Table of Contents

INTRODUCTION

ARES (Advanced Routing and Editing Software) forms the PCB layout module of the Proteus Design Suite and offers netlist based PCB design complete with a suite of high performance design automation tools.

ARES Version 7.xx is compatible with Windows 2000, XP, Vista and Windows 7. It includes a completely reworked modeless selection user interface, hardware accelerated display engine, new 3D Visualisation Tool and much more.

Layout Editor Features

Major features of ARES Professional include:

- 32 bit high-precision database giving a linear resolution of 10nm, an angular resolution of 0.1° and a maximum board size of +/- 10m. ARES supports 16 copper layers, two silk screens, four mechanical layers plus solder resist and paste mask layers.
- Hardware accelerated display using the power of your graphics to improve speed and provide true layer transparency.
- State of the art ergonomic user interface with modeless selection, selection and activity indicators and localised functionality via context menus.
- Netlist based integration with ISIS schematic capture, including the ability to specify routing information on the schematic.
- Automatic Back-Annotation of component renumbering, pin-swap and gate-swap changes.
- Live Physical Design Rule Checker and integrated Connectivity Rule Check reports.
- Powerful route editing features including topological route editing, auto track necking and curved trace support.
- World-class adaptive shape based autorouter capable of achieving completion on even the most complex of boards.
- Template and Technology Data scheme allows for flexible re-use of board outlines and design constraints.
- 2D Drawing with Symbol Library.
- Comprehensive package libraries for both through hole and surface mount parts including SM782 and IPC7351 standard SMT footprints.
- Unlimited Pad/Trace/Via Styles.
- Full metric and SMT support. This includes all dialogue form fields as well as the co-ordinate display and grid settings.
- Output to a wide range of printers and plotters including the Valor ODB++ format and traditional Gerber/Excellon. Also output in DXF, PDF, EPS, WMF and BMP graphics formats - to file or clipboard where appropriate.
- Built in Gerber Viewer - this enables you to check your Gerber output files before spending money on bureau fees or board manufacture.
- 3D Board Visualisation with solidworks compatible STL output and 3D DXF and 3DS output.

Auto-Placement

ARES includes an auto-placer module, an addition which in combination with the auto-router makes it possible to create PCBs almost entirely automatically. Alternatively, you can use it interactively, either by pre-placing critical components manually and auto-placing the rest, or

else by using it to place small sections of the design in turn with manual adjustments being made after each section is placed.

The auto-placer is highly configurable and can be set up to handle a wide variety of board types.

Auto-Routing

The most tedious, error prone and time consuming part of the traditional electronics development process is undoubtedly routing the PCB. As a result, it is in this area that the greatest benefits of Electronic Design Automation are to be found.

Our auto-router is a state of the art adaptive shape based router and you will find it capable of near 100% completion of designs that could take days by hand. It includes cost-based conflict reduction algorithms that allows the natural flow of the nets to emerge during routing and minimises both trace length and via count.

In addition, a direct interface is also available to the ELECTRA autorouter from Connect EDA and manual file export/import is supported for other routers such as SPECTRA™.

Power Planes

ARES supports Polygonal Gridless Power Planes which overcome most, if not all of the disadvantages associated with other methods of implementing copper fills. The essence of the approach is to generate polygonal boundaries around all the objects within the target area, and then to merge them together. The resultant multi-edged hole boundaries are subtracted from the original (user placed boundary) in order to establish whether or not there is complete or partial connectivity.

3D Board Visualisation

ARES includes a 3D Visualisation engine, allowing you to examine the board as it will appear in real life prior to prototyping. This not only assists with layout design but also provides information on height clearance. With simple, mouse driven navigation it is now the matter of a couple of mouse clicks to launch and examine your design in three dimensional space.

System Requirements

Performance in ARES is scaleable according to your computer specification. The following are the recommended minimums requirements :

- 1GHz or faster Intel Pentium processor (AMD processors fine but less optimised).
- Graphics gard supporting OpenGL Version 2.0 or higher and multi-sampling (MSAA).
- 256Mb RAM (recommended 512Mb).
- Windows 2000 or later.

In particular, note that if your graphics card does not satisfy the above requirements the software will run in Windows GDI mode. This means that display of the screen is handled by Windows and not your graphics hardware and therefore that some features of the software will not be available.

As a general rule, manufacturer graphics cards such as ATI and NVIDEA will satisfy these requirements whilst chipset graphics such as those supplied by Intel will not.

How to use this Documentation

Whereas the getting started guides (printed documentation) contains practical summaries and tutorials, the online help serves as a reference manual for the software. The tutorials are reproduced in this documentation for completeness as well as a vast amount of additional information. Using the table of contents or the index to quickly find references to topics of interest is recommended.

The Graphical User Interface and the general intelligence of the software itself will enable many

users to be productive almost from the outset. However, as with ISIS there is a great deal of functionality 'under the hood' and you cannot expect to master all aspects of the package immediately.

For those who need some initial tuition and guidance, we have followed the adage that the best way to learn is by doing - once you have installed the package by following the instructions in the next chapter, we suggest that you proceed to work through the extensive tutorial. This takes you right the way through the PCB design process from loading a netlist to performing the final CRC and DRC checks.

The remainder of the chapters provided background detail on all aspects of the system, and for quick reference, the final chapter deals with all the commands and any associated dialogue forms.

TUTORIAL

Introduction

The purpose of this tutorial is to familiarize you as quickly as possible with the main features of ARES to the point that you can use the package for real work. Users with modest computer literacy should find it possible to learn the package and produce their first board within a day or two.

- The tutorial proceeds by taking you through worked examples involving all the important aspects of the package including:
- Basic techniques for placement and routing.
- 3D Board Visualisation.
- Netlist based design including both manual and automatic routing .

More advanced editing techniques such as block editing and route editing.

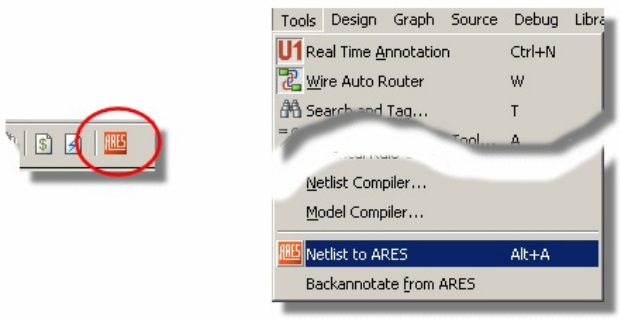
- Report generation.
- Hard copy generation.
- Library part creation.

We do urge you to work right the way through the tutorial exercises as many things are pointed out that if missed will result in much wasted time in the long run. Also, having worked through the tutorial and thus got a basic grasp of the concepts behind the package you will find it much easier to absorb the material presented in the reference chapters.

Note that throughout this tutorial (and the documentation as a whole) reference is made to keyboard shortcuts as a method of executing specific commands. The shortcuts specified are the default or system keyboard accelerators as provided when the software is shipped to you. Please be aware that if you have configured your own keyboard accelerators the shortcuts mentioned may not be valid. Information on configuring your own keyboard shortcuts can be found here.

Overview of the Layout Editor

We shall assume at this point that you have installed the package, and that the current directory is some convenient work area on your hard disk. This tutorial is a direct continuation of the ISIS Getting Started Guide so we will start by loading the completed schematic. This can be found in the ..\Samples\Tutorials\ directory of your Proteus installation and is called dsPIC33_REC.dsn. This circuit was designed and completed in the ISIS tutorial so all we need do is transfer the connectivity information (netlist) through to ARES for PCB Layout. We can do this via the Netlist to ARES command on the Tools Menu or via the ARES icon at the top right of the ISIS application.



Netlisting to ARES from the ISIS Schematic Editor.

After invoking this command a copy of ARES will open and present you with some options from which to create the new layout. Since we are starting from scratch in this tutorial we can select the 'blank' option which will then leave us with an empty layout.

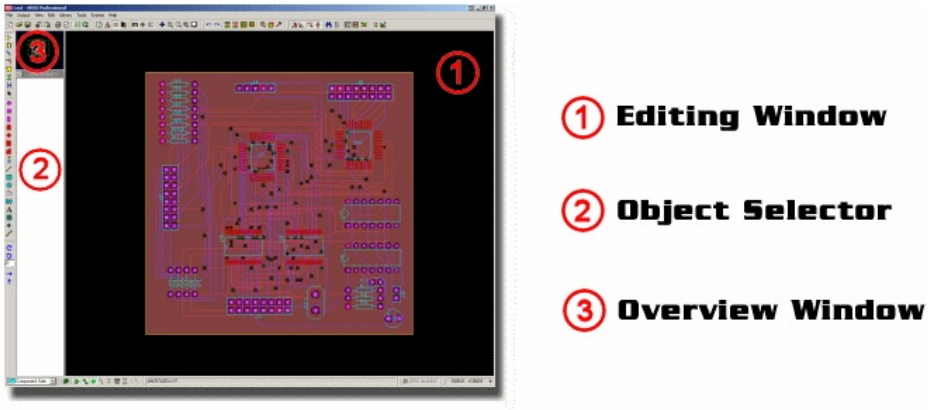
Templates and Technology Data re-use in general is beyond the scope of this tutorial. However, creating and saving templates and technology data is a valuable time-saver and is

well worth learning about. Please see the chapter on Templates & Technology for more information.

Much of the look and feel of the application is similar to ISIS and hopefully now familiar, although there are some important differences.

The Main Window

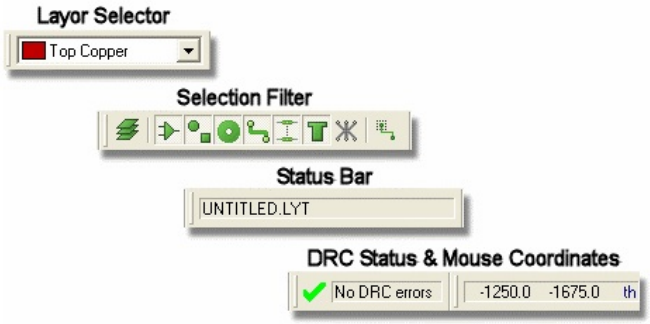
The largest area of the screen is called the Editing Window, and it acts as a window on the drawing - this is where you will place and track the board. The smaller area at the top left of the screen is called the Overview Window. In normal use the Overview Window displays, as its name suggests, an overview of the entire drawing - the blue box shows the edge of the current sheet and the green box the area of the sheet currently displayed in the Editing Window. However, when a new object is selected from the Object Selector the Overview Window is used to preview the selected object - this is discussed later.



Breakdown of the ARES Application Editor Space.

The Control Bar

The control bar at the bottom of the application is different from what we have seen in the ISIS application and essentially splits into four sections:

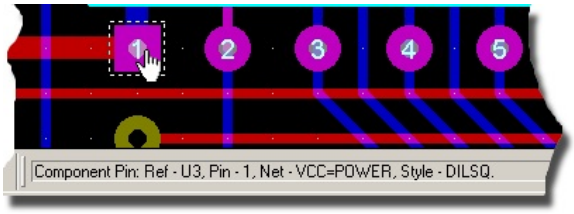


The bottom of the ARES screen

At the left hand side is the Selection Filter which allows you to configure both the layers and the objects that will be selected in the current operating mode. Typically, the default rules will suffice and this serves simply as a convenient override where you may wish finer granularity in selection at a given time. The Layer Selector combo box also determines the current layer or layer set and also applies to the placement of PCB objects.

ARES selection filter

In the middle is the Status Bar which provides textual ‘hints’ on the object currently under the mouse. This is particularly useful when you hover a mouse over a pad for example, as it will inform you which net the pad is on.



Status Bar in the ARES Layout Editor

Towards the right hand side is the live Design Rule Checker. This will report any physical design rule violations that occur while the board is being designed. A left click on this will launch a dialogue detailing the violations with the further option of zooming in to examine a particular error.

Design Rule indicator in the ARES layout editor

At the far right hand side is the co-ordinate display which reads out the position of the cursor when appropriate. These reflect not the exact position of the pointer but the location to which it has been snapped. Default snapping options are selectable from the View menu (or via keys CTRL-F1 and F2 -F4) and the snap values can be configured from the Grids command on the Technology menu

Co-ordinate display

The co-ordinates can be in imperial or metric units as set by the Metric (default key mapping 'M') command. You can also set a false origin using the Origin command (default key mapping 'O') in which case the co-ordinates change colour from black to magenta.

The dot grid on the Editing Window can be toggled on and off using the Grid command, or via it's keyboard shortcut (by default this is 'G'). The spacing of the dots normally reflects the current snap setting, except when zoomed out. In this case, the dot spacing is set to a suitable multiple of the snap spacing.

ARES can be set to display an X cursor at the position to which it has snapped the pointer through the X-Cursor command, default key mapping is 'X'.

We'll become familiar with all of these items and use them regularly as we work through placement and routing of the board.

Navigation

Navigation of the layout (middle mouse zoom, keyboard shortcuts, etc.), is identical to the ISIS schematic capture package. Refer to the ISIS Getting Started Guide or the ISIS reference manual for more information.

Visual Aids to Design

As in ISIS the ARES package will use visual effects to help you understand what is happening during board layout. There are two principle techniques:

ARES will 'twitch' an object when that object is under the mouse and the selection filter enables selection of that object type. This serves to identify when an object is 'hot'.

ARES provides dynamic cursors which change to identify what a left click will do at any given time (place an object, select an object, move an object, etc.). A list of cursor types is shown below:

Cursor	Description
	Standard Cursor.
	Placement Cursor - left click will place an object according to mode.
	Selection Cursor - left click will select the object under the mouse.

	Movement Cursor -left drag will move the object(s) selected.

Throughout this tutorial – and indeed whenever you are working in ARES – you should make use of these visual indicators to as an aid to understanding the software.

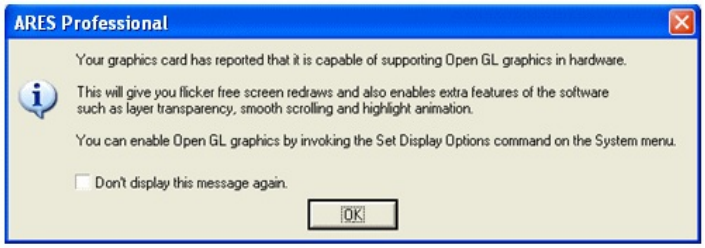
ARES and ISIS have a multiple 'undo' and 'redo' system via the shortcut keys CTRL+Z and CTRL+Y respectively. If you make mistakes throughout the tutorial remember that you can unwind as many times as you need to!

Display Options

ARES is capable of harnessing the power of your computers graphics card to speed up operation and provide true layer transparency on the layout. However, as not all machines have sufficient graphics cards the software is also capable of using Windows to perform display and graphical operations. The two modes of operation are called:

- Windows GDI Mode.
- OpenGL Hardware Accelerated Mode.
- Direct2D Hardware Accelerated Mode.

When you open ARES for the first time it will query your computers graphics card and inform you if your machine is capable of running in OpenGL mode or Direct2D mode.



Confirmation message that the graphics card supports Hardware Acceleration

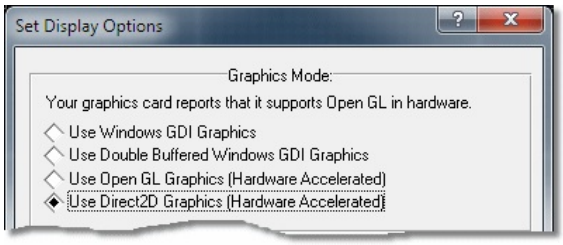
If your graphics card is not powerful enough to support hardware acceleration then ARES will simply default to Windows GDI mode. Should your computers graphics card be capable of both Direct2D and OpenGL we recommend first trying Direct2D mode as this tends to be more reliably implemented by graphics card vendors.

Configuration of the screen display takes place from the Set Display Options on the System menu. Many of these options are available only in OpenGL mode and will be disabled when Windows GDI mode is enabled.

Requirements for Graphics Acceleration.

Graphics Mode

The first section of the dialogue reports on whether your graphics card will support OpenGL or Direct2D hardware acceleration and, if so, allows you to switch from Windows GDI mode into one of these two modes.

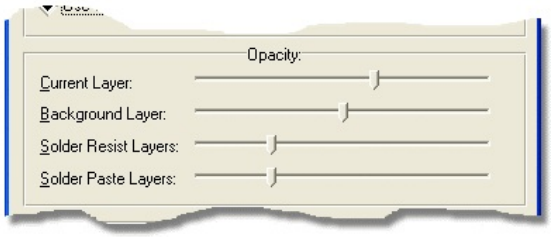


Direct2D mode selected in the Set Display Options

The following sections discuss the configuration options available from this dialogue form.

Opacity

When OpenGL graphics are selected the Opacity section of the dialogue allows you to configure the transparency of various layers on the board.



Sliders are used to set the opacity of the different layers

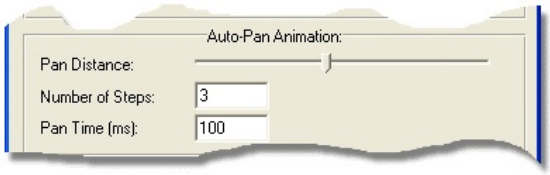
When in OpenGL mode, ARES places special significance on the current layer, giving it a higher opacity than other copper layers on the layout. This means that objects on the layer you are working are clearly visible and that objects on the other layers of the board are dimmed. You can control the relative opacity of both the current layer and the background layer(s) using the sliders on this section of the dialogue form. For example, if you wish to disable transparency altogether you could set the background layer slider to maximum.

A new feature in OpenGL mode is the ability to view the solder resist and paste mask around pads and vias on the layout. When enabled, you can adjust the opacity of these layers using the appropriate slider controls.

Layer Opacity Configuration with Displayed Layers Dialogue.

Auto-Pan Animation

In ARES, holding down the shift key and bumping the mouse against the edge of the Editing Window allows you to pan the screen. Similarly, dragging an object against the edge of the Editing Window will pan the screen in the direction of the drag. This navigation feature is called auto-pan and is controlled via the options in this section of the dialogue form.

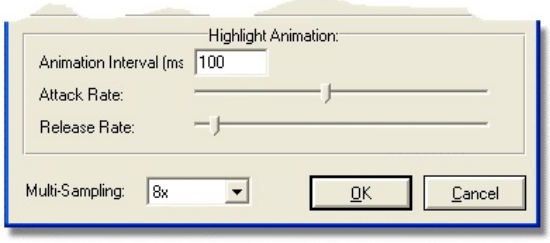


Setting the speed of panning in the ARES layout editor

You can adjust the distance moved on auto-pan (pan distance), the number of steps taken to cover this distance (number of steps) and the speed of the animation (pan time). All of these options are available regardless of whether you are working in OpenGL or Windows GDI mode.

Highlight Animation

When working in OpenGL mode objects under the mouse with smoothly increase their intensity to inform you that they are selectable. The highlight animation options on this dialogue allows you to control the smoothness and speed of this effect.



Setting the highlight options in the ARES layout editor

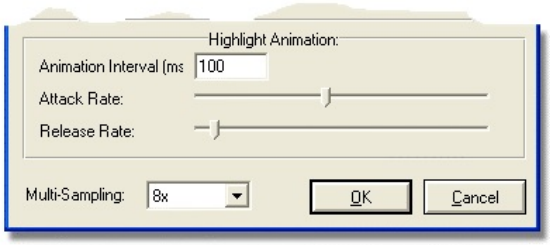
The animation interval controls the frame rate of the animation and therefore the smoothness of the fade-in/fade-out effect. Normally, the default value for this is suitable.

The attack rate specifies the speed at which an object will go from ‘fully off’ to ‘fully on’ whilst the release rate allows you to adjust the speed at which an object will return from ‘fully on’ to ‘fully off’.

These options are only available when OpenGL Graphics mode is selected.

Multi-Sampling (Anti-Aliasing)

Multi-Sampling is a method used by graphics cards to reduce anti-aliasing effects when graphics are displayed at different zoom levels. It is particularly applicable to text but impacts on all graphics on the layout.



Setting the Multi-sampling options in the ARES layout editor

When working in OpenGL mode you can specify the level of multi-sampling you want to use. The higher the level of multi-sampling the better the resulting display but the more GPU resources are consumed. If you select a level of multi-sampling which is not supported by your graphics hardware the software will reset the level to the closest one which your card can handle. For normal operation in ARES a multi-sampling level of 4x is quite sufficient.

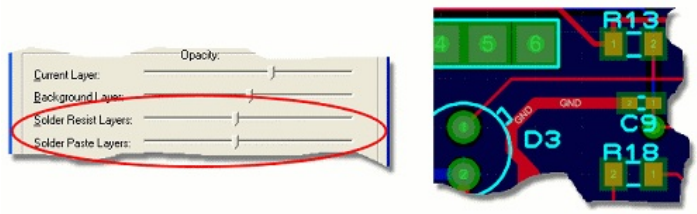
Displayed Layers Dialogue

The displayed layers dialogue allows you to control the visibility and colour of the various layers on the board. You can launch this from the View menu in ARES or by clicking the mouse on the status bar at the bottom of the application.

You can change the colour of a layer by clicking on the colour beside the layer and control the visibility of the layer via the checkbox beside the layer (where appropriate). All changes are updated live on the layout.

You can switch between the pre-supplied colour profiles (print and screen) or even create your own colour profile via the 'new' button at the top of the dialogue form. More information is available via the context sensitive help for the dialogue form (click on the question mark at the top right of a dialogue and then on a field of the dialogue form to launch).

Finally, if you are working in OpenGL mode, the Resist/Solder Paste Display options allow you turn on full display of these layers on the board, showing the resist and paste coverage around pads and vias. When enabled you can change the intensity of these layers by switching to the Thru-View settings tab and adjusting the appropriate slider controls.



The higher the sliders are the more the layers are shown

Component Placement

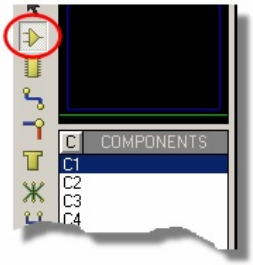
Since we have netlisted the project from our schematic design we have already supplied the ARES software with much of the information it needs to simplify the layout process. In particular, we have specified which footprints are associated with each schematic symbol and ARES can therefore pre-select these for us ready for placement. This brings us to an important distinction in the software; the difference between a component and a package.

Components and Packages

- A component is an instance of a footprint that has been netlisted through from the schematic.
- A package is a physical footprint that exists in the ARES Libraries.

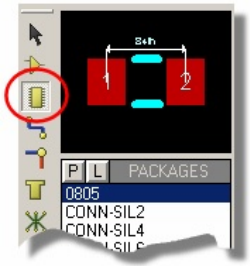
Selecting component mode will access footprints which have been specified as relating to parts in ISIS and carry connectivity information whereas selecting package mode will access ‘unbound’ physical instances of a footprint. When working with a layout which is driven from a schematic we therefore exclusively use component mode.

The Component Mode icon is second from the top directly underneath the Selection icon. Clicking on this will display a list of items in the Object Selector which correspond directly to the parts in ISIS that we used to create the schematic.



Component Mode icon

The Package Mode icon is directly underneath the Component mode icon and clicking on this will show us the physical footprints corresponding to the components in the layout.



Package Mode icon

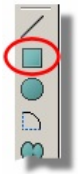
When placing, routing and laying out a PCB with a netlist (such as with this project) we will be working with components.

The Board Edge

Before we can place the components on the board we need to define what shape and size the board will be. For our project we need only a simple rectangular board edge but we do want to control the dimensions of the board (115mm wide and 40mm high).

The first thing to point out here is that ARES will operate in either imperial or metric units and you can switch between modes either by toggling the Metric command on the View menu or via the '**M**' keyboard shortcut. You may need to switch units for the placement of the board edge and also elsewhere in the documentation.

To start placing the board edge, select the 2D Rectangle icon from the left hand side of the application.



2D Square icon

Next, change the Layer Selector to the Board Edge Layer.

Layer Selector at the bottom left of the ARES layout

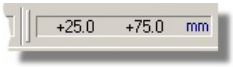
Move the mouse to the approximate starting point (e.g. top left) for the board edge. Now, hold the mouse still and press the 'O' key on the keyboard to set the origin of the co-ordinate system to the point under the mouse. This will be reflected in the co-ordinate display at the bottom right of the application. Left click to start placement and drag in the normal way. The co-ordinate display at the bottom will show you the dimensions of the board edge as you drag.

Origin co-ordinates

Press the 'M' key on the keyboard to toggle between metric and imperial units as required. Once you have the desired size of board edge click left again to commit placement. Don't worry about where in the Editing Window you have placed the board edge – we will move it to the centre of the world area shortly.

The Board edge is always yellow

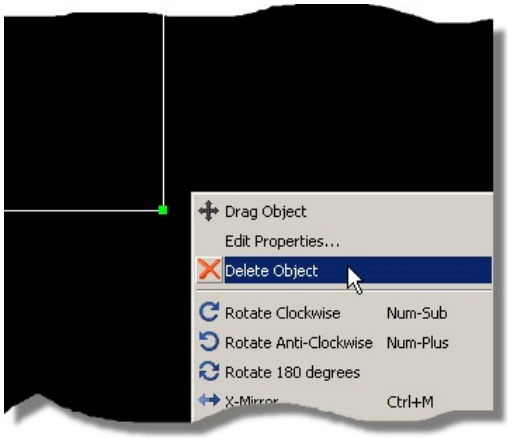
Finally, restore the co-ordinate system to it's global origin by pressing the 'O' button on the keyboard again. The colour of the co-ordinates will change from Magenta to black to indicate that the global origin is now in use.



Global origin

Placing a board edge should always be the first task as the software needs to know the boundaries of the board in order for example, to know the limits of where the autorouter can function or the size of a power plane.

