complex
# command menu, including submenus and command aliases.
MENU ‘Grid {\
             Metric   {\
                        Fine   : Grid mm 0.1; |\
                        Coarse : Grid mm 1;\
                      } | \
             Imperial {\
                        Fine   : Grid inch 0.001; |\
                        Coarse : Grid inch 0.1;\
                      } | \
             On  : Grid On; | \
             Off : Grid Off;\
           }’ \
     ‘Display {\
                Top    : Display None Top Pads Vias Dim; |\
                Bottom : Display None Bot Pads Vias Dim; |\
                Placeplan {\
                         Top : Display None tPlace Dim; |\
                         Bottom : Display None bPlace Dim;\
                         }\
              }’\
     ‘–’\
     ‘Fit : Window Fit;’\
     Add Delete Move ‘;’ Edit Quit\
     ;
```

The backslash \ at the end of a line shows that a command continues in the next line. Here the MENU command runs from the first line after the comment to the last line.

The pipe sign | has to be used if a command within braces {} is followed by another command.

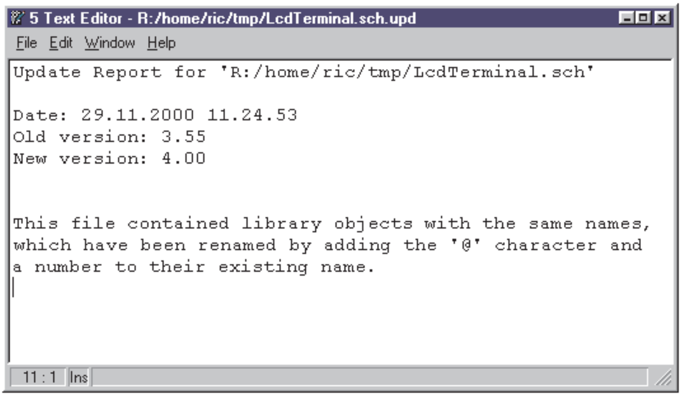
E. Text Variables

```
Text variable Meaning

>NAME         Component name (eventually + gate name) 1)
>VALUE        Component value/type                    1)
>PART         Component name                          2)
>GATE         Gate name                               2)
>SHEET        Sheet number of a circuit diagram       3)

>DRAWING_NAME   Drawing name
>LAST_DATE_TIME Time of the last modification
>PLOT_DATE_TIME Time of the plot creation

 1) Only for package and symbol
 2) Only for symbol
 3) Only for symbol or circuit diagram
```

F. Error Messages

When Loading a File

Library objects with the same names.



```
Update Report for 'R:/home/ric/tmp/LcdTerminal.sch'

Date: 29.11.2000 11.24.53
Old version: 3.55
New version: 4.00


This file contained library objects with the same names,
which have been renamed by adding the '@' character and
a number to their existing name.
```

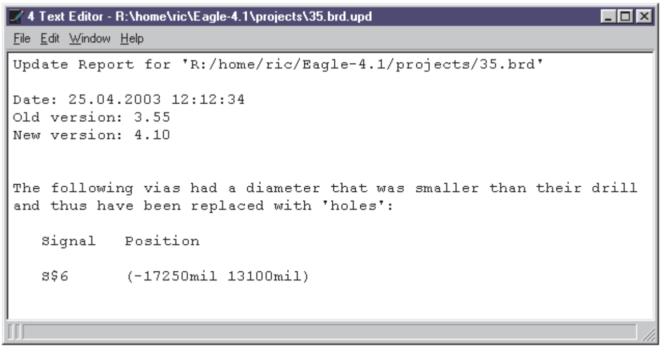
The text editor shows this message if you attempt to load an older file (BRD or SCH) that contains different versions of a library element. In this case it added @1, @2, @3... to the names of the devices so that they can be identified.

This message can also appear if you use CUT and PASTE commands.

Pad, Via Replaced with a Hole

In older versions of EAGLE it was possible to define pads in which the hole diameter was larger than the pad diameter. This is no longer permitted.

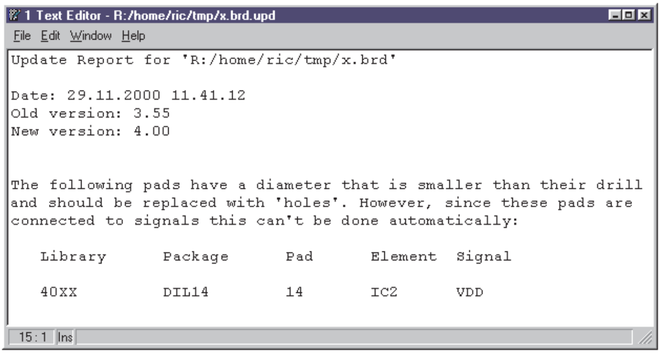
If you attempt to load a library file that was created with an earlier version and that contains such a pad, the following message appears:



```
Update Report for 'R:/home/ric/Eagle-4.1/projects/35.brd'

Date: 25.04.2003 12:12:34
Old version: 3.55
New version: 4.10


The following vias had a diameter that was smaller than their drill
and thus have been replaced with 'holes':

   Signal   Position

   S$6      (-17250mil 13100mil)
```

The pad or via is automatically converted into a hole, provided it is not connected by CONNECT to a pin in one of the library's devices.

If there is such a connection to a pin, the following message appears:

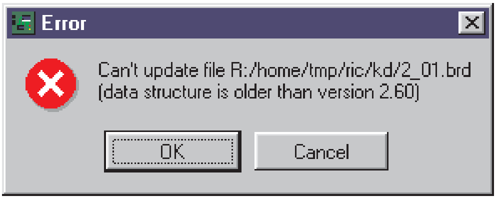


```
Update Report for 'R:/home/ric/tmp/x.brd'

Date: 29.11.2000 11.41.12
Old version: 3.55
New version: 4.00


The following pads have a diameter that is smaller than their drill
and should be replaced with 'holes'. However, since these pads are
connected to signals this can't be done automatically:

   Library      Package      Pad      Element   Signal

   40XX         DIL14        14       IC2       VDD
```

In that case the file must be manually edited in order to correct the pad.

Can't update file

If this message appears when loading an EAGLE file that was made with a version earlier than 2.60 it is necessary first to convert the file.



Error

Can't update file R:/home/tmp/ric/kd/2_01.brd
(data structure is older than version 2.60)

The program *update26.exe*, which is located in the *eagle/bin* directory, is used for this purpose.

Copy the file that is to be converted into the directory containing both *update26.exe* and the file *layers.new*. Then open a DOS window under Windows, and change into this directory. Type the command:

```
update26 dateiname.ext
```

The file is converted, after which it can be read by the new version of EAGLE. If the conversion is successful, the message in the DOS box is: *ok...*

If the message *Please define replacement for layer xxx in layers.new* should appear, it means that you have defined your own layers in layout/schematic/library.

Because of the new layer structure used since version 2.6, a new layer number (greater than 100) must be assigned.

This requires you to edit the file *layers.new* using a simple text editor, adding, for example, a new layer number as the last line of the file.

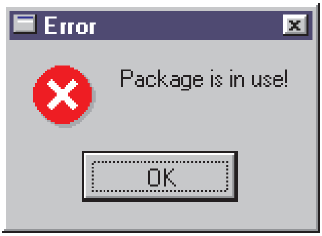
If, for instance, you have used layer 55, and want to give it number 105, enter:

```
55  105
```

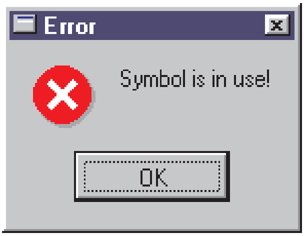
In a Library

Package/Symbol is in use

If a package or symbol is used in a device, no pads or pins may be deleted or added. Therfore the following messages appear after the PAD or PIN command have been selected:



Package is in use!



Symbol is in use!

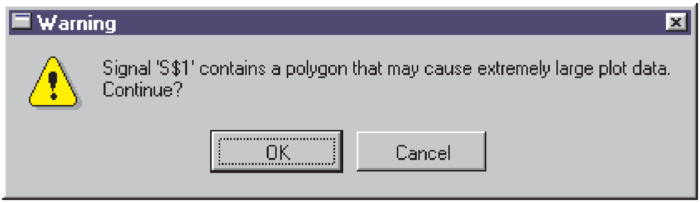
It is allowed to CHANGE pins or pads.

To change the number of pins/pads you have to delete the corresponding device(s) first (REMOVE command).

This message appears also if you try to remove the whole package/symbol from the library with the REMOVE command. You have to delete the whole device or the package variant or symbol in the device before.

In the CAM Processor

Signal contains a polygon that may cause extremely large plot data



Signal 'S$1' contains a polygon that may cause extremely large plot data. Continue?

This message always appears if you have drawn a signal polygon in the layout whose line thickness is less than the resolution of the selected output driver (*Device*).

In order to avoid unnecessary large plot files you should assign a higher value to the polygon’s line width (CHANGE width).

Index