If you need to delete the board edge or resize it you can do so via the context menu options and drag handles that will appear when you right click over the board edge graphic.



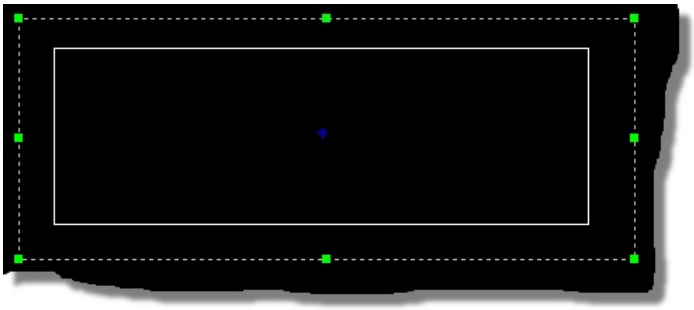
Right click on the 2D graphic and select 'Delete Object'

Complex board edges can be directly imported onto the board edge layer using the Import DXF function on the File Menu.

Work Area, Co-Ordinates and Snap

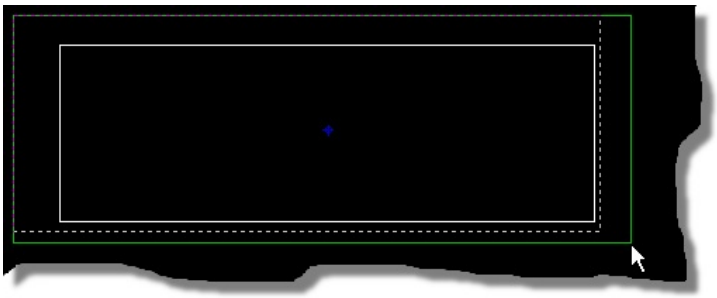
It is likely that your board edge covers a small portion of the Editing Window, which is less than ideal as all of the work will be taking place inside this area. We can of course zoom into the area (point the mouse and roll the middle mouse button or use the F6 key) but the default zoom level is designed to show the entire Work area. The Work area is the area inside the dark blue box at the edges of the Editing Window. Let's tidy this up a bit before we move on.

Start by switching into Selection Mode and then left click and drag a selection box around the board edge.



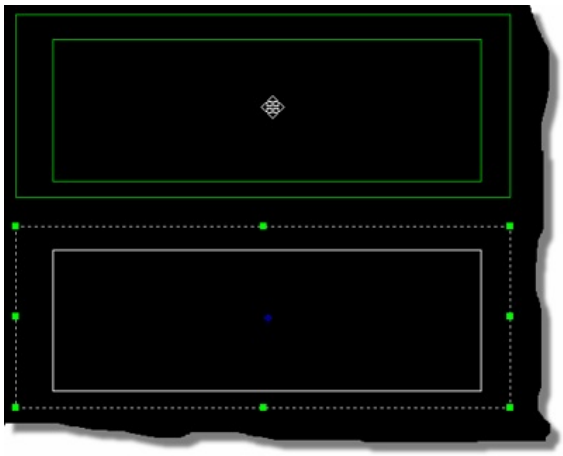
Drag a selection box around the object

If you get it wrong and don't completely cover the board edge, use the green handles to resize the selection box until it completely encompasses the board edge.



Re-sizing a 2D graphic

Now place the mouse inside the selection box, depress the left mouse button and drag into the centre of the work area, releasing the mouse button to commit the placement.



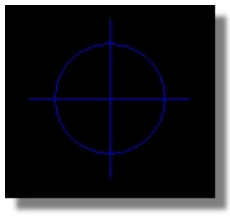
Dragging a 2D graphic

This is a very important technique worth mastering as it allows you to easily block select and then perform actions on (move, delete, etc.) groups of objects.

Having centred our board edge in the Work area we can shrink the work area to a suitable size. From the Technology Menu, select the Board Properties command and then set to something like 175mm by 100mm, such that the resulting work area forms a reasonably boundary to the board.

Select 'Board Properties' from the Technology menu and then select the board size

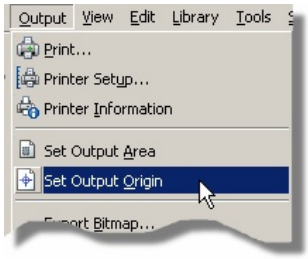
We saw briefly when placing the board edge that we can specify a temporary origin and use that to place items of a given size (or to place items a specific distance away from our origin point for example). We can however, also specify the location of the Output Origin. This is the default origin used whenever we do not explicitly set a temporary origin and is shown on the layout as a small blue marker.



Global origin

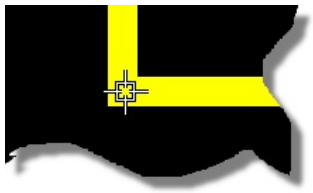
It is generally useful to move this to a corner of the board and particularly so if we have mechanical constraints that we need to consider during placement (e.g. mounting hole locations). For our design, we'll move the origin to the bottom left corner of the board edge.

1) From the Output menu choose the Set Output Origin command.



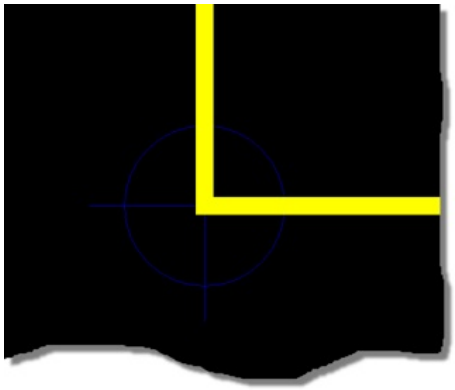
Setting the output origin from the Output menu.

2) Move the mouse towards the bottom left of the board edge. Roll the middle mouse button as required to zoom in for more accurate placement.



Moving the Output origin

3) Left click the mouse to commit placement.



The Global Origin has now moved to the specified place

The co-ordinate display will now provide values relative to the position at the bottom left of the board by default, and relative to any specified temporary origin when set

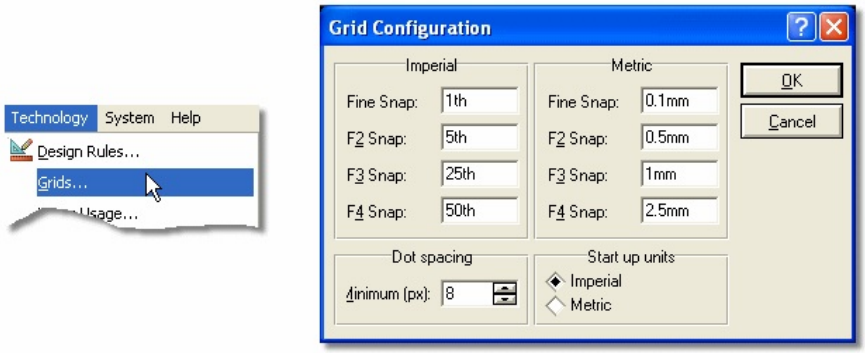
Remember that you can also toggle metric or imperial units via the command on the View menu or the 'M' keyboard shortcut.

A useful trick once we have specified our origin is to use the Goto-XY command on the

View Menu to move the mouse to the exact location we want to place position sensitive parts.

ARES has a default grid and will snap objects onto the grid, making it easy to form connections and control board layout. There are four hotkeyed snap settings for both metric and imperial units, changeable from the View Menu or by keyboard shortcuts. It follows that changing the snap setting to a lower unit will allow finer granularity, whereas raising the snap setting will make it easier to select objects at higher zoom levels. It is not advisable to change the snap settings regularly through a design but rather to choose the highest appropriate setting for the board you are working on.

You can change the four default snap settings via the Grids command on the Technology Menu

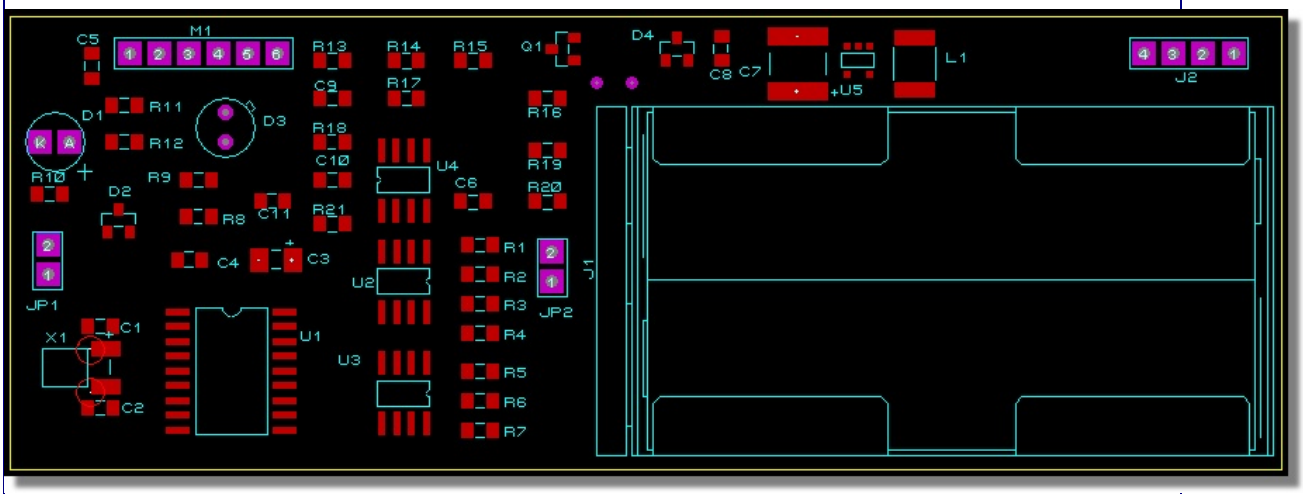


Snap settings changed from the 'Grids' option in the Technology menu

The grid display itself can be toggled between off, dots and lines via the 'G' key on your keyboard, whilst the colour of the grid can be configured from the Displayed Layers Dialogue (View Menu).

Placing Components and the Ratsnest

Now that we've handled all the basics we can finally start to get our components on the board. The following screenshot shows a fully placed board and we can use this to get our approximate positioning.



This is the PICDEM2 board as we've laid it out

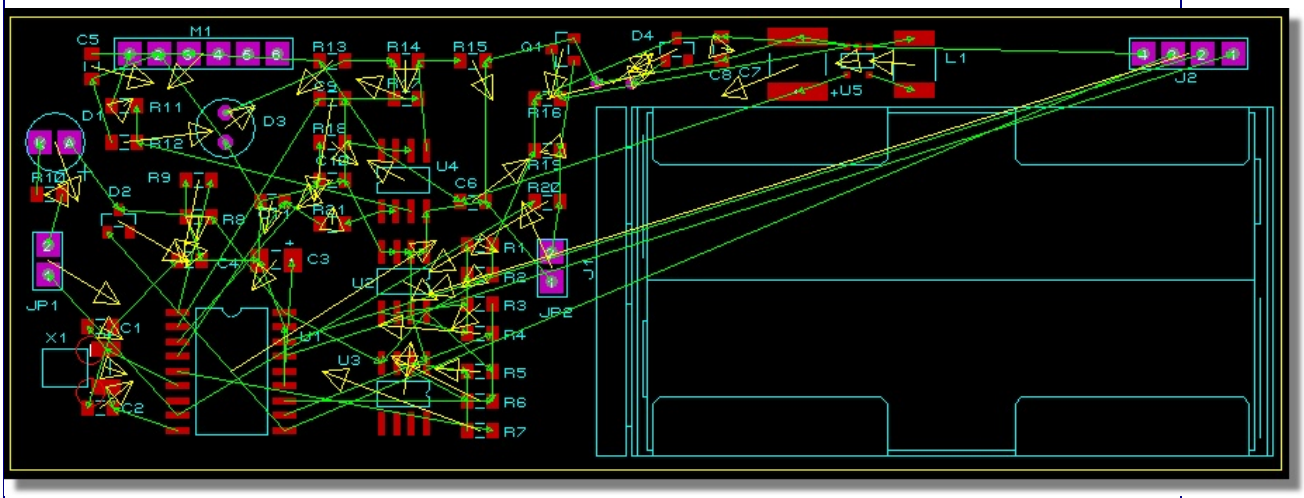
Placing a component in ARES is very similar to that in ISIS.

Start by selecting Component Mode from the left hand side icon set and then ensure that the layer selector is set to Component Side – we won't be placing any solder side components in this project.

Component mode Icon and Component layer

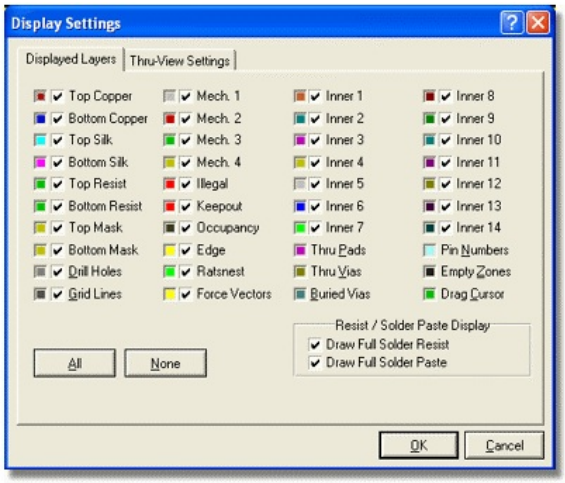
Let's get the AA battery holder down first on the right hand side of the board. Select J1 from the Object Selector, position the part using the previous screenshot as a guide and then left click again to commit placement. Note that once we have placed the part it is removed from the Object Selector and we can continue by placing the J2 connector above and towards the right of the battery.

You should notice both during placement and afterwards that there are green 'elastic' lines between the two components and also a yellow arrow from each component. The green lines are ratsnest lines and indicate connections that have to be made between the two parts, whilst the yellow arrows (named 'force vectors') indicate a preferred position for the part to minimize ratsnest distance. The force vectors are provided as guidance only and are based solely on logic to reduce ratsnest lines. Since we will be using the earlier screenshot to dictate positioning we can turn them off.



PICDEM2 board once netlisted

From the View Menu, select the Layers command. The resulting dialogue form shows all the layers in ARES with colour and visibility configuration options. All we need do here is deselect the checkbox for the 'Vectors' layer and exit the dialogue.

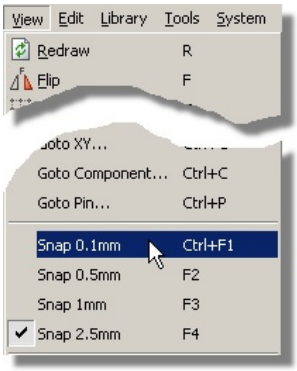


Visible layer settings

You can also launch this dialogue form by clicking the mouse on the status reporting bar at the bottom of the application.

It is important to remember that this dialogue form controls visibility only; to control whether objects on a layer were selectable/editable we would use the Selection Filter which is discussed in more detail later in the documentation

You may find that you need more control over the positioning in order to move connector J2 into position. Remember that objects are linked to a snap grid so all we need do is reduce the snap grid via the options on the View Menu, for example to 1mm snap.

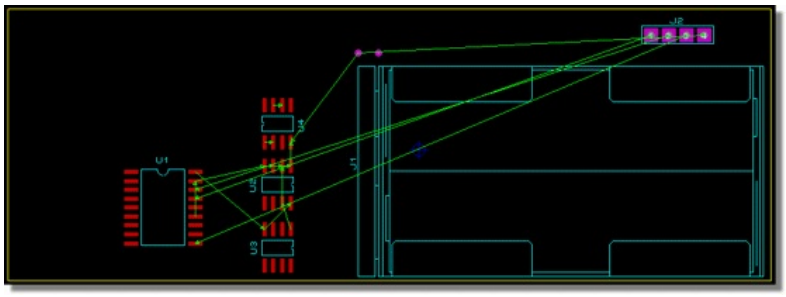


Snap Settings in the View menu

If you are working in imperial units you can either change the snap to 25th or use the 'M' shortcut to switch to metric units

Now that we have the granularity we need, simply right click on the part to select it and then drag it into the desired position. We can move the connector closer to the board edge if we move the part label underneath the part. Again, the process is identical to the one we covered in the ISIS tutorial – remember this time to right click on the label itself and not the component body.

If we now consider the other principle components we can go ahead and place U1 (dsPIC33), U2 (I2C memory), U3 (temp/humidity sensor) and U4 (dual op-amp) in exactly the same way, such that we end up with something like the following screenshot.



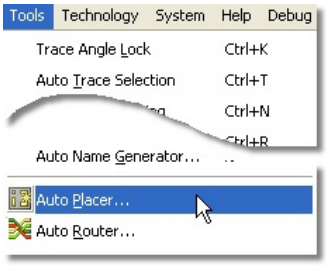
Placing packages / footprints on the board

Note that the S08 footprints used for the U2, U3 and U4 components have been rotated appropriately to reduce ratsnest length. This is best done at placement time as the ratsnest display will update live to give visual guidance.

Start by placing U1 in the normal way and then start placement of U2 by left clicking once on the Editing Window. Now use the '+' and '-' keys on your numeric keypad to rotate the component as you move it into position, left clicking the mouse again to commit placement as before. Repeat the process for U3 and U4, moving or rotating parts after placement (right click and then context menu options) until your board approximates the previous screenshot.

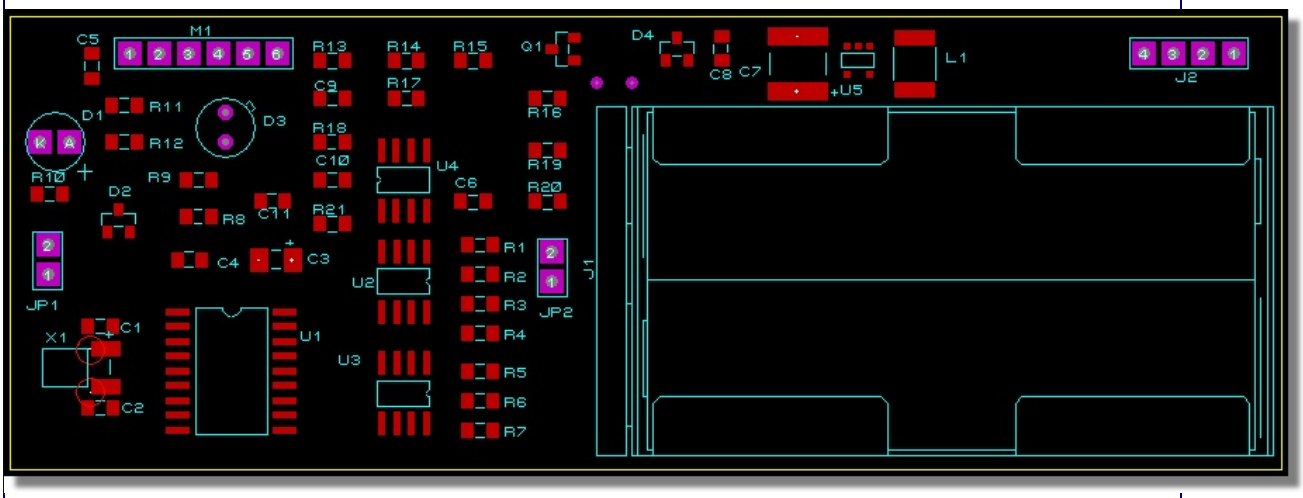
Generally speaking you have two options for placement of the board. You can either manually place the components in their positions or, if you have the Advanced Features Set (Proteus PCB Design Level 2, Level 2+ or Level 3), you can use the Autoplacement feature to get all the components on the board and then move them into the desired locations. In either case, you may find it useful to temporarily disable the ratsnest lines during placement; remember that this is controlled from the Layers dialogue form as discussed earlier.

The autoplacer can be invoked from the Tools Menu in ARES and for our purposes all the default options will suffice.



Select the Autoplacer from the Tools menu

Whichever route you follow the task is to lay out all the components on the layout, using the following layout as a reference. Try to leave some space at the bottom of the board so that we can run traces down and along from connector J2



The board should look like this once all the components are placed

The following points are useful to bear in mind as you continue placement:

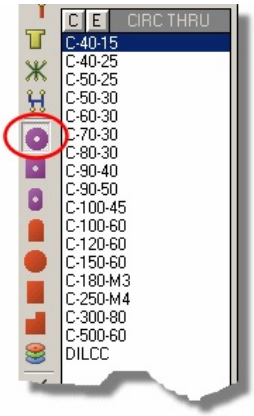
- The middle mouse will zoom you in and out as you place components (as will keyboard shortcuts F6 and F7)
- Right clicking on a component once placed will present a context menu option, allowing you to move, rotate or delete the part.
- Changing the snap settings to a finer grid will allow more accurate positioning at the expense of more precision being required to select the part.
- If you place a part in an illegal position (e.g. over another part) you will get one or more design rule violations. For now, simply move the part to a legal position – we will look more closely at design rules in the next chapter.

Design Rule Checker will find 2 errors

Once you feel comfortable with placing and moving parts, feel free to move on to the next section. We will load a board with all the parts placed before we discuss design rules and routing.

Mounting Holes and Pad Styles

Before we move on to look at connectivity considerations we should complete the physical layout by placing mounting holes for the board. In our case we want to use pad with a 3mm hole and a diameter of around 0.18in and to position them conveniently for mini-locking PCB supports. The first thing we need to do is switch into circular through hole pad mode and scan for a suitable pre-supplied pad in the Object Selector.



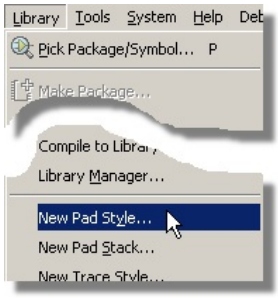
Through Hole Pad icon

The nomenclature of pads in ARES is designed for easy reading and tends to follow the following format:

`<PAD TYPE> - <DIAMETER/SIZE> - <HOLE>`

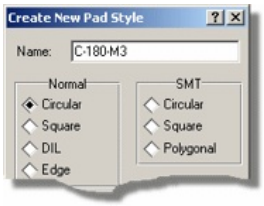
Units are in imperial unless prefixed with an 'M' so for example a C-40-15 is a circular pad with a 40th diameter and a 15th hole and a C-200-M3 is a circular pad with a 0.2in diameter and a 3mm hole. Our spec requires a pad with a 0.18in diameter and a 3mm hole, which we can see does not exist in the pre-supplied set. We therefore need to create the pad as follows:

1) From the Library Menu select the New Pad Style command.



Select New Pad Style from the Library menu

2) Enter a name for the pad; we recommend you follow the standard naming convention (i.e. C-180-M3).



Create a name for the new pad style

3) Specify the pad type – in our case we want Circular through hole pad.

4) The diameter of the pad is 180th or 0.18in.The drill mark is the size of the mark output in a drill plot, 30th would be fine. The drill hole needs to be 3mm and the guard gap should be enlarged to 20th The guard gap is the amount by which the pad diameter is expanded on the resist plot.

Setting the dimensions of the pad

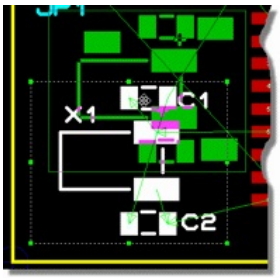
5) At the bottom of the dialogue form we have the choice whether to make this pad style permanent for future designs (Update Defaults option) or local only to this design (Local Edit option). Unless there is a particular reason for a local edit only we recommend that you leave this on the default setting.

6) When you exit the dialogue form you should see that the new pad style is available for placement from the Object Selector.

C-180-M3 Pad style in the Object Selector

When entering specific values you may find it easier to simply type in the values that you want rather than using the up and down arrows beside the controls.

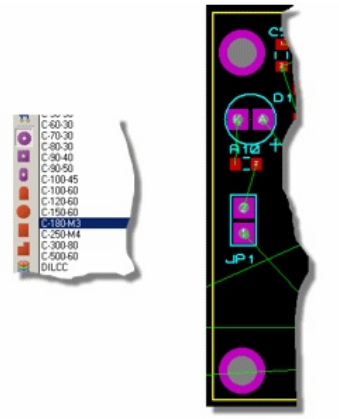
We are going to want to place two mounting holes at the top and bottom left of the board and then a third at a specific location (for mini-locking PCB supports). You may therefore have to move some circuitry out of the way to make room; for example, the crystal block at the bottom or the pressure tranducer at the top. As we've seen earlier we can easily do this by entering selection mode, drawing a selection box around the circuitry and then dragging to a new position. This is shown below with the crystal circuitry.



Moving the Crystal

Remember that you can resize the selection box to include or exclude items using the drag handles if you don't get the right size the first time.

Once you've cleared some space, go to the circular PTH pad icon again, select the C-180-M3 pad style and place two at the top and the bottom left of the board.



Select the correct pad from the Object selector and place the holes on the Layout

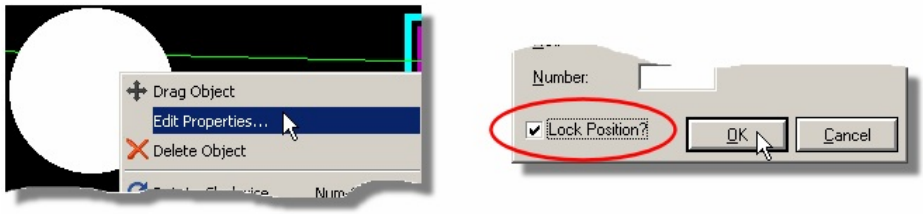
We want to position the third mounting hole above the battery and in a reasonably accurate position. Specifically, we want to place the hole 35mm up from the bottom of the board and 87.5mm along from the left. We've already set our global origin to the bottom left of the board so this is set to be the reference point for the co-ordinate system. All we need do therefore is move the mouse up and right from the bottom of the board edge until the co-ordinate display reads correctly (you may need to switch into Metric units using the 'M' shortcut key toggle).

Placing the third drill hole

If required, we can move the DC/DC converter circuitry as before, then switch back to pad mode and place the final mounting hole at the correct co-ordinates.

Alternatively, having set the output origin and know the co-ordinates you can right click on the mounting hole and select the move-to command from the context menu to programatically position the part.

With position sensitive components it is often useful to ensure that they cannot be nudged or moved inadvertently after placement. You can lock any object in position by moving the mouse over the object until it is encircled by a dashed line, right clicking on the object and selecting Edit Properties from the resulting context menu. The Lock Position checkbox can then be selected to prevent movement or deletion of the part.



Edit the pad and tick the Lock Position box

We have now completed the physical layout of the board. If you decided not to follow the full layout process you can load a version of the board in its current form from the `..\Samples\Tutorials\` directory of your Proteus installation (`dsPIC33_REC_u.lyt`)

Design Rules and Net Classes

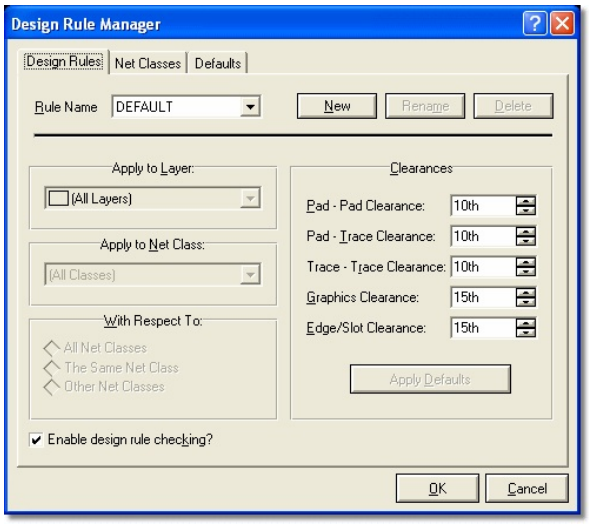
Now that we have a placed board we need to configure the software to obey any design constraints or electrical considerations relevant to the layout. We can do this largely from a single dialogue form called the Design Rule Manager. Start by launching this dialogue form now

from the Technology Menu in ARES.

Launch the DRM from the Technology Menu

Design Rules

The first tab of this dialogue form allows us to configure constraints and minimum clearances for the layout. We have a DEFAULT rule loaded which must apply to all layers and all net classes, providing a set of clearances between objects equivalent to manufacturing guidelines.

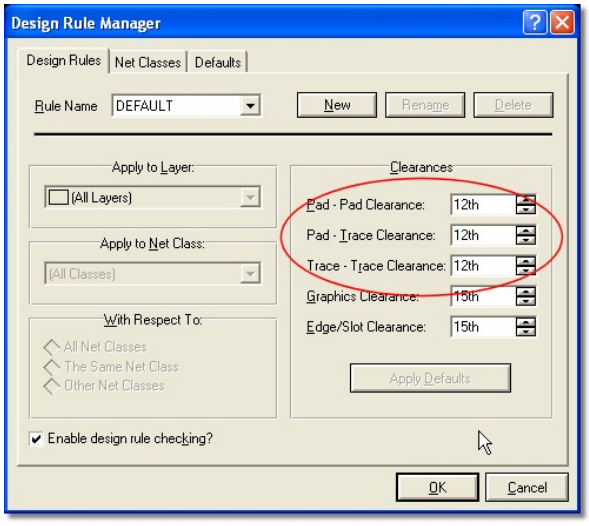


The Design Rule Manager dialogue form

This rule and these clearances are created automatically for each layout in order to provide a minimum set of constraints for a board. You can change the values of the default clearances applied to new layouts via the Design Rule Manager on the *Technology* menu..

The first thing we need to decide is whether a single rule is suitable for all layers and all traces on the layout. It is possible to create as many new rules as required and we can limit their effect to a given layer and/or a given set of connections (net class). A great deal of information on this is available in the reference manual but for our purposes we can manage fine with configuration of the existing rule.

Given that this equipment is intended to work out-doors we'll need to increase the clearances between pads and traces to improve the insulation against condensing moisture. A 20% increase should be sufficient so we need to change the Pad-Pad, Pad-Trace and Trace-Trace clearances from 10th to 12th.



Setting the clearances to 12th

The graphics and edge clearances are fine at the default values and, as don't need to create additional rules we can move on to the net classes tab of the dialogue form.

Net Classes

This is the place where we configure trace and via styles and control which layers we route on when auto-routing the board. The selector at the top allows us to switch between different net classes and configure each separately.

Let's start with the POWER Net class, which should be the default selection. As we discussed in the ISIS Getting Started Guide any nets that include a power or ground terminal are automatically assigned to the POWER net class, unless manually overridden.

We'll set the track size to 25th, not so much for current considerations but to reduce track impedance (we'll also place a single low impedance ground plane later to help with this). In ARES nomenclature this corresponds to track style `T25`.

The neck style – if configured – specifies the track style for necking on the current net class. We won't need to worry about this and can therefore leave it at it's current setting.

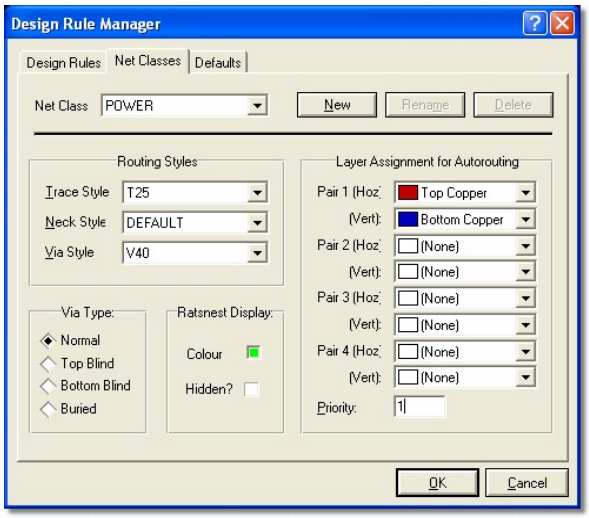
Given that we don't really have current constraints the choice of via's is a trade off between the higher manufacturing costs of smaller via's and the decreased routing quality of large via's. For the current this circuit is dealing with and for a standard 1.6mm FR-4 support plated with 35um copper, vias with the standard 0.4mm hole are a good compromise. For the power tracks we need a suitable annular size and 40th is a reasonable choice. We can therefore set the via style to be `V40`.

> If you are not sure of the characteristics of a particular style, you can view them by selecting the appropriate mode icon (via, track, pad, etc.), highlighting the style in question in the Object Selector, and then clicking on the 'E' button at the top of the Object Selector.

The options at the bottom of the dialogue form allow us to change the via type (clearly not relevant on a two layer board) and also to change the colour or visibility of the ratsnest lines. The latter can be useful if we are manually routing and wish to quickly distinguish between power and signal connections.

The layer assignment pairs on the right hand side tell the autorouter which layers to route on for multi-layer boards. Again, for our two layer board, there is nothing to configure here.

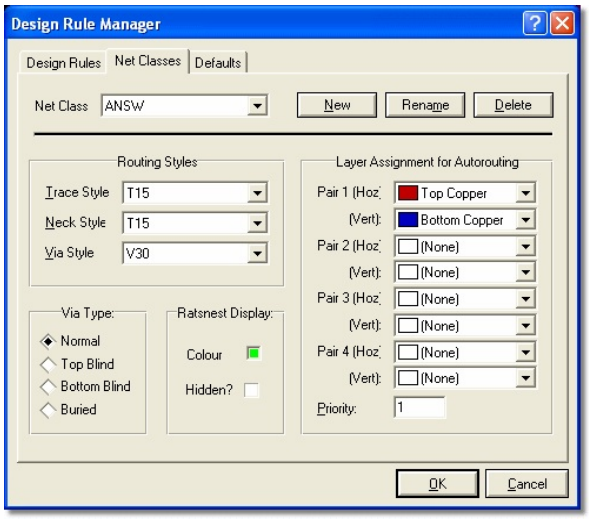
We should now have finished setting up the `POWER` net class – your dialogue form should now be configured as per the following screenshot.



Select the POWER class in the Net Classes tab

Lets move on to the next net class in the selector; the `ANSW` class. You may remember that we specifically named this net class in the ISIS Getting Started Guide in order that the 5V switched power supply for the analog circuitry (the output of the DC/DC converter) could be handled separately in ARES. What we want for these connections is a track size larger than the standard `SIGNAL` net class but smaller than the `POWER` net class, so lets change the Trace

Style to T15 (15th track). For consistency, the neck style can also be T15 and we can use a smaller via (with the same 0.4mm hole) by selecting the V30 Via Style.

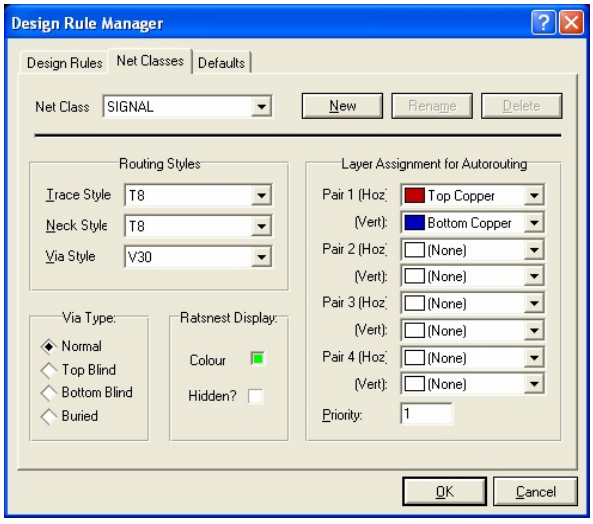


Setting the ANSW class

If you don't have a V30 via style you can create one easily by going to the Library Menu, selecting New Via Style and filling out appropriately.

Creating your own net classes in ISIS is extremely simple and provides you with the flexibility you need both with regard to trace and via configuration but also with design rule constraints

The final net class we have is the standard SIGNAL net class that encompasses all non-power and unspecified connections. We've deviated from the widely used "8/10 Rules" (8th trace, 10th clearance) in order to compensate for outdoor usage but the standard 8th track width (style T8) is fine. We can also set the neck style to 8th and keep our standard V30 via style for these connections.

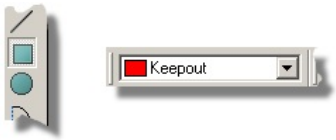


Setting the Signal class

Having finished configuration, exit the dialogue form to return to the layout.

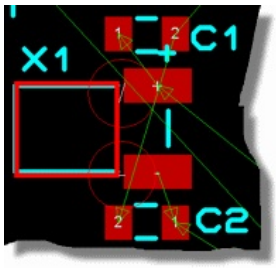
Keepout Areas

We can also introduce tracking constraints by limiting areas in which tracks can be placed. A good example of this is the crystal towards the bottom left of the layout where we do not want tracks under this area. To form a keepout area, start by selecting the 2D graphics icon and changing the layer selector to be keepout.



Keepout icon and it selected in the Layer Selector

Next, place a small box around the silkscreen of the crystal in exactly the same way as we did for the board edge (left click to start placement, drag out the area, left click again to commit placement).

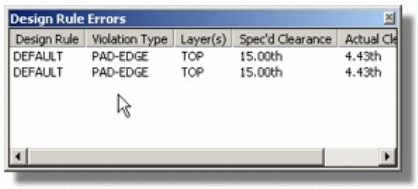


Keepout causes 2 DRC errors

Unless you are very skilful you will be presented with a box indicating that design rules have been detected.

A message will alert you to the DRC errors

Checking the box on the dialogue form will prevent it from appearing in the future. If you left click the mouse on the DRC section of the status bar a small window will appear providing information on the errors. You should see that they are of type PAD-EDGE as the pads of the crystal are closer to the keepout graphic than the 15th specified in the Design Rules.



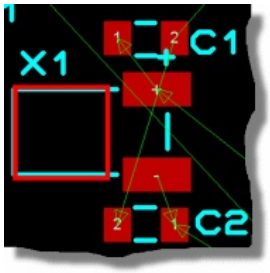
The DRC window

We have two options here:

1) Ignore the DRC errors. The keepout graphic will not impact on connectivity.

2) Move the graphic to a legal distance from the pads. The easiest way to do this is to first change the snap level down (View Menu), right click on the graphic and select the Drag Object context menu option, and then change the snap level back when you are finished.

When you are finished your keepout area should look something like the following screenshot.



Final Keepout area

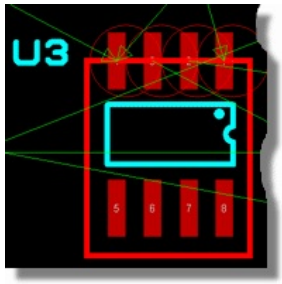
We have a similar problem around the temperature/humidity sensor (U3). We are going to want

a slot machined on the board to cut the thermal path and reduce the measurement errors of the sensor (we want to measure environment temperature, not the one conducted by the PCB). We therefore need to ensure that no tracks are placed in the area we are going to slot through:

1) Select the 2D Graphics rectangle icon and then change the layer to be KEEPOUT.

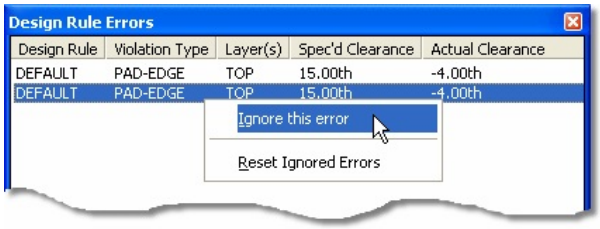
Select the 2D square mode and Keepout in the Layer selector

2) Place a rectangle around the bottom half of the IC as shown below.



Keepout creating DRC errors

This time we cannot move the keepout to remove the DRC errors as the positioning is sensitive. Instead we can manually override the errors by launching the DRC dialogue, right clicking and selecting the Ignore option from the resulting context menu.



Right click on the error and select Ignore this error

Routing the Board

Having configured the board constraints we can now move on to actually making connections and routing the board.

Placing a Route Manually

Let's start by manually placing some tracks on the board. Typically, we would manually lay out connections where a specific path is desired for the track or where we need greater control over track position. On our layout we want to make sure that our connections from connector J2 follow a sensible path around the board so we can start here and route them manually.

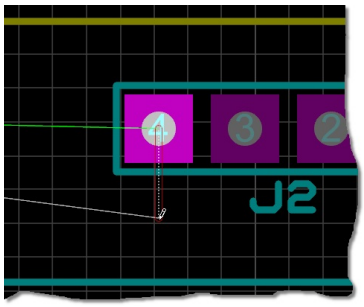
Start by selecting track mode at the left hand side of the Object Selector and changing the layer selector to be on Top Copper.

Trace mode icon and the Layer Selector

If we look at pad 4 of the J2 connector we can see that the closest rastnest line is directing us across to the GND pad on the step converter. This is not ideal as we would have to navigate the mounting hole and then track into the small SMT pad. Instead, we will route this down and across the bottom of the board to a more convenient ground connection.

ARES features a sophisticated 'follow me' routing algorithm for manual routing in which the

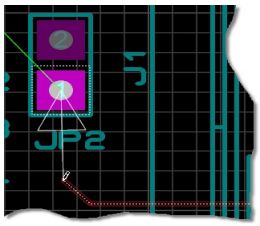
route being placed will follow the path of the mouse as best it can while obeying all of the design rules for the board. You start track placement by left clicking the mouse on pad 4 and then move the mouse downwards.



Routing on Top Copper

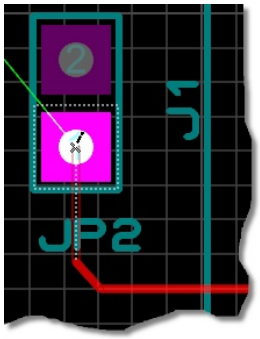
You should see that the closest legal destination for the track is now highlighted in white. This will update as we move the mouse and since we are not going to route to this destination we can ignore it for now. When the mouse approaches the bottom of the board, you could just change direction to the left and the track will corner to follow the mouse. However, since we want a tighter corner it is better to left click the mouse to place an anchor and then change direction to the left.

You will see that as we move further to the left the ratsnest guide will change to show us that we can terminate the track on pin 1 of jumper JP2.



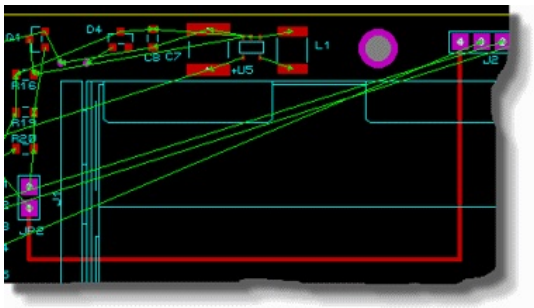
Routing to pin 1 of JP2

Once we are underneath the jumper we can change direction and move upwards towards the destination pin.



Connecting to Pin 1 of JP2

Finally, we can clicking left over the pin to finish placing the route. This will both commit the route and remove the ratsnest line corresponding to this connection.



Route completed from J2 pin 4 to JP2 pin 1

Note that we did not need to select the width of track to route with as we configured this in the previous chapter. ARES recognizes that we are routing a track on the GND net, applied the rules for the POWER net class and selected the specified 25th track style for us

Manual routing is probably the most common action you will perform with the software and it is vital that you understand how it works. The basic rules of operation are :

- Left click on pad, track or zone border to begin routing from that object.
- Left click at any point during routing to commit the route up to the mouse position (we call this anchoring).
- Right click to terminate the route at it's last committed / anchored point.
- ESCAPE key to abandon route entry completely.
- SPACEBAR to float a via on the end of the route and left click to then place the floating via.
- Double left click to drop a via at the current mouse position.
- Move the mouse backwards over existing tracking to rub-out.

We strongly recommend experimenting with manual routing for a while on this board until you are comfortable with how it works. The following is a summary of how to perform common actions; try these while routing any of the remaining connections.

Panning and Zooming

While placing a track you can use the middle mouse wheel (or F6 & F7 keys) to zoom in and out during routing. Panning will happen seamlessly when the mouse is at the edge of the Editing window during routing.

Placing Anchors

The follow me routing algorithm will move the track being placed according the way you move the mouse. If you want a track to follow a particular path then you need to help by left clicking whenever you change direction. This places an anchor or commits the route up to the mouse pointer such that the follow me router will not change it. You will see this happening as the outline track becomes solid.

Getting Stuck / Re-routing while routing

Since the manual routing system obeys the design rules for the board you don't need to worry about clearances while placing routes. You can however route to a place where you are blocked (the routing icon will change to a no-go sign at this point). Often you just place a via and continue on but sometimes it is better to rewind and try another path across the current layer.

Moving the mouse backwards over the route being placed will rub out that portion of track so to rewind and change direction just move the mouse back to the last good point - anything you had placed from that point onwards will be removed.

On densely packed boards in particular, speed of mouse movement is an issue. Remember that you are guiding the placement so moving slowly through tight spaces will work far better than ripping the mouse from source to destination.

Placing Vias

If you double click during placement you will place a via at the point the mouse is at and can then continue routing on the associated layer. If you press the spacebar you will float a via on the end of the mouse and can then position the via manually before placing with a left click. Using the spacebar has the advantage of snapping to legal objects (e.g. via under SMT).

In either case the placement of the via is also design rule aware and ARES will not let you drop a via in an illegal spot. If you use the floating via method (with the spacebar) ARES will attempt move the via to the nearest legal spot. This can be extremely useful on busy boards or when you want to butt a via tightly against another object .

Layers used for via's are defined in the Layers Pairs dialogue on the Technology Menu. Examples of routing with vias and discussion on layer pairs follows further in the tutorial.

Object Hugging

Since the design rules are all live during placement it is quite easy to hug a track to another track or to wrap a track around an object. If the mouse is over an object where the route being placed cannot travel then the route will follow the mouse path as closely as it legally can, essentially hugging the barrier object.

Abandoning a Route

- If you want to abandon the route at the last committed (solid) segment then right click the mouse.

- If you want to abandon the route at the mouse pointer then left click to commit up to mouse pointer and then right click to terminate the track.

- If you want to abandon the route completely then hit the ESCAPE key..

Making Connections

If you are connecting directly to a pad then a left click of the mouse will complete the connection and terminate route placement.

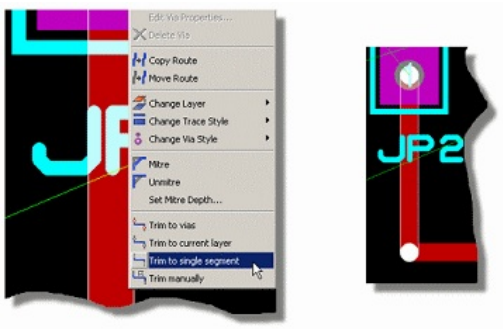
If you are connecting to a track then left click will commit the placement and right click will terminate the route forming the connection. The same procedure applies if you are connecting to a zone except that you must connect to the zone border.

Several short support movies showing different routing techniques are available to all professional users on our support forums (http://support.labcenter.co.uk). You will need to register and be activated in order to see these member forums.

Deleting a Route

Once the route has been completed we can delete either the full route or a specific part of the route if we are not happy with the placement. Let's assume that we were not happy with our placement of the last piece of track (upwards to pad 1 of JP2).

Start by right clicking on the track over this segment of track. This will highlight the entire track and the Delete Route option near the top would then remove the complete trace from the board. However, we have far more control if we use the options at the bottom of the context menu. In this particular case, select the Trim to Single Segment option; this will change the selected area of track to the segment we have clicked on.



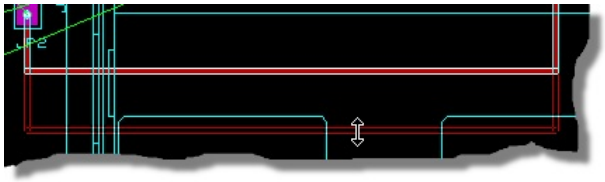
Trimming the trace to a single segment

Next, right click on the highlighted segment and select the Delete Route command; this will delete only this segment of the route.

Deleting a trace segment

Finally, zoom in, change the snap settings if need be and replace the route from the existing track to the terminating pad.

If things have gone more badly wrong you can simply delete the entire route and start again; what you are aiming for is something like the following screenshot:



Dragging a trace

Editing a Route

It is often the case that routes need to be nudged or moved into position after placement and we definitely don't want to be deleting and replacing routes or parts of routes all the time. To take an example, lets move our track lower down the board a little, towards the bottom board edge.

Start by right clicking on the horizontal section (segment) of the track; this is the section we want to move. Next, select Drag Route from the resulting context menu, move the mouse down to 'pull' the track into the desired location and then left click again to commit placement.

Moving tracks works as we've seen on a track segment. Using the Trim Manually option on the context menu for a track allows you to define your own segment and you therefore have complete flexibility with altering track topology.

Layer Pairs and Manual Routing

The track we have placed is of course only on top copper, whereas we often want to via up and down through the board when routing. ARES handles this with a concept called Layer Pairs. This means that every layer on the board has an associated layer so that via destinations are known during placement. For a two layer board this is obvious with top copper being associated with bottom copper and vice versa but with multi-layer boards configuring the layer pairs (Technology Menu – Layer Pairs) can be an important step.

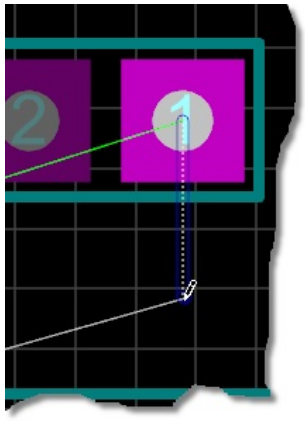
You can also use the Layer Usage command on the Technology Menu to define which layers are in use (and therefore appear in the Layer Selector). For multi-layer boards this is particularly useful as you can name the layers appropriately.

In our case, the default assignments are correct and no action is required. Let's place a couple more routes manually to see how this works. We'll look at connecting pins 1 and 2 of the J2 connector which are the transmission lines from the USART on the dsPIC processor.

We'll start with pin 1 and route from the pin on the bottom copper layer. Make sure that you have trace mode selected and then hit the space bar on the keyboard, noticing that this will toggle the layer between the two layers associated as a layer pair. If you have ended up on top copper simply hit the space bar again to set the layer to bottom copper.

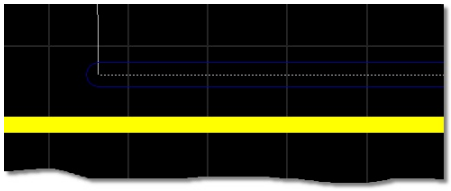
Trace mode icon and the layer selector

The processor is some distance from the connector and the easiest path would seem to be down and along the bottom of the board. Begin placing the track by left clicking on pin 1 on the connector and then moving the mouse downwards

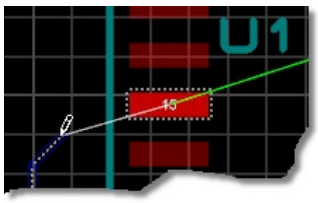


Routing from Pin1 of J2

When we get near the bottom of the board, left click the mouse to place anchors and guide the mouse to the left. To maximise space on the board you can route across the bottom of the board with the mouse over the board edge graphic - this will hug the track to the absolute edge of the board leaving only the Edge Clearance between the two.

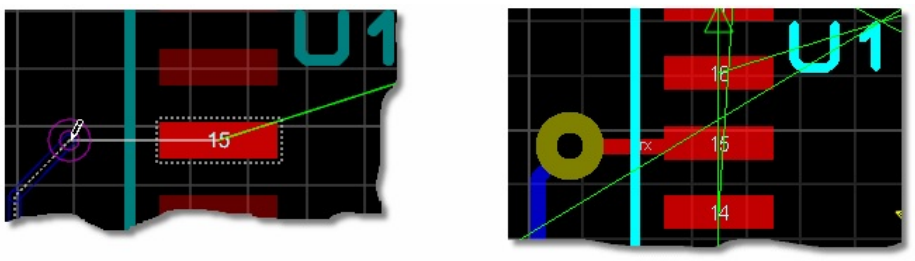


Once you get near the U1 IC bring the track up just past the right hand pads of the processor footprint.



Routing on bottom copper

We now need to place a via before we can connect to the SMT pad on top copper. As described above you can either double click the mouse at the point you wish to place the via or you can press the spacebar to float a via and then guide the placement point with the mouse before left clicking to commit. The latter method has the advantage of allowing you to butt the via as closely as possible against the pad thereby minimising the stub track length on top copper.

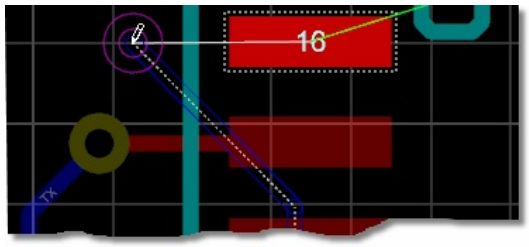


Placing a via and connecting to U1

Quite a lot has been covered in making these connections and, unless you are fairly skilful, there has probably been some finger trouble along the way. We'll continue with the other connections from the J2 connector to help familiarize ourselves with the techniques we've introduced.

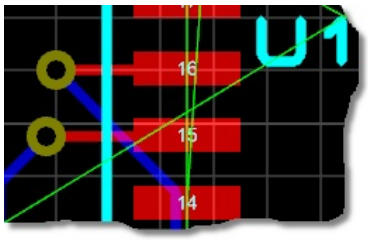
Pin 2 of the connector should follow an almost identical path to pin 1 and again we can start

placement on bottom copper (check the layer selector before you start placement; the space bar will toggle the layers). We can easily hug the track we have just placed by moving the mouse parallel to the path we want to travel but over the original track as shown below.



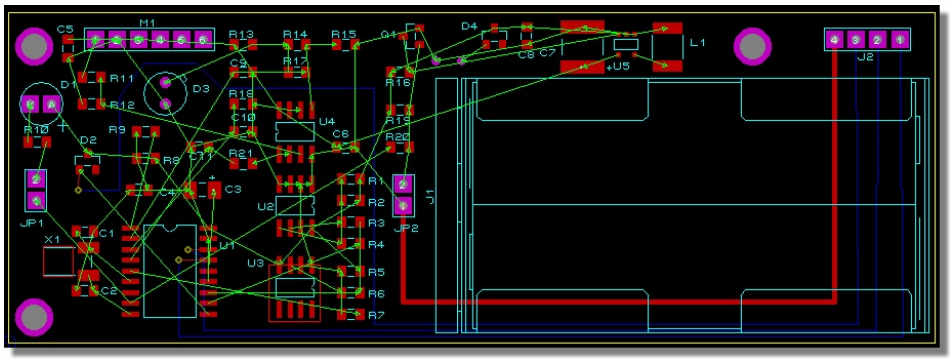
Routing on Bottom Copper

In order to make the connection, we can then route vertically underneath the pads and make a short 45 degree track before double clicking to place the via and complete the connection.



Routing a trace at 45 degrees

Of course there are many ways we can route the board and personal preference also plays a part. Feel free to experiment with the other connection on the connector, working around the battery holder, using the ratsnest to find a legal destination and then completing the route. The following screenshot is an example of what you might have when you are finished.

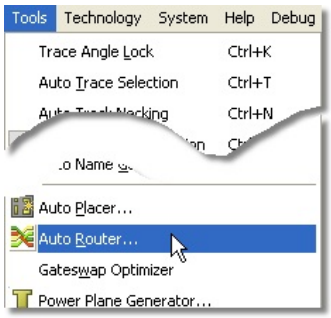


Manual routing finished

Basic Auto-Routing

From this point on we'll use the auto-routing engine to complete placement of the routes. As with the manual tracking the auto-router will obey all the design rules that we configured earlier.

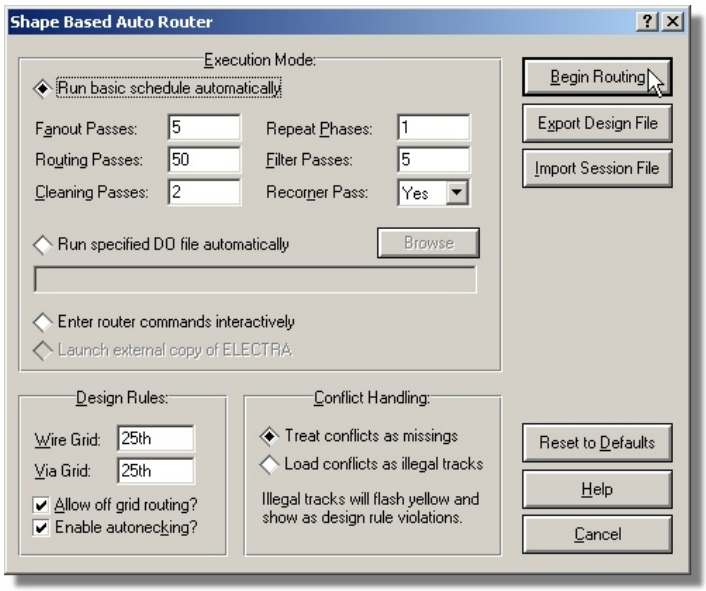
Start by invoking the auto-router from the Tools Menu in ARES or from the icon at the top of the application.



The Autorouter is launched from the Tools menu or the Autorouter Icon

The resulting dialogue form is reasonably complex but all fields have context sensitive help associated with them. You can also find additional information about the various type of routing

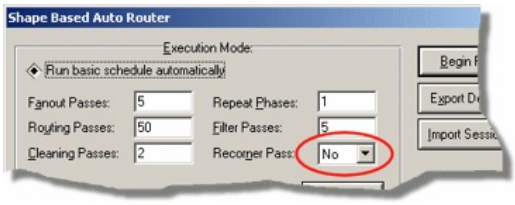
pass in the reference manual. For our purposes (and indeed for most boards of small/medium complexity) the defaults are more than adequate. We'll start by running in 'fully automated mode' which should be selected by default so simply hit the begin routing button to complete the remaining connections.



Autorouter dialogue form

Everything will happen quite quickly from this point onwards but you should see routing progress on the status bar as the engine works towards completion. Once the board is complete there are two immediate points of note :

- The auto-router has preserved those tracks that we placed manually and has not ripped and replaced them whilst working on the rest of the board.
- Once the autorouter has completed a final pass is made to corner the tracks. If you prefer this not to happen you can remove the Recorner Pass option on the dialogue form before invoking the router.



Select the Recorner Pass as required

The Selection Filter

Now that we have a completed board it's worth spending a little time looking at the techniques for selecting various object types on different layers.

ARES uses the Selection filter at the bottom left of the application window to determine which objects are available for selection at any given time.

The Selection Filter in the ARES layout editor

The left most button will determine whether the layer selector is active:

- When this is toggled off the selection applies to all layers on the board.

Layer selection off

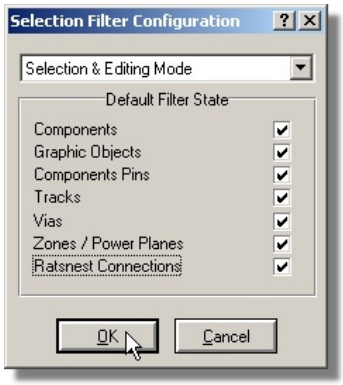
- When toggled on the selection applies only to the layer specified in the layer selector

Layer selection on

The other buttons represent different object types (tracks, components, graphics, etc.) and determine whether those object types are selectable. Hover the mouse over an icon for a tooltip description.

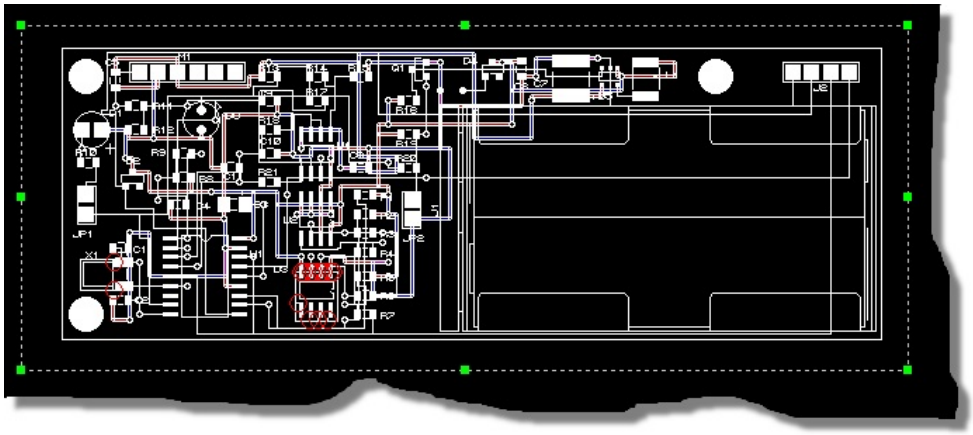
The Selection Filter allows you to select / deselect objects on the board

If you switch between the different modes of operation (for example from selection mode to track mode to component mode) you will see that the selectable object types will change according to the mode you are in. Whilst these are good defaults for normal operation you can change at any time simply by toggling on or off the layer switch or object type that you do or do not want to be selectable. If you find yourself changing selectable items regularly you can also change the defaults via the System Menu – Set Selection Filter command.



Selection Filter Dialogue form

Let's take a practical example and delete all the traces on the top of the board, except the one we have manually placed. Start by entering Selection Mode and left dragging a selection board around the entire board.



The entire board selected

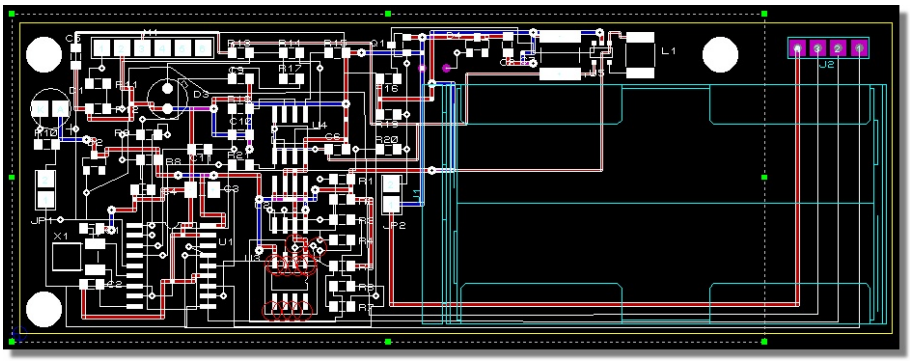
Next, deselect those objects that we do not want to delete, namely everything apart from tracks and via's. The tagged items will update automatically to provide visual confirmation of what is selected.

Uncheck what you do not want to be de-selected

Now change the layer to be Top Copper on the layer selector and toggle the layer button such that the selection filter applies only to the current layer.

Select Top Copper and uncheck the Layers icon

Use the green drag handle at the right of the tagbox to move the selection filter in until it does not encompass the J2 connector.



How the board should look once the selection box has been moved

Finally, use the icon at the far right of the selection filter to deselect items that are only partially inside the tagbox; in our case this will deselect the track we manually placed from J2, pin 4.

Disable full selection mode

We can now simply hit the delete button on the keyboard or right click inside the tagbox and select Block Delete icon to remove all the top copper tracks.

Bear in mind that the selection filter controls what objects are available for selection at any given time. If you ever find that you can't select an item the first thing to check is whether that object type is enabled for the mode that you are in. Alternatively, simply switch to Selection Mode where all items are selectable.

Advanced Auto-Routing

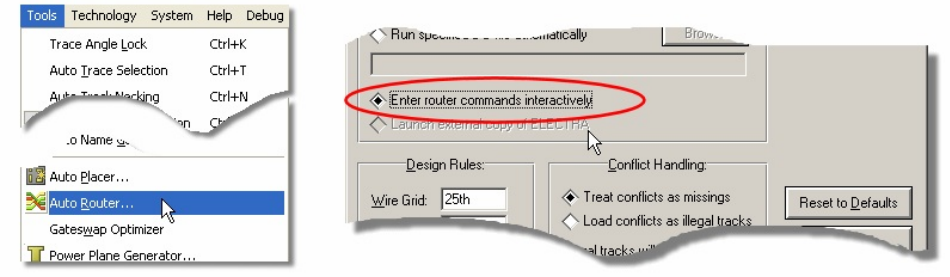
Given that we've ripped a lot of partials and removed via's as well we've actually made a bit of a mess here. Fortunately, we can fix the situation by simply re-invoking the auto-router and re-placing all our tracks. The router has it's own clean up phase so it will tidy up all the redundant partial tracks for us as it completes the new routing.

If you have a standard features version of Proteus (PCB Starter Kit, Level 1 or Level 1+) simply run the autorouter as we discussed in the previous section to re-route the board.

For those using the Advanced Feature Set, we'll use this section to show in brief some of the additional features available. These fall into two main categories:

- The ability to route only a specified area or set of connections.
- The ability to control the routing script; i.e. to determine which routing commands are executed and in which order.

Start by invoking the autorouter dialogue form (Tools Menu), switching to Interactive mode and then selecting the Begin Routing button.



Enabling the Interactive Autorouter

A command window will open at the bottom of the Editing Window, which will allow us to direct the routing progress interactively. ARES provides a rich command set to control routing, including such things as bend radius for mitering tracks and fanout length and direction from SMT pads. This is fully documented in the reference manual so we will concentrate on some

basic practical examples in this tutorial.

There are a couple of important points worth emphasizing at this stage:

- Any commands entered will action on a set of tagged connections or on the whole board if nothing is highlighted.
- Changing the circuit or switching modes will automatically exit the routing interface.

Essentially, what this means is that we can dictate what connections are routed by controlling which items are highlighted.

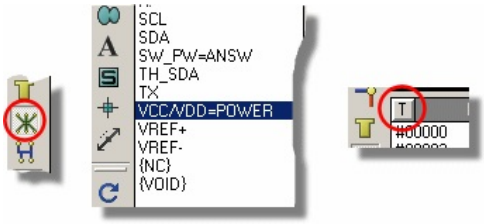
First of all let's clean up the loose tracking on bottom copper resulting from our previous rip of the top copper tracks. The basic syntax of most commands is:

`<command> <number of passes>`

so we can start by typing clean 2 to clean up the superfluous tracking.

Clean command in the router

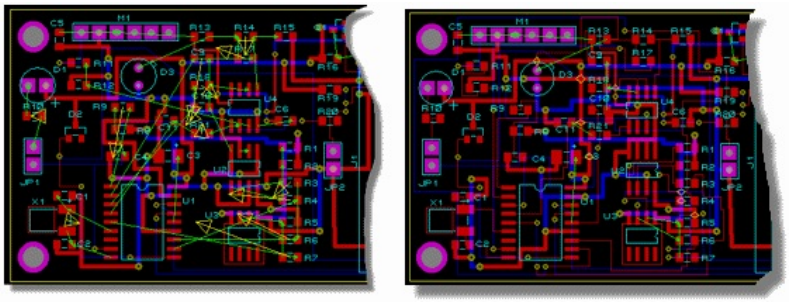
Next, lets assume that we want to route all the `VCC` connections. Select the `VCC/VDD=POWER` net in the Object Selector and click on the small 'T' button above the Object Selector to highlight all the connections on that net. Finally type in '`route 5`' to route these connections.



Tagging the VCC/VDD=POWER net

Clicking anywhere in the Editing Window will deselect the currently tagged items. When the focus is in the Editing Window you can also use the middle mouse (or keyboard shortcuts) to zoom in and out in the normal way.

Similarly, we can highlight connections on an area of the board and route them independent of the rest of the board. For example, right depress the mouse and drag a selection box around the left half of the board and then type 'route 10' to complete open connections in that area.

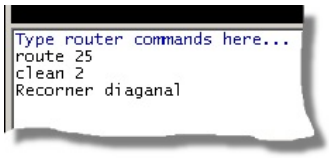


Routing Areas of the board

You may notice that this has left a couple of ratsnest lines or has not completed all of the connections. We could try to resolve this by issuing clean, filter and more route commands, but for our purposes it's easier to simply revert back to a 'whole-board' scenario and then completing routing. You can do this by left clicking on an empty area of the Editing Window to clear the selection. Typing `route 25`, followed by `clean 2` should see the board completed.

Route command

Finally, we can reduce trace length by typing recorner diagonal as a finishing touch.



Recorner command

Do note that there is much more flexibility in the command set than that shown in this tutorial, both in the number of commands and in the parameters that control the command actions. See the reference manual (Help Menu – AREA Help) for more information.

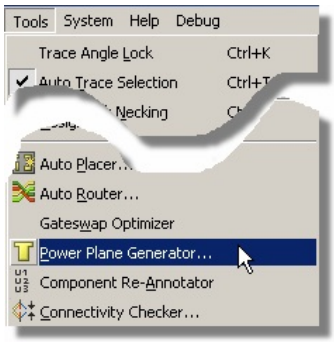
You can exit auto-routing at any time via the ESCAPE button on your keyboard.

Power Planes and Slots

Now that we have the board routed we are nearing completion of the actual layout phase. However, to minimize track impedance we are going to place a ground plane covering the entire board.

Placing Power Planes

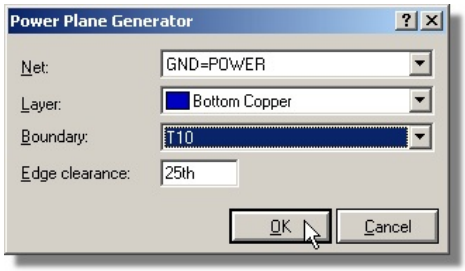
This is actually the easiest type of power plane to place and is available in all levels of the professional software. Start by invoking the power plane generator command from the Tools Menu



Power Plane Generator is launched from the Tools menu

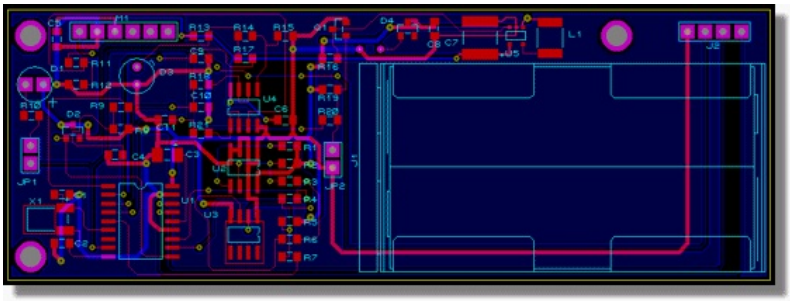
From the resulting dialogue form select the GND=POWER net, keeping the layer as bottom copper and setting the boundary style to be T10. This is the trace style in which the inner and outer boundaries of the zone are drawn and also determines the thinnest section of copper by which the power plane can make a connection. Setting this larger will prevent the copper flowing through small gaps (e.g. between pins) but making it smaller means that connectivity may be made only by thin sections of copper.

We can leave the clearance between the power plane and the board edge at the default value.



Power Plane Generator form

After you exit the dialogue form you should see that the power plane is generated across the entire board.



What the board looks like once a power plane has been generated on bottom copper

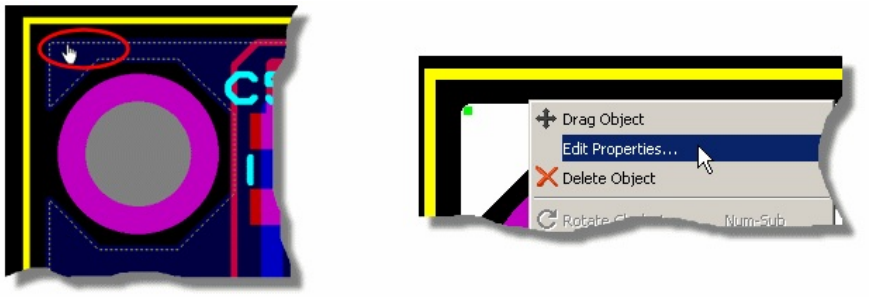
Nesting and Islands

There are several additional configuration options available to us now that we have placed the zone. Editing a zone is slightly different from most other objects in ARES in that you must right click on the border of the zone (to avoid continuous unwanted selections).

Firstly, check the selection filter and make sure that the zone object type is selectable (or switch to selection mode).

Make sure the Zone icon is enabled

Next, zoom in and move the mouse over the edge of the zone, right clicking the mouse when the zone is shown as active under the mouse. Select the Edit Properties from the resulting context menu.



Right click on the zone and select Edit Properties

The main options of interest are towards the bottom of the dialogue form and are explained below:

Relieve Pins

When checked pins on the same net as the zone will have thermal relief applied to them; the thickness of the thermal relief is determined by the Relief field of the dialogue form.

> The software will prevent you from using a relief track style that is larger than the boundary style thickness; this is to protect against the reliefs 'sticking out' of the boundary.

> The topology of the relief stems on a particular pad can be changed to a diagonal 'X' by editing the pad itself after placement. This is sometimes useful to maximise contact with a zone.

This option should almost always be left in it's default checked state.

Exclude Tracking

If this option is checked, the zone will treat tracks on its own net as obstacles. Otherwise the zone flows over such tracking, effectively ignoring it. Tracks on other nets, or loose pieces of track on no net are always treated as obstacles.

This option is normally left unchecked.

Route to this Zone

When checked this option allows the auto-router to route appropriate SMT pads to this zone

through an auto-via process.

Suppress Islands

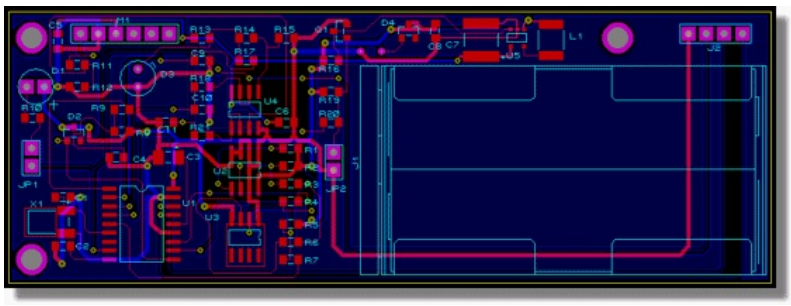
An island is defined in ARES as a block of copper or zone area in which no valid connections can be made. When checked, ARES will remove all such blocks from the board, leaving only areas of copper with connections.

Allow Nesting

It is quite common, particularly with larger boards, that the flow of a zone will encounter an obstacle through which it cannot pass. When checked, the zone will jump over the obstacle and continue flowing across the board.

This is a very useful option for example on busy boards or where you want to connect a pad where the topology of the board makes it impossible for a single zone flow to get through.

Given our relatively simple board, the default options are ideal for our purposes. Selecting the nesting option will have no effect for example as there are no inner boundaries on the board. However, if you want to experiment you should find that if you uncheck suppress islands option, the resulting zone populates far more of the board that in our initial configuration.



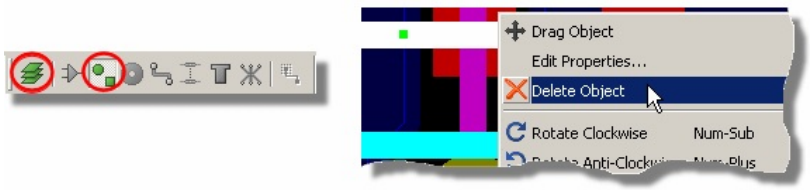
The zone with Suppress islands unchecked

We have covered only the basics of power plane functionality here relevant to our current design. For more information, including split planes, zone keepouts, stitching and bridge tracking please see the power planes section of the reference manual.

Slots

To complete the layout of the board we need to return to our temperature/humidity sensor (U3). In order to make accurate measurements, we want a thermal cutout around this part, ensuring that the temperature we are measuring is actually the environmental temperature and not conducted heat from the PCB.

You may remember that we placed a keepout area around this part to prevent the autorouter from placing tracks through the area we want to cut out. Start by zooming in around U3 and removing the keepout. You'll need to either switch to Selection Mode or to change the selection filter such that you can select 2D Graphics

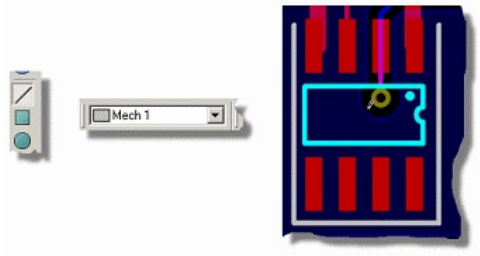


Right click on the Keepout and select Delete Object

While we are it, we might as well move across and repeat the process by deleting with the other keepout around the crystal.

Returning to U3, we now need to outline the cutout region. In ARES this is a two stage process; we need to place graphics appropriately on a mechanical layer and then designate the mechanical layer as the slotting layer when we generate output for manufacture.

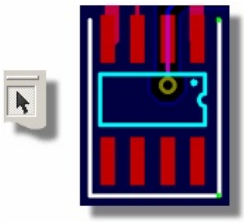
Select the 2D Graphics Line icon, change the layer selector to be MECH1 and then place three lines to form a 'U' shaped cutout.



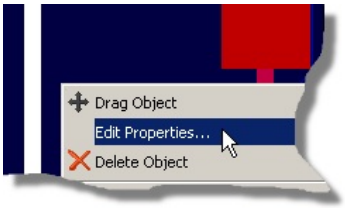
Placing a Slot

Finally, we can thicken these three lines to something more appropriate as follows:

1. Enter selection mode, hold the CTRL button down on the keypad and left click once on each line. This will select all three line.



2. Right click on any line and select Edit Properties from the resulting context menu.

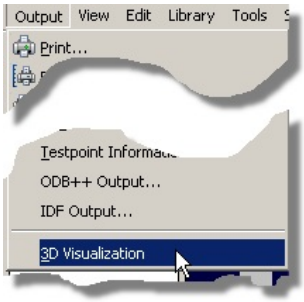


3. Uncheck the Follow Global checkbox, change the width to be 20th and then apply to all tagged graphics.

We will cover how to specify MECH1 as the slotting later in the documentation in the section on manufacturing outputs.

3D Visualisation

Now that the board is now routed and ready for production we first want to examine it in 3D in order that we can properly preview how it will look in real life and possibly make final design alterations prior to prototyping. Start by invoking the 3D Visualisation Engine from the Output menu in ARES.



Launch the 3D viewer from the Tools menu

Basic Navigation

The first thing we can do is view the board from different preset angles. Five preset views are supplied: top view, front view, back view, left view and right view and these are accessible via any of the following methods:

- Menu options on the View menu in the 3D Viewer
- From the navigation toolbar at the bottom of the 3D Viewer
- From keyboard shortcuts F8 through F12 whilst in the 3D Viewer.

3D navigation icons

Now that we can look at the board from a number of angles the next thing is to be able to look at it at a specific zoom level. Again, there are numerous ways to zoom in and out of the board:

- Roll the middle mouse wheel in and out (recommended)
- Menu options on the View menu
- From the icons on the Navigation Menu
- From keyboard shortcuts F6 (zoom in) and F7 (zoom out)

Experiment now with custom zoom levels and different preset views – whilst it is pretty subjective we envisage most users changing views via the navigation toolbar or keyboard shortcuts and using the middle mouse wheel to zoom in and out.

Bareboard View & Height Clearances

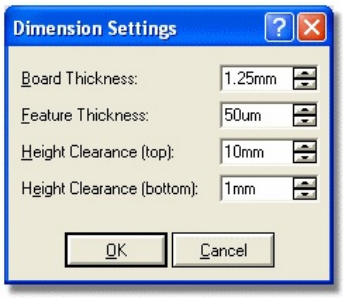
For inspection of resist and hole depth it is often useful to view the board without components. Selecting the bareboard view will remove all the physical components from the board.

Show / Hide components on the board

By contrast, if you need to fit the PCB into a chassis and wish to check height clearances you can enable the height boundary box via the icon at the bottom of the display

Show / Hide the Height icon

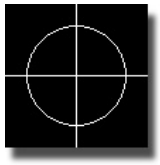
Specification of the height takes place from the Dimensions command on the Settings menu of the viewer.



3D Dimension settings dialogue form

Custom Views

The next logical step is to be able to customise the view. This works conceptually by 'attaching' the mouse to the camera such that as you move the mouse the camera moves to the area of the board that you are interested in. You can invoke the Navigation mode either from the View menu, the crosshairs icon on the Navigation toolbar or simply by clicking the left mouse button.



3D Cursor

You will know as soon as you are in Navigation mode as a crosshair cursor will appear over the mouse and your view of the board will change as you move the mouse. Using this together with the middle mouse wheel zoom will allow you to both 'fly pass' the board and to easily zoom in to closely examine a particular area of the board. Exiting navigation mode is as simple as right clicking the mouse.

For example, if we start in Front View (use the F9 keyboard shortcut) and we want to examine the resistors on the right we might proceed as follows:

1) Left click the mouse to enter navigation mode.
2) Move the mouse over the resistors.
3) Roll the middle mouse button to zoom in as required.
4) Right click the mouse to exit navigation mode.

The final necessary piece to completely customise the view is to allow users to spin or 'orbit' the board. This is done in navigation mode by holding down the left mouse button and moving the mouse. Essentially this will spin the board as you move the mouse – when you release the mouse button the camera will follow the mouse around the current view of the board as normal. Try this now, experimenting with different views of components on the board.

Remember that, if you are struggling to get the view you want you can use the keyboard shortcuts or navigation toolbar to return to one of the preset views. You should find however, that with only a little practise you become quite proficient at navigation.

To summarise:

- Left click enters navigation mode.
- Camera follows mouse around the board in navigation mode.
- Using the middle mouse wheel (or shortcut keys) allows you to zoom as you move the camera.
- Left depressing the mouse in navigation mode allows you to spin/orbit the entire board.
- Right click of the mouse exits navigation mode.

When you are finished close the 3D Viewer from the File Menu to return to ARES.

More information on 3D Visualization including creation your own 3D models, customizations and applying 3D data to legacy designs can be found in the 3D Viewer section in the online reference manual.

Board Output Options

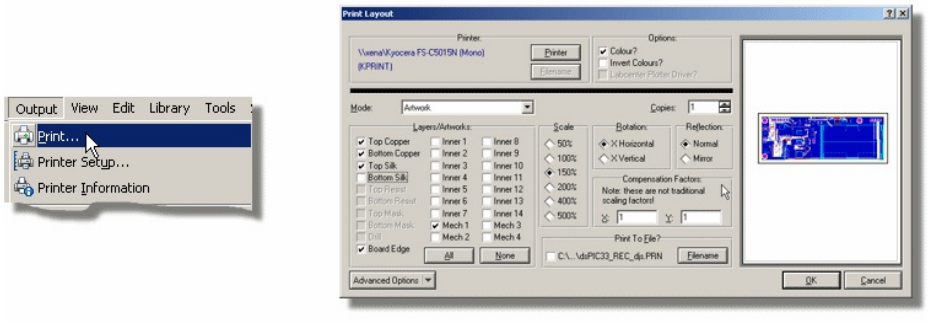
Last, but by no means least, we come to the crucial business of reproducing the pretty on screen graphics on paper or film. Under Windows, most hard copy devices are supported through the normal Windows printer drivers. Additionally, we supply our own drivers for pen-plotters, Gerber photoplotters and Excellon NC drill machines.

Printing

We will deal here firstly with printing to an ordinary Windows printer device - it is unlikely that you will have a photoplotter to hand! The first step is to select the correct device to print to using the Printer Setup command on the Output menu. This activates the Windows common dialogue for printer device selection and configuration. The details are thus dependent on your particular version of Windows and your printer driver - consult Windows and printer driver

documentation for details.

Then, with a layout loaded, invoke the Print command from the Output menu.



Printing is found in the Output menu and the Printer Dialogue form

The dialogue forms offer a number of controls, all of which have context sensitive help associated with them (context sensitive help on dialogue forms is accessed with a 'point and shoot' mechanism via the '?' key at the top right of the dialogue form). The default settings should do for getting something and you commence output generation by clicking on OK. Output can be aborted by pressing ESC, although there may be a short delay before everything stops whilst ARES and your printer/plotter empty their buffers.

With plotters in particular, you will probably need to experiment with pens, paper, and the various settings on the Set Devices dialogue form in order to get optimum results. Full details may be found under the chapter Hard Copy Generation in the reference manual.

ARES will remember your printer settings from the Printer Setup dialogue and maintain them independently of your printer settings for other applications. This means that you can configure a default set of printer options solely for use with the ARES application.

Output for Manufacture

ARES provides two main output options for board manufacture:

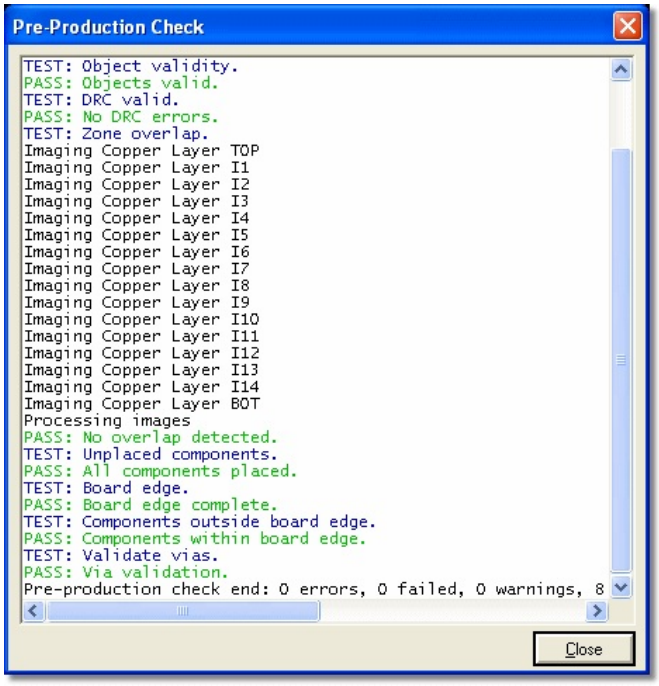
- Traditional Gerber/Excellon (available in all professional versions).
- ODB++ Manufacturing Output (available in Advanced Feature Set only).

From a user interface perspective both options are very similar but from a manufacturing perspective, the ODB++ option provides far more information than the older Gerber formats. Examples include:

- Inclusion of the connectivity information (the netlist) with the output fileset.
- Explicit support for plated/unplated specification on pads.
- Explicit support for fiducials.

All of this means that when viewing the output fileset from ODB++ the verification process is simpler and more complete. However, traditional Gerber/Excellon output is still prevalent in manufacture and would be sufficient for most purposes.

Regardless of your option the basic procedure is the same. When you invoke either of the output options you will most likely be prompted to run a pre-production check. This runs an automatic checklist to test for some common design errors and will report either a pass or a fail.

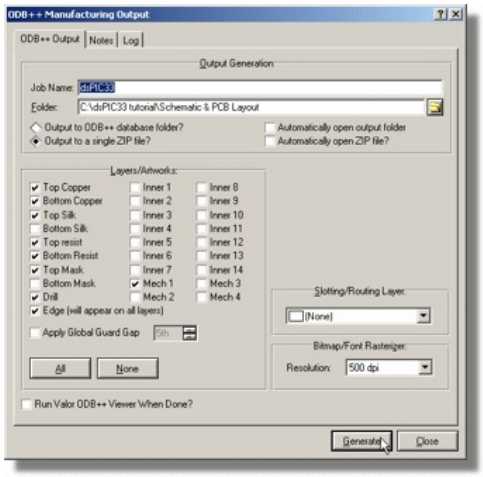


Pre-preduction checker

If your pre-production check reports errors we strongly recommend that you resolve them before proceeding to the manufacturing output dialogue. Please also note that pre-production check is an aid to the designer in quality assurance but not a guarantee; manual inspection of the layout is recommended and a prototype should always be made and tested before mass production.

More information on the Pre-production Check can be found in the CADCAM Output section of the online reference manual (Help Menu).

Assuming a pass you will then be provided with the manufacturing output dialogue form



The ODB++ Dialogue form

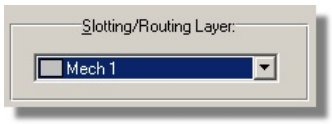
The options in the top section of the dialogue form are self-explanatory but there are some points of note regarding the rest of the configuration options.

The software attempts to populate the layer set with those layers used on the layout but this should always be verified. One of the most common problems with manufacture occurs when the manufacturer is not supplied with the full board information.

The Apply Global Guard Gap option will set the expansion of the resist plot around pads and vias to the distance specified when this option is checked. This will take effect on all pads and vias except those which have been manually altered on the layout. Some board manufacturers prefer to create the resist plot in house and this option could then be used to remove any resist expansion pre-manufacture. Unless otherwise directed we recommend that you leave this

option unchecked.

The Slotting /Routing Layer option specifies explicitly which layer of the board is to be used for defining the routing strokes for cutouts and slots. In our case, we have used MECH1 and we must therefore set this via the selector.



Mech1 selected in the Slotting/Routing Layer selector.

The Bitmap/Font Rasterizer option controls the thickness of trace used to render bitmaps and, more importantly, power planes. The higher the resolution the better tonality of the resulting bitmaps but the larger the files. It is also possible that some manufacturers have a minimum width requirement and in such cases it may be necessary to reduce the DPI settings to conform with this. Generally speaking however, the default settings are fine.

The option at the bottom left allows you to automatically load your output into either the Labcenter Gerber Viewer (CADCAM Output) or the Valor ODB++ Viewer (ODB++ Output). This is useful if you want to verify the output fileset before passing to your manufacturing house.

The ODB++ Viewer option will be disabled unless you have downloaded and installed the free Valor viewer from the Valor website.

If you intend to panelize the board you should use the CADCAM Output option, run the viewer when done and select Panelization Mode from the resulting dialogue form. See the reference manual for more information.

Finally, the Notes tab allows us to insert any relevant information or special considerations for the manufacturer. This is very important when we have specified a slotting layer as there is no standard way to pass this information through. A simple note that the information on MECH1 is for slotting will suffice.

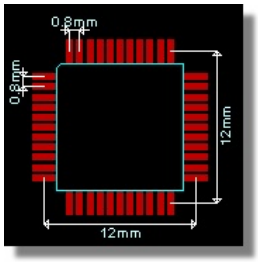
Having configured the necessary option we can generate the files and send for manufacture.

APPENDIX: Creating New Packages

ARES comes pre-supplied with a large quantity of footprints, and we have seen previously how to select and place these parts on to a layout. However, it may be necessary at times to create your own custom footprints or symbols – also a simple task with ARES – and this process is detailed below.

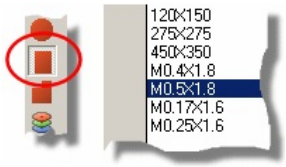
Drawing the Footprint

As an example, we will create a SQFP44 footprint with 0.8mm pitch and 12mm width.



SQFP44 package as it is in our libraries

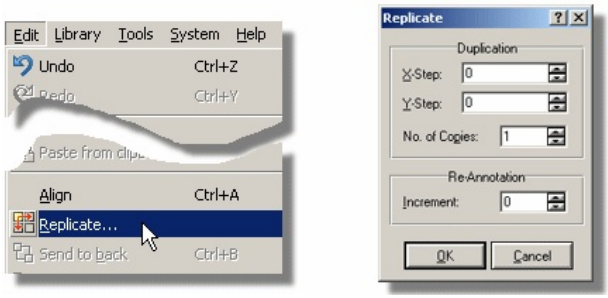
Start by selecting rectangular SMT mode. We want a pad 0.5mm by 1.8mm that should already exist as `M0.5x1.8`.



Selecting the correct SMT pad is vitally important

If the pad style does not exist you can easily create it as discussed earlier in the tutorial (Click on the 'C' button above the Object Selector).

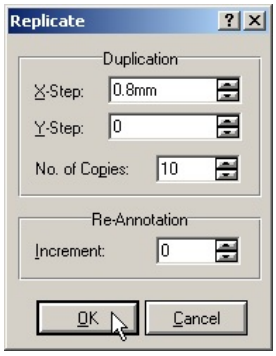
Make sure that the Layer Selector is on Top Copper and place one of the pads in the usual way. Right click to cancel placement immediately after placing a single pad and then left click on the pad to highlight it. With the pad highlighted, go to the Edit Menu and select the replicate command. Alternatively you can right click on the pad and select replicate from the resulting context menu.



Using the Replicate command to copy multiple pads

The replicate command will action on tagged objects so you should ensure that you have only the placed pad highlighted before invoking this command.

There are 11 pads on each side of the footprint so we need to replicate 10 times with an X-Step of 0.8mm as per the following screenshot.

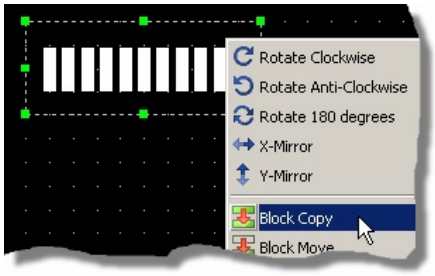


How the Replicate command should look

This will give us a single row of pads of the correct pitch (zoom in, change the snap settings and measure if you would like to confirm). The next step is to duplicate this row of pads to form the bottom of the footprint.

Start by right dragging a tagbox around the entire row of pads and then right click inside the selection box and launch the replicate dialogue form from the context menu. This time we want only one copy, 12mm down (or up) from the current row. Use negative co-ordinates if you want the duplicated row beneath the current one or positive co-ordinates if you want it above.

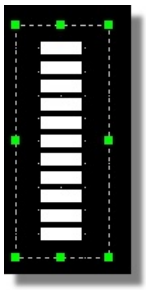
We now need to repeat the process with the other two rows. Drag a tagbox around a full row, right click inside the tagbox and select Block Copy from the resulting context menu.



Block Copying a row of pads

Move the mouse and drag the copy to a free area, left clicking to place and then right clicking to exit copy mode. Now, drag a tagbox around the row, right click and select the Block Rotate

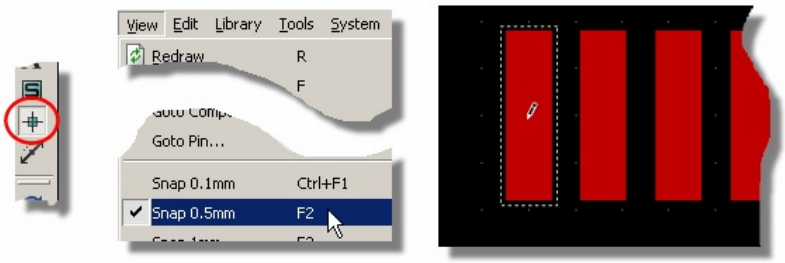
option, specifying a 90 degree rotation to align the pads correctly.



The row of pads should now look like this

In order to position this row of pads correctly it's best to first set place a marker on the location we want to move the row of pads to. For this footprint the centre of the top pad on the left hand side is 2mm below and 2mm to the left of the centre of the left hand most pad on the top row. This gives us enough information to accurately position the row.

Select marker mode in preparation and then move the mouse over the left hand most pad on the top row until it is encircled. You'll want to be on a fairly precise snap setting for this (e.g. 0.5mm or F2 shortcut key).



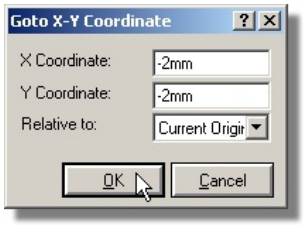
Placing a Marker on the first pin

If you find your snap settings in thou you need to switch to Metric mode, either by hitting the 'M' key on the keyboard or via the Metric option on the View Menu.

Now, hit the 'O' key on the keyboard to set a false origin at this location and then invoke the Goto-XY command from the Tools Menu.

'O' on the keyboard will set the coordinates to be "zero'

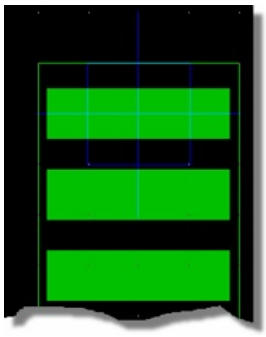
Type in –2mm in both the X and the Y co-ordinate fields (i.e. down and left) and make sure that the offset is relative to the current origin. Then exit the dialogue form and click the mouse twice to place the marker at this location.



The go-to dialogue form

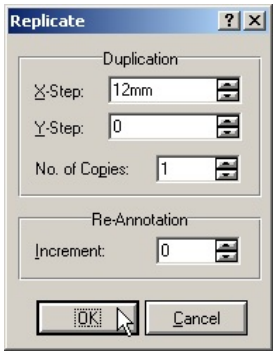
The marker we have placed will serve for now as the location we want to centre the top most pad of our third row. Simply draw a tagbox around the row, left depress the mouse inside the selection box and drag into position. You will almost certainly need to use the zoom commands (F6,F7 or the middle mouse wheel) and possibly also adjust the snap setting (CTRL+F1) to get

accurate positioning.



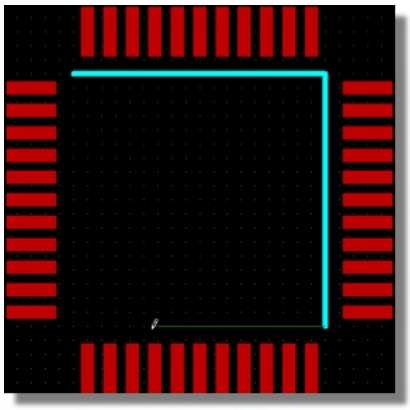
Placing the third row of pads

Having placed the row of pads right click on the marker and delete it – we'll come back to marker placement later on. The final stage of pad placement is to replicate the newly placed row of pads onto the right hand side. As before, draw a selection box around the row of pads and use the replicate command with an X offset of 12mm to duplicate the row.



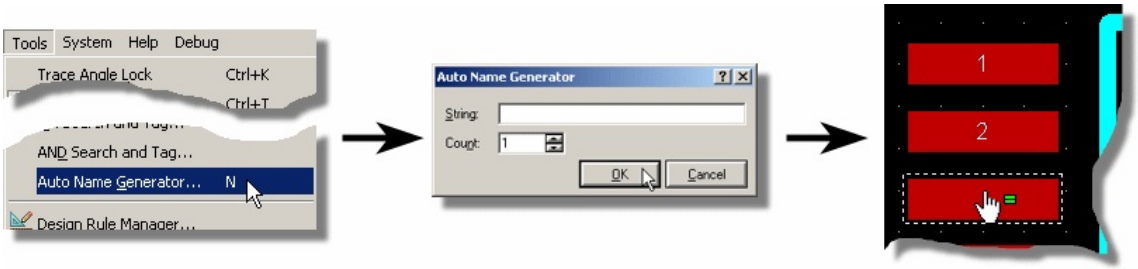
Replicating the fourth row of pads

Adding the silkscreen graphics is now straightforward. Select the 2D Graphics Line icon, make sure the Layer Selector is on top silk and place four lines along the inside edges of the pads to form a box. You'll find this much easier to do if you change the snap settings upwards (e.g. F2).



Drawing the Silkscreen on the inner side of the pads

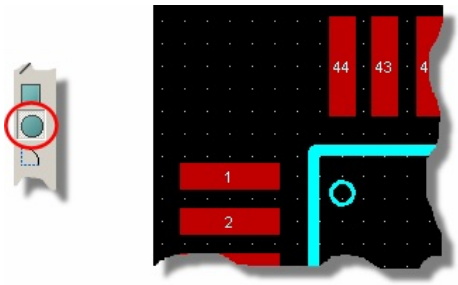
The next job we have is to number the pads. Start by invoking the Auto Name Generator from the Tools Menu. We don't need to enter anything in the string field here; simply leave the defaults and click on the pads consecutively to number them, Number 1 is the top pad on the left hand row.



Using the Auto Name Generator to number the pads

Remember to hit the escape key on your keyboard when you have finished numbering pin 44 to exit assignment mode.

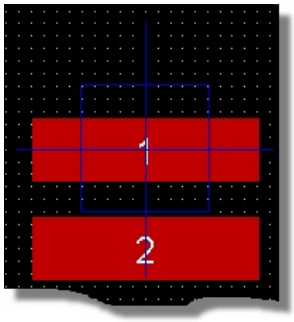
It's common to place a small dot beside pin one and you can do this via the 2D Graphics circle icon. Make sure that the Layer Selector is on Top Silk and turn the snap setting down to minimum (CTRL+F1) to achieve finer control over the size.



Use the 2D circle graphic to place a marker on pin1

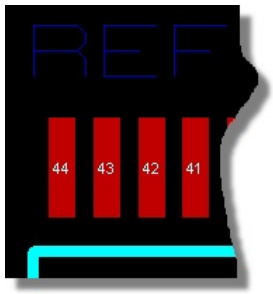
The final step is to specify where we want the origin of the device to be (for placement purposes) and where we want the silkscreen graphics for the package reference to appear. Both of these are done with markers.

Select marker mode and make sure that ORIGIN marker is highlighted in the Object Selector. You'll recall that we used this as a reference point earlier but it's real job is to specify the origin of the component for placement and rotation. This is really a decision for the user but typically the origin is either pin 1 or the centre of the component. For simplicity here, we will place the origin on pin 1; left click to begin placement, move the mouse over the centre of pin 1 and left click again to commit the origin marker.



Place a Marker on Pin1

Now, change the marker type to be REFERENCE in the Object Selector. The Reference marker dictates where the component reference (e.g. U1, R12, C3) is placed by the software relative to the component. Again, this is very subjective but we'll set it above and slightly to the left of the part.

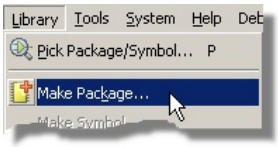


Reference Marker

Having completed the layout of the footprint, we can now move on to the (far simpler) process of packaging the part into the libraries.

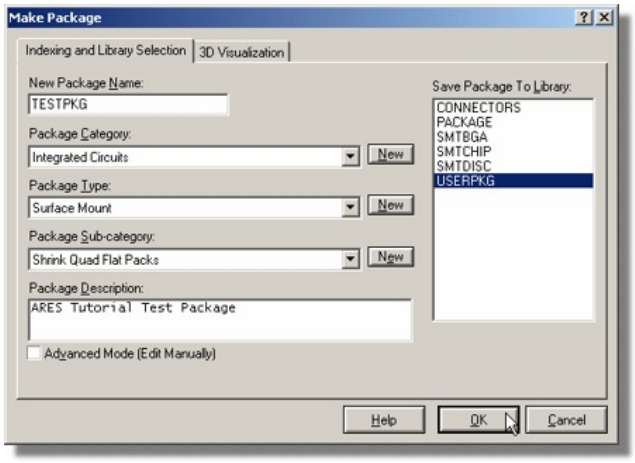
Packaging the Footprint

Drag a selection box around the entire footprint and then select the Make Package command from the Library Menu.



Invoke the Make Package tool from the Library menu

The first screen is fairly self-explanatory and similar to that we've seen in the ISIS application. Note that the package description is searchable when we are browsing for footprints so a little effort to make this as descriptive as possible is worthwhile. You will also want to create the part in the USERPKG library which is the default library supplied for user footprints. For our purposes, we'll call this package TESTPKG and give it some basic entries.



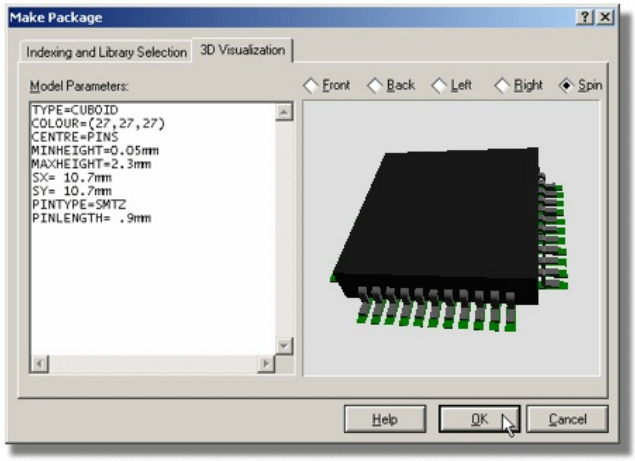
The Make Package form filled out correctly

You can create your own libraries via the Library Manager – please see the reference manual for details. You should not make your own parts into the other pre-supplied libraries as Labcenter may overwrite these libraries during upgrade installs.

When this is filled out, switch tabs to the 3D Visualisation tab (do not hit the OK button at this stage – we still have some work to do). Essentially what we need to do here is provide as much information as possible in order to get a sensible 3D image of the part which can then be used when we use the 3D Viewer to examine a board. This job is greatly aided by a 3D Preview on the dialogue form that will update live as we adjust parameters. Discussion of parameters and values is beyond the scope of this tutorial and is discussed in some depth in the online reference manual (Help Menu in ARES – Help Index). For our purposes, simply fill out the property fields as shown in the following screenshot.

3D Visualisation

When this is filled out, switch tabs to the 3D Visualisation tab. Essentially what we need to do here is provide as much information as possible in order to get a sensible 3D image of the part which can then be used when we use the 3D Viewer to examine a board. This job is greatly aided by a 3D Preview on the dialogue form that will update live as we adjust parameters. Discussion of parameters and values is beyond the scope of this tutorial and is discussed in some depth in the online reference manual (Help Menu in ARES – Help Index). For our purposes, simply fill out the property fields as shown in the following screenshot.



The 3D properties Filled out correctly

When you are finished click the OK button to commit the changes to the library.

If you now select the Package Icon, you will see that TESTPKG has appeared in the Object Selector, and can be placed like any of the packages you have used so far. Also, if you place the part on the layout and invoke the 3D Visualisation engine from the Output Menu you will see the 3D rendered image of the part.

SCREEN LAYOUT

The Menu Bar

Menu items

The Menu Bar runs across the top row of the screen and its main purpose (the selection of commands from the menus) is the same as with any other Windows application. In addition, the title bar area above the menu names is used to display certain prompt messages which indicate when the program has entered a particular editing or processing mode.

The Toolbars

As with other modern Windows applications, ARES provides access to a number of its commands and modes through the use of toolbars. The toolbars can be dragged to any of the four edges of the ARES application window.

Command Toolbars

The tools located along the top of the screen (by default) provide alternative access to the menu commands, as follows:

Command Icons

Mode Selector Toolbar

The toolbar located down the left hand edge of the screen select the editor mode, i.e. what happens when you click the mouse on the Editing Window.

Mode Icons

Orientation Toolbar

The orientation toolbar displays and controls the rotation and reflection for objects prior to placement on the layout.

Orientation icons

Toolbars visibility can be controlled from the Toolbars command on the View Menu.

The Editing Window

The Editing Window displays the part of the board that you are currently working on.

The contents of the Editing Window may be redrawn using the Redraw command which also redraws the Overview Window.

Panning

You can reposition the work area over different parts of the layout in several ways:

- By clicking left at a point on the Overview Window - this re-centres the work area about the marked point.
- By clicking the middle mouse button to ‘pick up’ the sheet, moving the mouse to re-position the sheet and then clicking the middle mouse button again to ‘drop’ the sheet.
- By moving the mouse over the Editing Window, holding down the SHIFT key, and 'bumping' the pointer against one of its edges. This pans the display in the appropriate direction.
- By using the Pan icon on the toolbar.

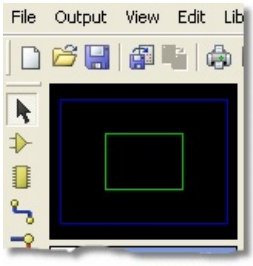
Zoom In / Zoom Out

You can magnify or reduce the display of the board using the Zoom In and and Zoom Out commands which can also be invoked via keyboard shortcuts (by default, the F6 and F7 keys). Additionally, you can use the shortcut key for Zoom All (F8 by default) to display a view of the entire board. You can also use the corresponding icons on the toolbar or simply roll the middle mouse button in or out to zoom in/out respectively. In particular with the latter, you can roll the mouse button while panning by clicking the middle mouse button to pick up the sheet, rolling the mouse to zoom out/in whilst moving the mouse to re-position the sheet. While this sounds complicated it is in fact extremely intuitive in practice.

Variable Zoom

An arbitrary degree of magnification can be achieved using the Shift-Zoom feature. A given area of the board can be selected to fill the Editing Window by holding down [the SHIFT key, pressing the left mouse button and dragging out a box around the desired area. The area can be marked on either the Editing Window or the Overview Window.

The Overview Window



The Overview Window

This window shows a simplified representation of the whole drawing, and has a 500 thou grid on it. The blue box marks the outline of the work area as set by the Board Properties command

(Technology Menu) whilst the green box indicates the region on view in the Editing Window.

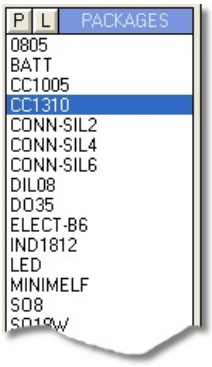
Clicking left at a point on the grid re centres the Editing Window around this point.

The Overview Window is also used to display previews of objects that are selected for placement. This features helps you to orient (rotate and mirror) an object correctly before placing it on the board.

The width and height of the Overview Window can be adjusted by dragging its borders. If you drag the vertical border right over to the other side of the application Window, ARES will re-organize the display so that the Overview Window and Object Selector are located at the right hand side.

The Overview Window pane can also be dynamically hidden by right clicking on it and selecting the 'Auto-Hide' command from the resulting context menu. This is particularly useful where the parts are on the layout and you want to maximise the Editing Window space for layout of the board.

The Object Selector



The Object Selector

The Object Selector is used for picking components, packages, pad & trace styles and so on from those that are currently available. It always carries a label indicating what it is listing and this serves as a prompt additional to the state of the Icon Panel as to which mode is current.

The width and position of the Object Selector can be adjusted in conjunction with the width and height of the Overview Window, as described above. You can also hide this pane to maximise the Editing Window space by right clicking on the Object Selector and selecting the Auto-Hide command from the resulting context menu.

The Context Menu for the Object Selector provides a number of mode specific options, giving you quick and easy access to functionality during PCB design.

The Layer Selector

The Layer Selector

The Layer Selector is used to display and select the current layer or layer set. A layer is a single layer, whereas a layer set is a combination of single layers, such as 'ALL'.

The layer selector can be operated with either the mouse or the keyboard. The default keyboard controls are as follows:

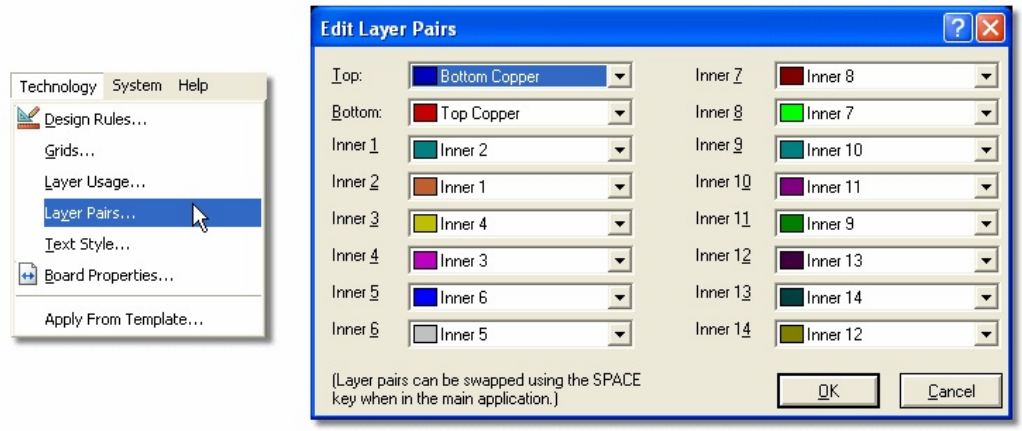
Shortcut	Description
SPACE	Selects the next layer in the current layer pair sequence.
PGDN	Selects the next layer down the list.
PGUP	Selects the previous layer up the list
CTRL-PGDN	Selects the last layer in the list.

CTRL-PGUP	Selects the first layer in the list.

All the keyboard mappings above are configurable via the Set Keyboard Mapping command on the System Menu.

Which layers are displayed in the list is determined by the current editor mode, such that PCB objects cannot be placed on inappropriate layers.

Which layers are displayed is controlled with the Layers command on the View menu whilst the layer pair sequences can be redefined using the Layer Pairs command on the Technology menu.



Technology > Layer Pairs and the Edit Layer Pairs Dialogue Form

This latter command defines the layer that the SPACE key will select for each layer. Thus you can make pairs, triplets quadruplets or whatever.

Co-ordinate Display

The current co-ordinates of the mouse pointer are displayed down at the bottom right of the screen by default. The read out can be in imperial or metric units and a false origin may be set. False origin commands (keyboard toggle 'O') and metric or imperial (keyboard toggle 'M') can be configured via the Set Keyboard Mapping command on the System Menu or activated directly from the View Menu.

The X-Cursor command will display a small or large cross at the exact location of the current co-ordinates (that is, the precise snap location for placement). Again, this can be toggled via the View Menu or by using the mapped key on the keyboard (default 'x').

CO-ORDINATE SYSTEMS

Overview

The fundamental unit of linear measurement in ARES is 10nm (ten nanometers). Given 32 bit representation of the co-ordinates this allows a board size up to +/- 10m (ten meters) with some headroom for calculations. The 10nm unit divides exactly into both 1um (micron) and 0.1 thou (one ten thousandth of an inch) giving exact representations of both imperial and metric dimensions to these resolutions.

Rotational angles are stored to 0.1degree.

The co-ordinate display can be switched to reading metric units by invoking the Metric command or via the keyboard shortcut (by default this is the 'M' key). Restoring imperial mode is achieved by invoking the Metric command again.

Dimension Entry Fields

A number of the dialogue forms contain fields that deal with dimensions such as pad style sizes, track widths, design rule clearances and so forth. All these fields are handled in a such a way that both imperial and imperial units can be used at will. The operation of these fields is in accordance with the following rules:

- Values are displayed in units chosen by a heuristic algorithm which detects whether a dimension is a whole number of imperial or metric units. Thus 25.4mm (however originally entered) will always display as 1in, when the field first appears.
- Displayed values always carry a unit. If you delete the existing value, the original displayed unit will be used.
- If you prefer, you can enter a new value with an explicit unit. Valid units are:

Units	Description
th	Thou (10e-3 inch)
in	
um	micron (10e-6 meter)
mm	millimeter (10e-3 meter)
cm	centimeter (10e-2 meter)
m	

- Whatever units are chosen, values must be less than +/- 10m, and are held at a resolution of 10nm.

Output Origin

For CADCAM output in particular, but also when working on a board which has to fit into a pre-designed mechanical casing, it is useful to be able to define a reference point on the board which relates to the mechanical design's co-ordinate system. The output origin defines this point within ARES, and is drawn in blue on the Editing Window as a target symbol.

- You can re-position the output origin using the Output Origin command.
- The output origin does not affect the co-ordinates used in Region files.

False Origin

Although the Origin command appears on the View menu, it should only be used via its keyboard short cut (default key is 'O'). Its function is to zero the co-ordinate display at the

current mouse position, a feature that is most useful when laying out a complex pattern of pads given a set of dimensions on a drawing of the component.

When a false origin is set, the co-ordinate display changes colour from green to magenta as a reminder.

False Origin set to 'ON'

Cancelling a false origin is done by invoking the Origin command a second time.

Grid Display

The Editing Window includes a grid display by default. This can be toggled between ‘off’, ‘dot display’ and ‘line display’ via the ‘G’ button on your keyboard. The spacing of the dots normally equals the current snap setting (see below), but if you are zoomed out a good way, it will be multiplied by a factor of 2,4,8 etc. The threshold at which this occurs can be adjusted on the Grids command on the Technology menu.

The grid dots on the Overview Window are 0.5in apart.

Snapping Grids

Although the motion of the mouse arrow itself is smooth, you will see that the co-ordinates normally jump in fixed multiples of, for example, 50 thou. The size of these multiples is set by the Snap commands and the purpose of this is to facilitate the placement of objects on a fixed grid.

Under normal circumstances, boards tend to be routed on a 50 or 25 thou grid.

Three snap grids for each of imperial and metric modes are available at any one time, though you can redefine their values using the Grids command on the Technology menu. The current snap options are selectable from the View menu or by using shortcut keys (the preconfigured keys for the snap commands are CTRL+F1 & F2-F4).

The snapping points are computed starting from the current origin as set by the Set Origin command. This provides the means to measure out pad placement for a new component, since you can set the origin at the first pin, set one of the snap values to the pin spacing, and then advance the cursor along the pad positions using the cursor keys.

Real Time Snap

As well as snapping to the grid dots, ARES will also snap the cursor to pads and/or tracks which lie off the currently selected snap grid. This process takes place in real time - as the pointer is moved - hence the name Real Time Snap.

The rules for what is snapped to are as follows:

- Pads are always snapped to. Only pads whose layer range overlaps the current layer range are scanned.
- Tracks are snapped to when the either the Trace icon or Via icon is selected. Only tracks on the current layer are scanned.
- Ratsnest lines are snapped to when the Ratsnest icon is selected.
- The range within which the pointer will 'see' an object is either half the current snap spacing or, if grid snap is disabled, the actual bounds of the object concerned.

RTS is extremely useful when routing between components whose pins are on different grids, since it alleviates the need to constantly switch between routing grids. In this context, the Trace Angle Lock feature is also very important.

OBJECT PLACEMENT

Overview

There are five basic types of object in ARES:

- Components
- Packages
- Pads
- Graphics
- Zones

All are placed in much the same way although there are some slight differences which are detailed in the following topics.

Placing Components

ARES makes a distinction between Components and Packages in that a component is a part specified in the netlist whereas a package is not linked to the netlist, has no part ID, and plays no part in netlist driven design verification / modification. Assuming you are using a netlist, 99% of the parts you place would usually be components. You would only use packages for parts which, for whatever reason, were not entered on the schematic. Heatsinks are the most common example of this. If a netlist is in use, parts which are not specified in it must not be given names.

To place a component

1. Select the Component icon from the Mode Selector toolbar. In this mode, the Object Selector lists all components not yet placed.
2. Choose the component you want to place from the Object Selector.
3. Set the Rotation and Mirror icons to determine the required orientation.
   You can set th[e rotation to non-orthogonal angles by typing into the text edit-box.
4. Set the Layer Selector to determine the required layer. Note that the Layer Selector and the Mirror icon are interlocked, since a reflected component must be on the underside of the board.
5. Point at the desired position for the component on the board and left click the mouse.
6. Move the mouse to the desired location – a ‘shadow’ component will follow the mouse at the stage.
   The '+' and '-' keys on the numeric keypad will rotate the component anticlockwise and clockwise respectively in steps of 90 degrees.
7. When the mouse is over the location that you wish to place the component left click again to drop the component at this point and exit placement mode.

Components on the underside of the board

When a component is placed on the solder side of the board, it is effectively 'turned over'. This means that it suffers both a reflection in 2D space and an inversion of the layers of its pads. More specifically:

- The silk screen legend moves from the Top Silk Screen to the Bottom Silk Screen.
- Pads on the Top Layer only, move to the Bottom Layer and vice versa. This caters for underside placement of surface mount components.
- Pads on multiple layers - most commonly normal through hole pads - and padstacks are left alone. On the assumption that a padstack has been defined in a particular way to reflect the usage (signal, ground plane, etc.) of a particular layer, it would be unhelpful to

invert pad stacks for solder side placements.

Placing Packages

As discussed above, the term Package is used to refer to a placed library part which has pins but is not linked to the netlist. In ARES, they are useful in the following circumstances:

- If no netlist is available, or you are doing a simple 'quick and dirty' job, you can lay out a board by placing packages and wiring them up using traces and vias. In doing this, you are using ARES as a computerised light box, rather than as a true EDA tool.
- Sometimes you need to place objects which have pads and graphics but are not actually components in the electrical sense of the word. Heatsinks with mounting holes are a common example of this. These can be placed as unlabeled packages, and as such, will play no part in the netlist driven functions of the design work.
- In order to edit a library-package, it is necessary to place it as a package, tag it and then decompose it using the Decompose commands. The individual elements can then be edited as required prior to re-making the library-package using the Make Package command.

To place a package

1. Select the Package icon from the Mode Selector toolbar. In this mode, the Object Selector lists all the packages picked from the libraries.
2. Assuming that the package you want has not yet been picked from the libraries, click the 'P' for pick toggle at the top of the Object Selector.
3. Pick the package or packages you want from the Library Pick Form.
4. When you have finished picking packages, close the form in the usual way.
5. Highlight the package you want to place in the Object Selector.
6. Proceed from step [3] in To Place a Component above.

A practical example on creating footprints can be found in the appendix to the tutorial.

Placing Pads

The main reason for placing lone pads is when defining a new library-package but they are also useful for test points, drilling alignment targets and such like. ARES offers six shapes of pad, namely: `Circle, Square, DIL, SMT, SMT Polygonal` and `Edge Connector.`

In addition, ARES supports pad stacks, in which a different pad shape can appear on each layer of the board.

The various pad shapes may be selected from the Mode Selector toolbar.

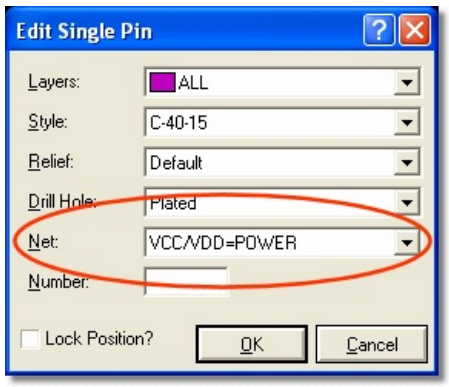
Given the selection of a pad shape, an actual pad style of that shape may be selected from the Object Selector. You'll see that all the pad styles have names this is a unique feature of ARES and has two advantages over other systems:

- There is less chance of confusion or error when choosing a style (provided that you use sensible names!).
- There is no limit to the number of styles you can have.

Pads are placed in much the same way as components and packages. The Rotation icons affect the placement of DIL, SMT and Edge Connector pads whilst the Layer Selector again determines the layer set of the pads.

You can edit a pad style by selecting it in the Object Selector and then clicking on the toggle. Any changes to the pad dimensions take effect as soon as the display is redrawn you can force a redraw with the Redraw command. This feature is particularly powerful should you find the need to tweak pad styles globally in order, for instance, to change the pad sizes used in the standard package library.

You can also place loose pads and then manually attach them to a net. This is particularly useful for RF work where you want to stitch zones together for example. After placement, simply tag and edit in the normal way and select the appropriate net from the resulting dialogue form



Assigning a Pad to a Net

Placing 2D Graphics

The 2D drawing facilities are intended, in the main, for adding text and graphics to the silk screen layers though they can also be placed on other layers by first selecting the desired layer from the Layer Selector. Objects placed on the Edge layer appear on artwork produced from any of the other layers making it a suitable location to draw the board outline.

There are 9 types of standard 2D graphic object, namely:

- Line
- Box
- Circle
- Arc
- Path
- Text
- Symbol
- Marker
- Dimension Graphic Object

Lines are drawn by clicking left at each end.

Boxes and circles are placed by left clicking once and then dragging out the desired area, left clicking a second time to commit placement.

Arcs are placed by drawing a quadrant in the same manner as for a box or circle. They can be further adjusted by tagging and dragging the four handles. ARES arcs are actually Bezier curves.

Paths are closed figures with boundaries defined as multiple line and arc segments. Click left at each required vertex and hold the CTRL key down to place arc sections. Further adjustments may be made by tagging the path and moving the green ‘handles’ about. We strongly recommend that paths are not used to create copper fills as they are not-recognized by the connectivity database. Use zones instead.

Text is placed by clicking once on the Editing Window at the point where you want the text to appear. A form then appears which allows input of the text itself and also allows control of the text dimensions. Text orientation can be set using the Rotation and Mirror Icons.

Selecting the Symbol icon displays symbols in the Object Selector. Symbol selection is identical to package selection as described earlier. More information on the symbol library facility is contained in The Symbol Library.

Markers are used in the creation of library parts to define the origin, and also the position for

component labels.

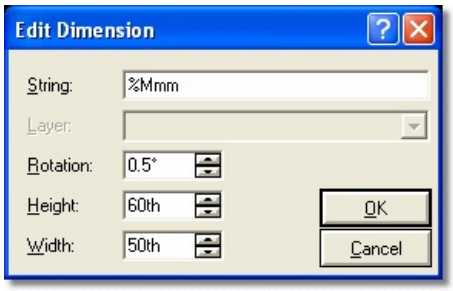
You should always use a 2D Graphics Path or 2D Graphics Box to form your board edge. It is important for the software to know the bounds of the board in order to limit copper flow, auto-routing and auto-placement.

Dimension Graphic Object

The dimension graphic object is a useful tool for sizing objects and positioning objects relative to one another. You place it exactly like placing a 2D graphics line but it will display the length of the line drawn on completion. This provides an easy and convenient way to take measurements in ARES. By editing the dimension object in the usual way (see Editing an Object on page 86) you can specify which units you want displayed according to the options given below :

Syntax	Description
%A	Automatic Mode. This is the default mode and will use the units currently specified in ARES.
%M	This syntax specifies that the units for the dimension graphic object are millimetres
%C	Specifies units of centimetres for the dimension graphic object.
%T	Specifies units of thou for the dimension graphic object
%I	Specifies units of Inches for the dimension graphic object.

Note that, with the exception of Automatic mode, you can append unit specifiers to the syntax. In the example below the units are entered as millimetres and suffixed with the standard unit specifier.



Dimension settings

Placing Zones

Zones represent the means by which ARES handles power planes. In this section we will just describe briefly how to place a simple power plane; further documentation on power planes is given under Power Planes .

To place a power plane on a layer

1. Make sure that you have specified a board edge. This is necessary to bound the area into which the zone will be drawn. To do this draw a board outline of the correct dimensions using the 2D graphics tools on the Board Edge layer. Ensure that the boundary forms a closed polygon. The best way to do this is to use a Path object.
2. Select the Power Plane Generator tool from the Tools Menu.
3. Select the net, required boundary trace style for the zone, layer for the zone and clearance from the board edge. Each of these options has context sensitive help available direct from the dialogue form.
4. Hit the OK button to render the zone with the specified options.

To place a polygonal or partial power plane (PCB L2 or higher required).

1. Select the Zone icon from the Mode Selector toolbar..
2. Select the required boundary trace style for the zone from the Object Selector.
3. To define the zone boundary, either:
   - Drag out a box with the left mouse button held down.
   - Click at several points to form a polygonal boundary.

   If a zone boundary is placed over existing tracking, it will pick up its net from that tracking. This makes it quite easy to place lots of small zones in order to ‘fill in’ between tracks.
4. Select the required net and hatch/fill style from the Edit Zone dialogue form, then click OK.
5. ARES will generate the ground plane

Object Editing

Overview

Due to is its object oriented design, all objects in ARES are treated in much the same way and we can discuss generally their dragging, copying, movement, rotation, deletion and editing. You should note that traces and vias are not regarded as o[bjects for this purpose; their editing is dealt with further under Manual Routing. However, there are operations where traces attached to objects are affected by operations performed on the object - such cases are discussed here.

The Selection Filter

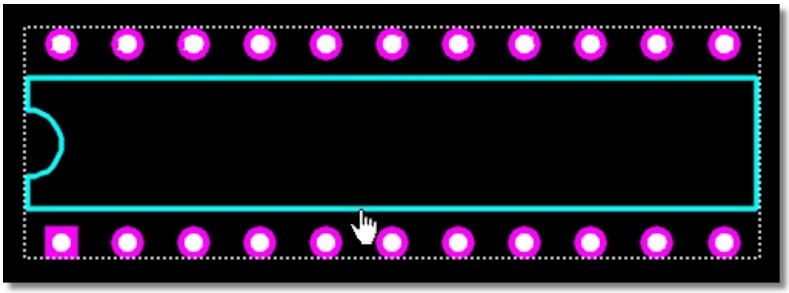
Board design by it's very nature consists of several layers (silkscreen, copper layers, edge layer, mechanical layers etc.). Despite this, ARES provides a modeless selection mechanism where the object under the mouse is identified regardless of whether it is on the current layer. This works extremely smoothly in most cases but the selection filter provides the means by which y[ou can override the default behaviour in a given situation.

The selection filter can be found at the bottom left of the ARES application window

Selection Filter

The purpose of the selection filter is simply to determine which object types are considered 'hot' in any given mode. Additionally, the icon directly to the right of the layer selector determines whether all layers or only the current layer are considered to be 'hot'. By toggling these icons you can therefore have complete control over both which objects and which layers are hot at any given time.

When the mouse is over a 'hot' object a visual cue will be given to light dashed line will appear to provide instant feedback telling you what exactly the mouse is considered to be pointing at.



When the mouse is over a 'hot' object it becomes highlighted

Sophisticated algorithms are in place to cater for the cases where, for example, two tracks on different layers are placed in parallel. Whilst the obvious solution is to look for a point on the desired track which is visible on it's own, the behaviour in such a case is prejudiced by the layer currently selected in the Layer Selector. You could therefore switch the layer selector onto the layer of the trace you want to select and then click on the track.

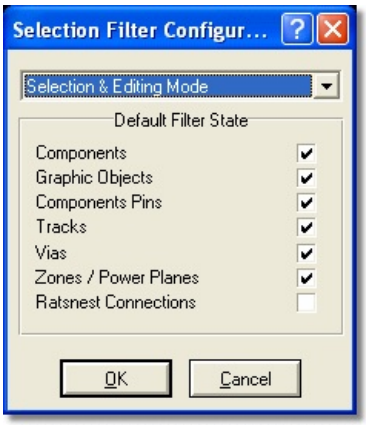
It is **extremely** important to realise that this object 'twitching' indicates only the object that will be acted upon when you click the mouse. It does not indicate what the action will be (selection, placement, etc.) – indication of action is provided by the mouse cursor and is summarised below :

Cursor	Description
	Standard Cursor

	Placement Cursor - Left click will place an object according to the current mode
	Selection Cursor - Left click will select the current object under the mouse
	Movement Cursor - Left click will move the current object(s) selected

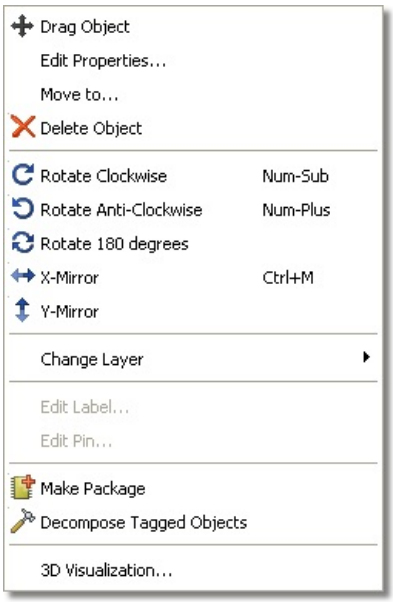
Using these two visual indicators together provides an extremely intuitive method for both knowing what object the mouse is over and also what action will be performed when you left click the mouse on that object.

Finally, while considerable time has been given to choosing a sensible default selection set for each mode in ARES you can override it if you decide that you prefer to operate in a different way. Default selection filters on a per mode basis can be configured via the Set Selection Filter command on the System menu.



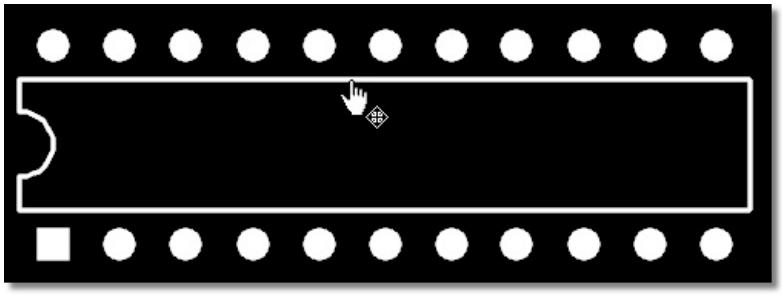
Tagging a Single Object

Provided the object is considered ‘hot’ (as defined via the selection filter)any object may be tagged by pointing at it and clicking right. This action highlights the object and presents a context menu containing relevant actions for that object.



Right click to reveal the Context menu

Alternatively, where an object is ‘hot’ and the mouse cursor indicates selection (pointed hand) a left click on the object will select the object



A useful trick for precision movement of a part is to map the 'Drag Object' command on to a keyboard shortcut. You can then tag the part, invoke the keyboard shortcut and use the keyboard arrow keys to move the part by a single grid setting.

Tagging a Group of Objects

A group of tagged objects may be assembled by by dragging a box around them using the right mouse button. Once the tag box is created, the enclosed space can be refined by dragging the green 'handles' as required. The 'trace bounds' selection icon can be used to determine whether only fully enclosed traces or partially enclosed traces are tagged.

After placement of the tagbox the selection filter can be used to refine which objects are tagged, providing a live update according to the current configuration – see Block Editing Facilities for more information.

Only one tag box can exist at any one time - drawing a new one will delete the old one, and untag the objects within it.

Untagging All Objects

To untag all the objects, you need simply to point where there is no object and click left. This will also remove the tag box if one has been defined.

Performing an explicit Redraw command (default key mapping 'R') will also untag all objects this can be handy if the board is very densely packed and there is nowhere free of objects on the screen.

Deleting an Object

Any tagged object can be deleted by pointing at it, clicking right, selecting Delete Object from the resulting context menu. Any traces connected to its pads will remain in place, allowing a different pad or package to be placed over them.

Dragging an Object

You can drag (i.e. re position) any tagged object by pointing at it and then dragging the mouse with the left button depressed.

Any traces attached to the object will be rubber banded (stretched) unless both ends connect to it, in which case the route will be moved wholesale.

Editing an Object

Some objects, especially packages, components and graphics text have properties that can be edited with a dialogue form. There are two ways to do this:

- A tagged object can be edited by pointing at it and then clicking left as if to drag it but without moving it.
- You can right click on an object and select Edit Properties from the resulting dialogue form.

Locking an Object

You can lock components, packages or single pads on the layout such that they cannot be moved, rotated or deleted. This is useful if, for example, you have connectors or mounting holes that need to be in a specific position on the board and you wish to ensure that they are not inadvertently moved.

To lock an object on the PCB

1. Place the object on the layout in the specified position.
2. Edit the object by right clicking to highlight and then selecting Edit Properties from the context menu to launch the resulting dialogue form. Note that for components you must click on the component outline – if you click on a component pad you will get the Edit Pad dialogue form instead.
3. Check the box at the bottom of the dialogue as shown below to lock the object. Note that only the object types specified above can be locked

The Lock Position check box

While no visible indication of a locked object is provided on the layout (to avoid clutter) you can find all the locked objects on the board as follows :

To find all locked objects on the PCB

1. Tag the entire board (right click and drag a box) and then attempt a block operation.
2. From the resulting dialogue box select the option to tag only the locked objects on the board.

Highlighting a Component by Name

When the Instant Edit icon is selected, the Object Selector lists all the components in the design. Clicking on the selector toggle tags the selected component causing it to be highlighted, and zooms in to display it in the centre of the screen.

Alternatively, you can locate a component using the Goto Component command on the Display menu.

ROUTE PLACEMENT & EDITING

Overview

ARES provides a variety of methods for the manual placement and editing of tracking and vias. In particular, existing routes may be modified by any of the following methods:

- Over placement - place a new section of route over an old one and the new will replace the old. This provides an easy way to change the thickness of sections of tracking.
- Segment dragging - the path of a tagged route can be easily changed by dragging its segments. Neighbouring segments are automatically stretched so as to maintain an orthogonal path. This is a very intuitive method of working for users accustomed to other Windows drawing packages.

In all cases, ARES will validate newly placed or modified route sections against both the connectivity data specified in the netlist and the physical design rules for the layout. Sections of tracking which violate the connectivity rules are marked as 'dirty' and will flash in yellow to indicate that they are in illegal positions whilst tracks in violation of design rules are logged and navigable to via the Design Rule Checker dialogue form at the bottom right of the ARES application.

Trace Placement - No Netlist

⚠ We do not recommend professional board layout without the benefit of a netlist as it renders ARES blind to connectivity and severely limits functionality such as power planes.

Placing a new trace on the primary layer is done by first selecting the Trace icon and the required trace style from the Object Selector and then left click to start placing a trace. Placement will follow the path of the mouse while obeying the default board constraints as defined in Design Rule Manager. Left clicking at consecutive nodes (change of direction) will commit the route placement to that point and right click will finish track placement.

Double clicking the mouse will place a via if it is legal to do so and routing can then continue on the associated layer as defined by the Layer Pairs command. Hitting SPACEBAR will float a via on the end of the mouse and a single left click will then deposit the via and switch layers. Floating vias have the advantage of snapping to valid destinations (such as via under SMT).

The type of via used can be changed by selecting the Via icon and choosing one of the via styles from the Object Selector. You can also place, replace, tag, drag and delete vias manually in this mode. The mouse buttons work in the same was as for ordinary objects.

You can change layers whilst routing using the Layer Selector keyboard controls. The default controls for these commands are PGUP, PGDN, CTRL-PGUP, CTRL-PGDN – you can configure your own shortcuts via the Set Keyboard Mapping dialogue on the System menu.

Should you start to enter a trace, and then decide you wish to abort it entirely, simply press the ESC key.

Trace Placement - Netlist Loaded

If a netlist is loaded and you commence routing from a pin assigned to a given net, ARES will:

- Indicate the net name of the given net on the status bar.
- Indicate to which other pins the first one connects, dynamically adjusting the netlist to indicate the closest connection point and provide a ‘shadow’ trace to show you the layer

on which you are currently routing.
- Select the appropriate trace and via styles for the given net according to the net class to which it is assigned. This function can be disabled by toggling the Auto Trace Selection option on the Tools menu. This can be useful if the odd part of a net needs to be a different thickness. See Routing Net Classes for more information on net classes.

You can then place route segments and vias as described in the previous section. Assuming that you take the route to one of the indicated pins, ARES will automatically terminate routing mode when you click left on it.

Should you wish to connect to a point other than a selected pin, this is possible by clicking right to terminate routing as described in Trace Placement No Netlist Loaded. ARES will still detect any ratsnest lines that have been connected and will remove them from the display though this takes a little longer than if you route directly from pin to pin. In fact, ARES will remove connected ratsnest lines even if neither end of the placed route starts on a pin - Advanced Netlist Management can check a whole net's connection status very rapidly.

During placement ARES will restrict the track being placed according to the design rules for the layout. This prevents any major blunders such as connecting to another net and ensures correct clearances between objects on the board. It also enables you to wrap traces around obstacles or to hug other parallel traces on the layout.

The exception to this is that tracks placed in the BRIDGE trace style are completely ignored by the ARES connectivity and design rule checking systems. This is deliberate as this style is specifically intended for joining distinct nets (e.g. star points) but it does mean that considerable care is required when routing in the BRIDGE style.

When Auto Track Necking is enabled, ARES also checks for physical design rule violations. In the first instance, it will neck the track to avoid them, but if there is still a violation then it displays a warning message and beeps. The route is, however, placed. This process can take some time, and if you find that trace placement is becoming unduly tardy, you can try disabling ATN from the Tools menu.

Curved Track Segments

ARES fully supports curved tracks and supports them as true arcs for output purposes. The segments also play a full role in connectivity and design rule checking. To enable curved tracks you must first turn off Follow Me Routing from the Tools Menu.

To place a curved segment, first commence routing as usual (e.g. click left on a pad) but then, before moving the mouse, hold the CTRL key down. Move the mouse roughly to trace the arc, click left to fix the arc endpoint and then release the CTRL key. You can place a curved segment at any stage of laying a route - it does not have to be the first or last segment.

Two important points about curved tracks:

- Because ARES renders a curved track as an arc, the mechanism that 'backs off' a trace end from any pad to which it connects does not operate for curved track segments. It follows that if curved segment connects directly to a pad, the centre hole of the pad may be obscured.
- It is possible, using the route editing facilities described in the following sections, to break a curved segment into sub-sections. ARES will allow this but the sub-sections will then be treated as runs of short linear segments. As such, they may not render satisfactorily on pen plotters

Auto-Track Necking

In many cases, the reason for necking down a track is so that it can pass between two pads or other obstacles without violating the design rules. The Auto Track Necking feature allows ARES to do this for you.

In order to auto-neck during routing you need to pass through the gap in a straight line (i.e. not at an angle). The track neck will appear once you are more than half way through the gap.

The neck thickness is controlled by the Design Rule Manager command on the Technology menu. The dialogue form allows you to enter the clearances for pad-pad, pad-trace and trace-trace and also the trace style to neck to. The default neck style is T10 - a 10 thou track.

To disable ATN totally, use the toggle command on the Tools menu. You may find this speeds up route placement considerably, since the analysis involved in performing ATN can be quite complex, even if the end result is that the route is OK.

Trace Angle Lock

Many people like to keep all tracking running at either 90 or 45 degrees as this gives a tidy, professional look to a layout. To make this easier, ARES has a Trace Angle Lock command which restricts route segments to 90 or 45 degree angles only.

TAL also has a more subtle effect when combined with Real Time Snap. Consider the following routing problem... You are trying to route from a pad on the grid to one that is off it. You have moved down, clicked left and moved the pointer over to the destination pad. Real Time Snap has detected the pad, and the cursor has locked onto it. However, because your first click left was on a grid dot, the horizontal part of the track doesn't line up with the pad centre. The diagram below shows the position at this stage:

TAL set to 'Off'

When Trace Angle Lock is enabled, clicking over a pad or track in this circumstance will activate Trace Angle Fixup. The result is that the previous node - in this case the route corner, is moved so that trace angle lock rules can be maintained. The result is a tidy connection as shown below.

TAL set to 'On'

Trace Angle Lock & Fixup are active by default. You can disable or re-enable them using the Trace Angle Lock command on the Tools menu.

Tagging a Route

To re route, delete or copy a section of tracking requires that you first tag the route containing it.

To tag a route

1. Ensure that the trace selection icon on the Selection Filter is enabled.
2. Point at an exposed part of the route on the selected layer, and click right. This will both select the route and present a context menu of appropriate actions that can be performed on the route.

Provided that you are not in trace mode (or put another way, that the mouse cursor is a selection cursor – pointed hand – when over the trace) you can simply left click to tag the route.

To tag a section of a route

1. Ensure that the trace selection icon on the Selection Filter is enabled.
2. Point at an exposed part of the route on the selected layer, and click right. This will both select the route and present a context menu of appropriate actions that can be performed on the route.
3. Select one of the Trim options from the bottom of the context menu.

If you select Trim Manually then the point at which the mouse was clicked becomes an anchor point and you can move the mouse to the other anchor point, left clicking to select the section of trace between the endpoints.

Right click and select 'Trim Manually'

Please note that you cannot select sub-sections of tracking that run through pads, vias or T-junctions using this method.

Changing the Width of a Route

ARES provides two methods of changing track width. The first relies simply on the general ability of ARES to cope with new objects being placed over old.

To change the width of tracking by overplacement

1. Ensure that there is no tagged route, and that Auto Trace Selection is disabled on the Tools menu.
2. Select the new trace style in the Object Selector.
3. Place the new tracking directly over the old. ARES will work out that there is overplacement, and will replace the old tracking with the new in the database.

The second follows the principle of allowing various editing operations to be performed on the currently tagged object.

To change the width of a tagged section of tracking

1. Tag the required section of tracking as described under Tagging a Route. The Object Selector will display the current trace width. If the tagged route has several widths, then the width at the point under the cursor is displayed.
2. Click right on the tracking to display the context menu.
3. Select a new trace style from the menu.

Changing the Layer of a Route

To change the layer of a tagged section of tracking

1. Tag the required section of tracking as described under Tagging a Route.
2. Click right on the tracking to display the context menu.
3. Select a new layer for the tracking from the menu.

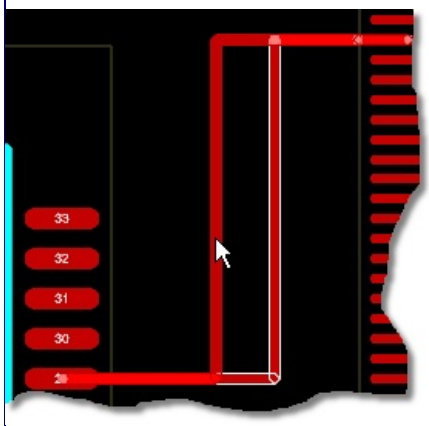
Modifying a Route

To modify an existing track

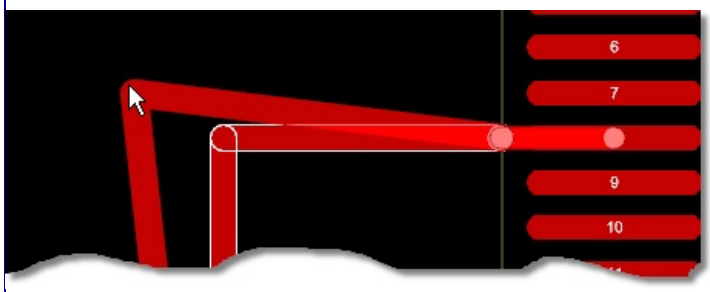
1. Tag the route as described under Tagging a Route.
2. Drag the segments and/or corners of the route using the left mouse button.

The rules of track manipulation are detailed below:

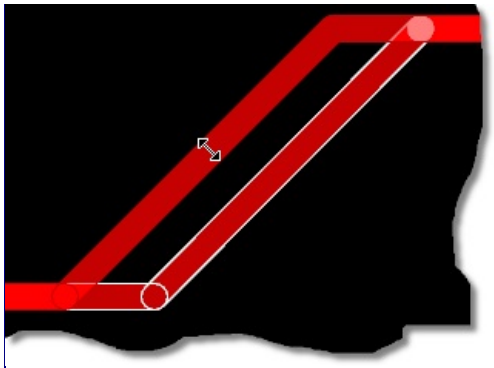
If you drag horizontal or vertical segments, they will move horizontally or vertically respectively preserving the angle between the adjacent segments.



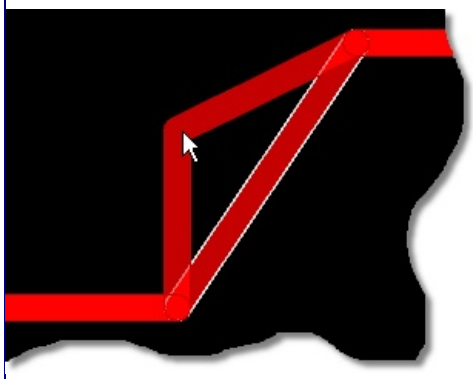
If you drag on a corner of a route, then the adjacent segments will simply stretch diagonally.



If you drag in the middle of diagonal segments with trace angle lock off then the diagonal segment will move such that the angle between the tracks at both ends of the segment is preserved.



If you drag in the middle of diagonal segments with trace angle lock on then the behaviour depends on the angle of the diagonal. If the angle is a multiple of 45 degrees then dragging will be as when trace angle lock is off (see above). However if the angle is not a multiple of 45 degrees then a new corner will be created. This is designed so that, in the common case of adjusting a miter preserves the angle between the diagonals.



While these rules may seem a little obscure in textual format a lot of thought has gone into their design and you will quickly find them both easy and intuitive to use.

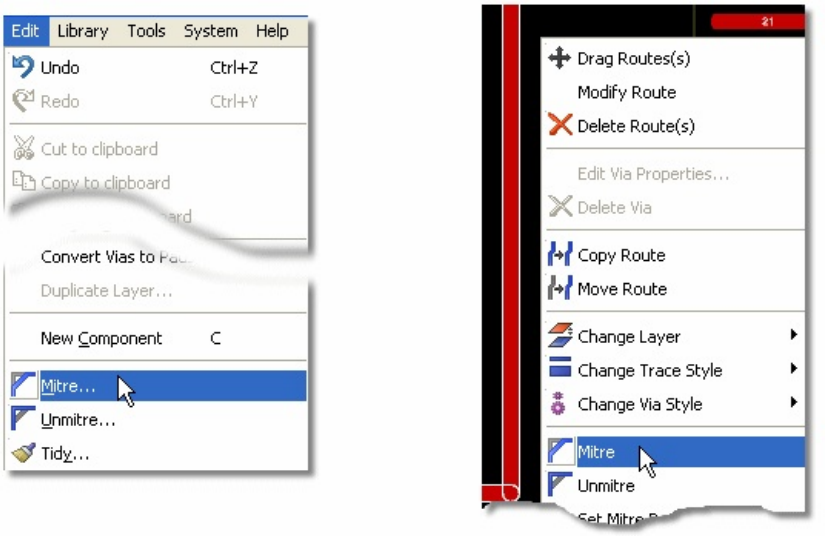
To re-route a section of tracking

1. Tag the route as described under Tagging a Route .
2. Place a new section of tracking starting and finishing on the tagged route. ARES will then remove the section of tracking that the new route shorts out.

Note that vias which are made redundant (i.e. secondary vias) will also be removed, along with the shorted out section.

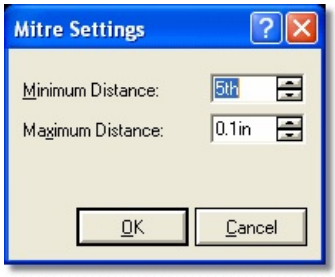
Mitring a Route

ARES will allow you to mitre the tracks on the board either as a global command on all the tracks on the board (via the Edit menu) or on a per track basis (by right clicking on the track). This is typically at it's most useful as a post routing step to minimise track length or to avoid solder traps.



Mitre traces from the Edit menu or by right clicking on the track

Configuration of the mitre depth takes place either via the Set Mitre Depth command on the context menu when mitring a single track or directly via the mitre command when mitring the entire board. The dialogue form is very simple and is shown below



Mitre settings dialogue form

The settings on this dialogue determine the minimum and maximum distance that will be used for determining whether or not to mitre a track. For most purposes the default settings are perfectly adequate.

⚠ Note that running the unmitre command after a mitre does not guarantee that the board will be returned to the condition it was in prior to mitring. For example, you may have manually mitred tracks on the board which will be unmitred by a global unmitre. If you want to undo a mitre operation directly you should use the undo command (CTRL+Z on the keyboard by default).

Copying a Route

To duplicate a route

1. Tag the route as described under Tagging a Route .
2. Click right on the tracking to display the context menu.
3. Select the Copy command.
4. Click left at each position for the duplicate routes; click right to finish.

This feature provides an excellent way do memory buses and other repeated patterns of tracking

Deleting a Route

To delete a route

1. Tag the route as described under Tagging a Route .

2. Click right to display the context menu.
3. Select the Delete command.

Any interconnections specified in the netlist that were formed by the deleted tracking will automatically appear as ratsnest lines.

Note that you can also delete all tagged tracking by using the Block Delete icon. This can be quicker than the above procedure if no other tracking is tagged.

To delete a section of a route

1. Right depress the mouse at the start of the section of route you want to delete.
2. Drag the mouse to the end of the section of track you want to delete and release the right mouse button.
3. The section will now be highlighted – proceed as outlined above to delete.

Tidying the Routes

Extensive route editing can lead to the creation of unwanted secondary nodes between tracks segments that join at 180 degrees. These nodes waste memory and also degrade the quality of plotted output. One of the processes carried out by the Tidy command is to scan for and remove such nodes.

The Tidy command may be found on the Edit menu.

BLOCK EDITING

Overview

Four block operations are provided: Copy, Move, Rotate and Delete and each operates on the currently tagged objects and the traces & vias within the tag box. If there is no tag box, then a default one is assumed comprising the total area occupied by the currently tagged objects. If there are no tagged objects, nothing at all will happen.

Block Copy

After creating a tag box as described in Tagging a Group of Objects, right clicking inside the tagbox and selecting the Block Copy command will cause a second box to appear over the tag box which you can then re position using the mouse. Clicking left will cause a copy to be made of the objects within the tag box whilst clicking right will abort the operation. Notably, the point at which you right click inside the tagbox defines the pickup point for the copy providing you with full control over the operation.

Control over what is tagged is handled via the selection filter. In particular, you can determine whether partial traces or only traces for which both ends of the trace are included in the tagbox are copied.

Beware that copying components will generally leave the netlist management out of kilter as there will be multiple placements of the same component references. If, having routed a module that is to be repeated several times, your intention is to copy it out to save re-routing, you should use the Auto Name Generator to renumber the components in the copied section(s) and the re-load the netlist. Alternatively, together with sub-circuits in ISIS (set a base annotation on the sheet) you can use the replicate command to offset the annotation to match a particular sub-circuit.

Block Move

After creating a tag box as described in Tagging a Group of Objects, left depressing the mouse inside the tagbox and dragging will pick up the block and move it. Again, the pickup point is the point inside the tagbox at which you depress the left mouse button. The mouse cursor will change to be the movement cursor to provide a visual indication when you can perform this operation.

Trace segments with one end only inside the tag box will be rubber banded.

Alternatively, you can right click inside the selected box and use the 'Move-To' command on the resulting context menu to programmatically position the block.

If you have one or more locked components/pads inside the tagbox you will be presented with a dialogue form allowing you to either select everything apart from the locked objects or to leave only the locked objects selected. The former will then allow you to move everything apart from locked objects while the latter will identify locked objects which you can then edit to unlock.

Block Rotate

After creating a tag box as described in Tagging a Group of Objects, right clicking inside the tagbox and selecting the Block Rotate command presents you with the Rotation dialogue form allowing you to specify an any-angle rotation on the selected block.

Control over what is tagged is handled via the selection filter. In particular, you can determine whether partial traces or only traces for which both ends of the trace are included in the tagbox are copied.

- Trace segments with one end only inside the tag box will be rubber banded although this generally leaves something of a mess it's best to rotate only completely enclosed sections of a layout.
- Rotation of whole groups of components to non-orthogonal angles (even 45 degrees) creates large numbers of off grid pads. Routing such boards can be very tricky, even with features such as real time snap and trace angle fixup.

The autorouter will also perform poorly under such circumstances.

Block Delete

After creating a tag box as described in Tagging a Group of Objects, clicking on the delete button on the keyboard (or the Block Delete icon on the resulting context menu) will delete all tagged objects. Use the selection filter to control the object types and layers affected by the delete operation.

Remember that you can control which objects are selected in the tagbox by adjusting the selection filter, either before or after the tagbox has been created

FILING COMMANDS

Overview

ARES makes use of the following file types:

- Layout Files (.LYT)
- Backup Files (.LBK)
- Region Files (.RGN)
- Library Files (.LIB)
- Netlist Files (.SDF)

Layout files contain all the information about one board, and have the file extension 'LYT'. They contain within them copies of all packages and styles used on the board, so the complete design can be given to someone else solely by giving them a copy of the layout file. Backup copies of layout files made when saving over an existing file are given the extension "LBK".

A region of a board can be exported to a region file and subsequently read into another layout. Region files have the extension 'RGN' and are read and written by the Import and Export commands on the File menu. They are analogous to section files in ISIS. ARES region files are in an ASCII format and for the advanced user, this opens the possibility of manually editing the layout database, or else writing utility software to perform specialist operations on it. Import of data from other PCB design packages is also a possibility. Contact our technical support department or look on our Web Site if you want a copy of the format specification.

Package and symbol libraries have the extension 'LIB'.

Starting a New Layout

To start over on an empty work area, use the New Layout command on the File menu. The layout filestem is set to 'UNTITLED' until such time as you save the layout.

Loading a Layout

A layout may be loaded in various ways:

- From the command line as in : `'ARES my_board'`
- By using the Load Layout option once ARES is running.
- By double clicking the file in Windows Explorer.

Saving a Layout

You can save a board when quitting ARES via the Exit option, or at any other time, using the Save Layout command.

- In both cases it is saved to the same file from which it was loaded; the old file being re named. Under Windows 3.1 it is given the extension ‘LBK’; under Windows 95 or NT is given the suffix ‘Backup of…’.
- If no filename was given at load time or the New command was issued, the name 'UNTITLED.LYT' is used.

The Save As command allows you to save the layout to a file with a different name.

If you save the layout with a different name from the schematic you need to change the 'Default Layout' setting in ISIS to specify that the netlist is associated with the correct layout.

Import/Export

The Export Region command creates a region file out of all currently tagged objects. This file

can then be read into another layout using the Import Region command. After you have chosen the region file, operation is identical to the block copy function.

A common difficulty arises from exporting a region file from a design where you have made local changes to pad/trace styles. Region files do not save this information and so when you import the region file all pad/trace styles will revert back to their default values. The solution of course is to create new styles in the first instance reflecting the modifications that you want to make. You can then safely create your board and export/import it via region files into other designs.

⚠ Region files are not a good way to panelize multiple boards for manufacture because netlist information is lost and imported ground planes will lose connectivity as a result. The Gerber Viewer provides a much better way to perform panelization.

Auto-Save

ARES has an auto-save facility whereby your work will be backed automatically at regular intervals. The interval defaults to 15 minutes and can be changed using the Set Environment command on the System menu.

Should ARES terminate unexpectedly, the next time it starts it will look for the auto-save file from the previous session and prompt to reload it.

The auto-save file will be stored in your Windows temporary directory, normally addressed by the environment variable TEMP. PROTEUS auto-save files have the extension 'ASV'.

⚠ This feature is not an excuse for not saving and backing up regularly - there is no guarantee that the auto-save file will be recovered in all cases.

Backups and Last Loaded Files

In addition to the auto-save ARES provides extra security for your layouts via a Backup and Last Loaded mechanisms.

Whenever a layout is saved any existing layout file on the disk is backed up to either a loas-loaded or back-up file. For a layout with the file name MYFILE.LYT the last-loaded file has a file name of "Last Loaded MYFILE.LBK" and the back-up a file name of "Backup Of MYFILE.LBK". Note that a different file extension (LBK) is for back-up and last-loaded files so they don't clutter the Load Layout dialogue form. To see these files when loading a layout you must change the Files Of Type field (at the bottom of the Load Layout file selector dialogue form) to "Backup Layout Files".

For a new layout, the back-up process works as follows. The first save operation saves the layout file. For each subsequent save, any existing back-up file is deleted, the existing layout file is renamed to the back-up file name and the layout is then saved.

For an existing layout, the back-up process is slightly different. The first save operation after loading the layout deletes any last-loaded file and renames the current layout file as the last-loaded file. For each subsequent save (during the editing session) the normal back-up scheme applies – that is, for each save, any existing back-up file is deleted, the existing layout file is renamed as the back-up file and the layout is then saved.

What this scheme implies is that the layout file (MYFILE.LYT) is always the last saved version of the layout and the back-up file (Backup of MYFILE.LBK) is usually the version of the file prior to the last save. The last-loaded file (Last Loaded MYFILE.LBK) is a version of the layout that is known to load successfully and represents the version of a layout at the start of an editing session regardless of how many times you save the layout during that editing session.

PAD,TRACE & GRAPHICS STYLES

Overview

ARES features a sophisticated and flexible system for defining shapes and sizes for pads, tracks and 2D Graphics.

Each pad and track is given a unique style name, and this name is then used to refer to the style whenever it is used. The dimensions and other characteristics for each style are held in layout-global table, and this makes it possible to change the dimensions of all the pads or tracks in a particular style extremely easily. At the same time, the use of names means that there is no serious limit on the number of pad and trace types that you can have per layout.

Individual 2D Graphics can be customised to specific dimensions, both in terms of line width and, where appropriate, fill style. This allows, for example, thicker line styles for 2D Graphics text than for mechanical outlines for PCB footprints.

Pad Styles

ARES supports 7 types of pad, namely:

- Circular PTH
- Square PTH
- DIL PTH
- Circular SMT
- Rectangular SMT
- Polygonal SMT
- Edge Connector

and the list of styles defined for each type displays in the Object Selector whenever the appropriate pad icon is highlighted.

When one of these lists is displayed, you can edit an existing pad style by highlighting it in the selector and then clicking on the selector toggle (the 'E' symbol). All fields on the resulting dialogue form have context sensitive help associated with them.

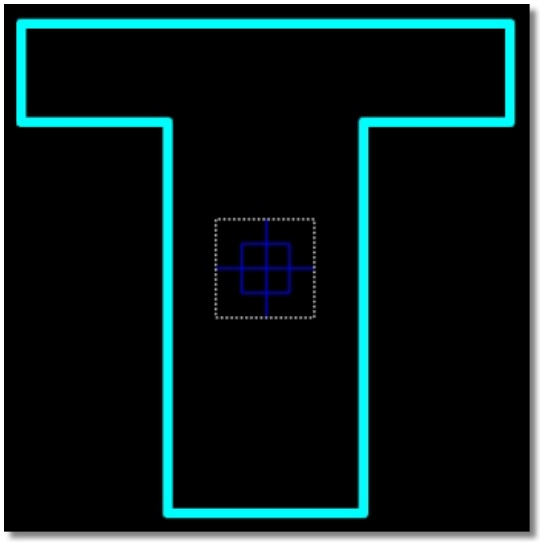
Polygonal Pads

ARES supports a user defined, polygonal pad style in order to cater for parts requiring unusual pad shapes.

To define a new polygonal pad style

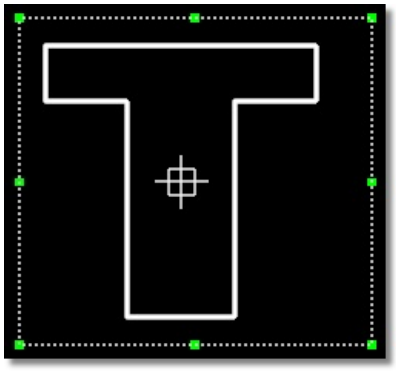
1. Using the 2D graphics Line, Arc or Path icons, draw a closed path to define the shape of the new pad style. It does not matter which layer you draw on for this purpose.

2. Place an Origin marker within the closed path to specify the origin of the pad style. The origin is used as the connection point for tracking, and also serves as the point of alignment if the pad style is used within a pad stack.

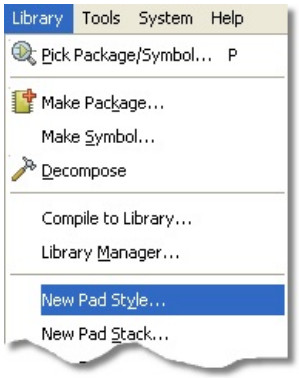


Note that the origin must lie within the closed boundary i.e. within the copper area of the pad.

3. Tag the 2D graphics and origin marker by drawing a tag-box with the right mouse button.



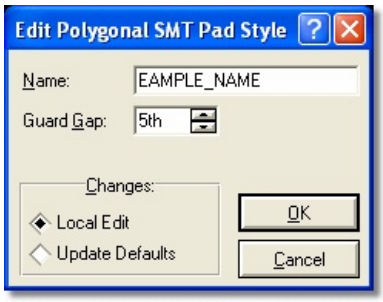
4. Invoke the New Pad Style command from the Library menu.



5. Enter a name for the new pad style.

6. Set the Polygonal pad style type under SMT.

7. Click OK.
8. Specify a Guard Gap for the pad style, if required.



9. If you wish to create the style only in the current layout, and not in DEFAULT.STY, then select Local Edit instead of Update Defaults.
10. Click OK to create the ne[w pad style.

Pad Stacks

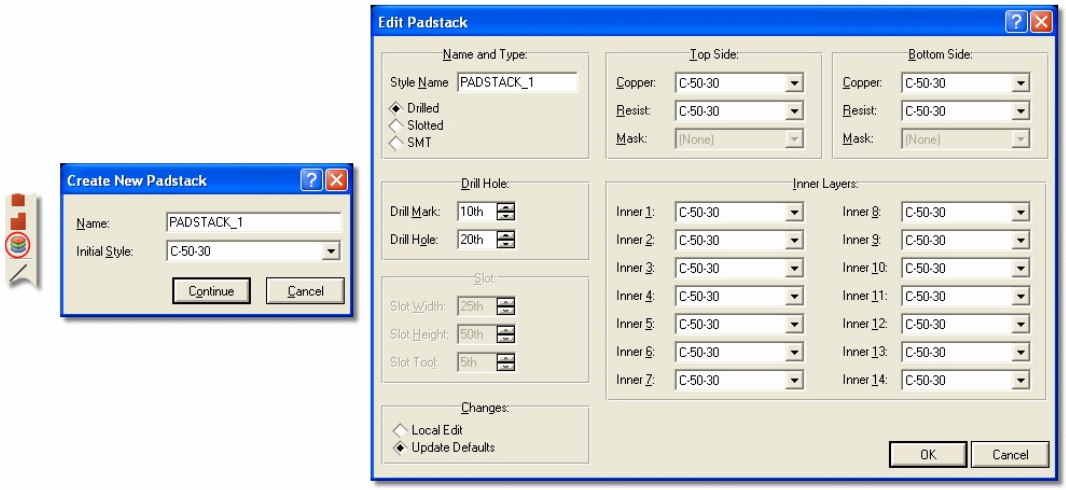
The ordinary through hole pad styles can be placed on a single layer or on all the copper layers, but this does not provide a very convenient or satisfactory method for defining component pins which have different pad shapes on different layers. In particular, the ARES connectivity system does not recognize connectivity between single layer pads on different layers.

Instead, ARES provides the facility to define Pad-Stacks. There are several points about a pad stack:

- A pad stack can have a circular hole, a rectangular slot or be surface only. The latter type of pad stack is used where you wish to defined explicit styles for the solder resist and/or solder paste mask apertures.
- For each layer, you can assign a different pad style, or no pad.
- For obvious reasons the hole or slot diameter of a pad stack is the same for all layers.
- You must use a pad stack to create a pad shape with a slotted hole. You cannot specify a slotted hole in an ordinary pad style.

To define a new pad stack

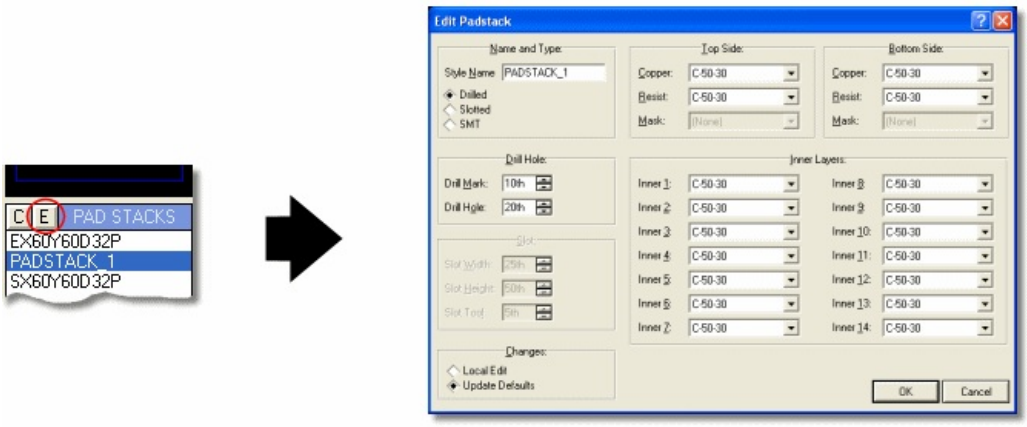
1. Select the Create New Pad Stack command from the Edit menu.



2. Enter a name for the pad stack, and a default pad style. All layers will start out with this style of pad.
3. Click Continue to proceed to the Edit Pad Stack dialogue.
4. On this form, you can adjust the pad-layer assignments, and also assign the layer-global drill mark and drill hole/slot size.

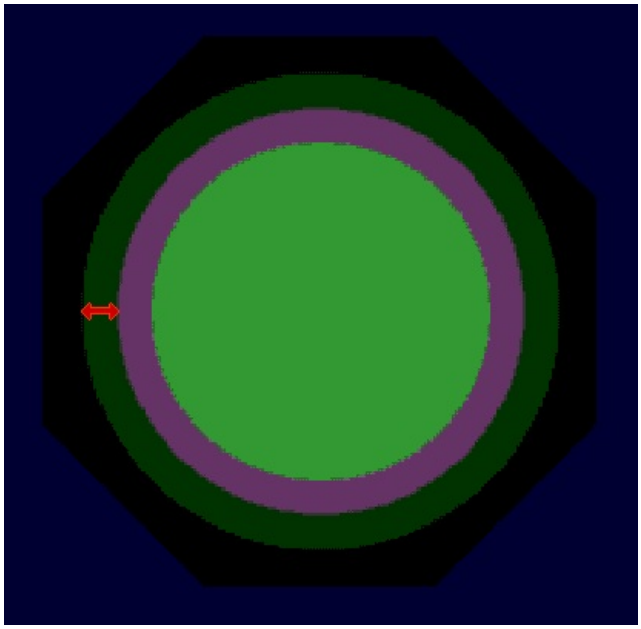
To edit an existing pad stack

1. Select the Pad Stack icon.
2. Select the pad stack you want to edit from the Object Selector.
3. Click the 'E' for edit toggle on the Object Selector to bring up the Edit Pad Stack dialogue form..



Pad Styles and the Solder Resist Layer

The solder resist artworks are generated automatically from the pad shapes present on the top and bottom of the board. Each pads' radius is enlarged by the amount specified in the Guard Gap field of its style, as shown below:



Note that in the case of a pad stack, you can define the pad style used for the top and bottom resist layers explicitly, using the selectors in the Edit Pad Stack dialogue form. Note that the guard gap of this style is still applied in determining the final size of the aperture.

You can also specify a pad stack which has no solder resist - i.e. the pad will be covered by the resist coating.

If you are working in OpenGL (hardware accelerated) mode you can turn on live display of the resist and paste mask layers to inspect coverage using the Displayed Layers dialogue form. Bear in mind that resist and paste are inverse plots so 'no-coverage' on the layout equates to flood over physically, 10th expansion on the layout equates to 10th gap on the physical board, etc.

Pad styles can be edited once created by selecting the relevant pad style icon and then the 'E' button above the object selector/parts bin. You can then change the resist for that pad style (guard gap field) and limit the change to the specific layout you are working on by selecting the 'Local Changes' option at the bottom of the dialogue form. This is commonly done when you wish full resist coverage over via's on the layout.

Pad Styles and the Solder Paste Mask

The solder paste mask artworks are also generated automatically from the SMT pads present on the board. The paste mask aperture is exactly the same as that used for the pad itself. Once again, in the case of pad stacks, different pad styles may be specified explicitly for this purpose.

Trace Styles

Currently there is just one type of trace style, though in the future we may implement additional types for micro-strip etc. The available trace styles are displayed when the Trace icon is selected. You can then edit any the styles by highlighting its name and clicking on the selector toggle.

Trace styles have a single attribute which defines their width.

New trace styles can be created using the New Trace Styles command on the Edit menu.

Via Styles

Via styles are much the same as pad styles except that only circular or square vias are permitted. The list of via styles is displayed when the Via icon is selected.

New via styles can be created using the New Via Styles command on the Edit menu and edited in the usual way via the Edit Via dialogue form.

For stitching zones use pads with the correct hole size, edit the pads and use the connect to net option to connect the pads to the net that the zones are on.

Style Management

When you start a new layout, an initial set of styles appears and these come from a system file called DEFAULT.STY which is kept in the "Library" directory of your Proteus installation. This file contains a selection of common pad styles that we feel you may find useful.

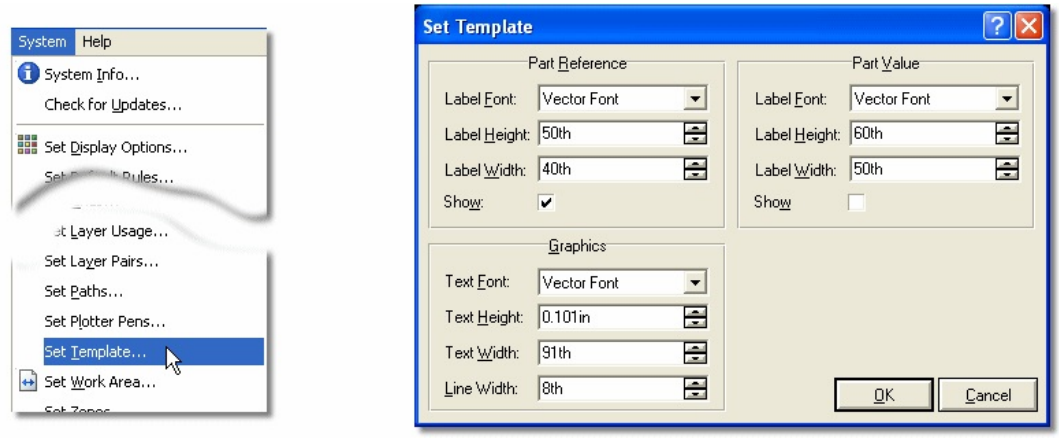
When you create or edit pad style you will see that the dialogue form contains a Changes box. If you select Local Edit then only the copy of the style maintained within the layout will be modified. If you select Update Defaults then DEFAULT.STY will also be updated.

Package library parts contain there own copies of styles used by their pads. When the package is brought into a layout for the first time, the style definitions will be taken from the library part if they do not already exist within the layout.

2D Graphics Styles

Global Defaults

Default values for 2D Graphics are set via the Technology Menu – Text Style command as shown in the dialogue below.



Setting the styles for the default 2D graphic styles

Changes to these settings are local to the current layout unless you save the current layout as a Template. In this case the technology information will be stored in the Template file and can be re-used by either initialising a new layout from the template or by applying the technology data to a layout via the command on the Technology Menu.

Individual Configuration

You can edit specific graphics in the normal way (right click to highlight, left click to edit). An example dialogue from editing a 2D Graphics Box is shown below.

Editing the 2D graphics box style

In order to change the width of the outline pen simply uncheck the Follow Global checkbox and enter the desired value in the edit field.

The fill style can also be edited for any 2D Graphics object which encompasses an area of free space. For example, you may well want to have a solid fill for a 2D graphics box but it has no meaning to have a fill style for a 2D graphics line.

Finally, you can tag multiple objects and then apply any changes to all tagged objects using the All Tagged Objects button. If you have multiple objects of different types tagged (for example, some graphics lines and graphic boxes) and exit with this button then ARES will update applicable changes to each object.

KEYBOARD SHORTCUTS

Configuration

ARES provides an interface for you to map the keyboard to a command set. This allows you to 'bring out' the functionality you commonly use according to personal preference. Context Sensitive Help is available for all fields within the configuration dialogue box.

Notes :

- Accelerators can use any combination of CTRL, SHIFT and ALT keys. For example, CTRL+X, CTRL+ALT+T,SHIFT+CTRL+1, etc.
- A Reset option is provided to reload all Proteus standard shortcuts. This will overwrite any configuration you currently had.
- Some European keyboards have an ALT GR key (normally found directly to the right of the space bar). Where present this translates to be CTRL+SHIFT for the purposes of assigning keyboard shortcuts.
- If you assign an ALT accelerator that is used as a menu accelerator then you will no longer be able to accelerate the menu from the shortcut. For example, ALT+T by default expands the Tools Menu within ARES. If you assign ALT+T to the Tidy command then you would have to have to access the Tools Menu using the mouse.
- Some commands are locked and their keyboard shortcuts cannot be assigned or re-assigned to another command. The assignation box on the dialogue box is grayed out for these commands.
- When the NUM LOCK is on the keys 0-9 on the numeric keypad are distinct from the 0-9 keys on the main keyboard. For example, you could assign CTRL+0 up to CTRL+9 on the main keyboard as shortcuts to the Most Recently Used file list (MRU list) and on the numeric keypad you can assign CTRL+NUM-0 up to CTRL+NUM-9 for different commands.
- When the NUM LOCK is off keys 0-9 on the numeric keypad action their secondary functions and these functions are considered identical to those on the main keypad. The key for the number 5 on the numeric keypad has no secondary function when NUM LOCK is off.
- You can save your custom keyboard configuration via the 'Export Keyboard Map' command on the Options tab. This allows you, for example, to port your keyboard configuration to another computer. Similarly, the 'Import Keyboard Map' command allows you to reload a custom keyboard configuration that has previously been saved.

TEMPLATES & TECHNOLOGY

Overview

ARES includes a comprehensive scheme for design re-use in the form of board templates and technology data. A user can create several templates representing common projects and then initialise new layouts from template to preload all of the configured information.

Template Files

A template file will contain the following:

- Board Edge
- 2D Graphics (e.g. Company Logo)
- Loose Pads (e.g. Mounting Holes)
- Additional components/tracks, etc.

In addition, the current settings of the technology data is always stored when a file is saved as a template.

Technology Data

The technology data set comprises all of the settings on the technology menu in ARES, namely:

- Design Rules and Net Classes
- Grid Settings
- Layer Usage
- Layer Pairs
- Text Styles
- Board Properties

Changes to these settings are local to the current layout until the user saves the file as a Template, in which case the current settings are saved inside the Template for re-use in future designs.

CREATING TEMPLATES

Overview

Most users will want to create one or more Templates of their own to reflect the type of boards that they develop. This is an extremely straightforward process:

- Define Technology Data.
- Define Physical Board Skeleton.
- Save the Template.

Defining Technology Data

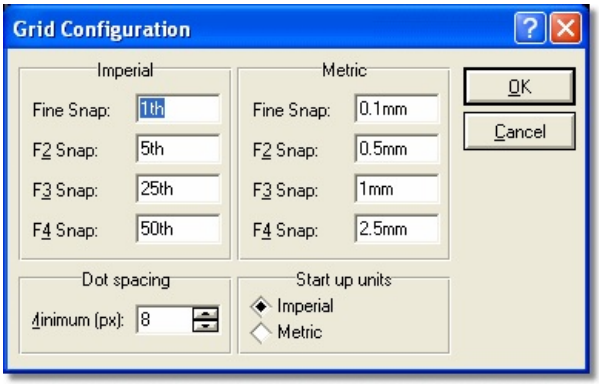
The technology data block stored in a Template file is basically all of the options in the Technology Menu.

Design Rules

This dialogue allows you to define and organise board constraints and also to provide routing information for the auto-router. Note that on the Net Classes Tab you can now create a new net class. This allows you to set up 'placeholders' so that if you work on a schematic and define that net class the design rules will be picked up when you initialise the layout file from the template.

Grids

This dialogue allows you to specify the snap settings for both imperial and metric co-ordinate systems. It also allows you to specify the default units of measurement for a new session.



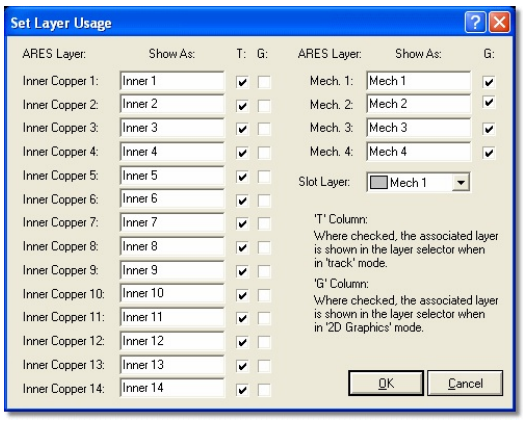
The start-up units define which units ARES defaults to when you start a new session rather than which units you were last working on when you closed ARES.

Layer Usage

This dialogue allows you to specify which layers are used in the board and to name the layers appropriately. Selecting or de-selecting a layer will add/remove it from the Layer Selector when in Track (T) or Graphics (G) mode.

For example, on a two layer board you might de-select all the inner layers from track mode as they will never be used.

You can also specify which layer is to be used for routing/slotting information and configure the selection of the mechanical layers. This information will be passed to the output system when generating Gerber or ODB++ files.



Layer Pairs

This dialogue allows you to specify the via through level for each layer on the board. This is used when routing manually and determines which layer you end up on when you place a via.

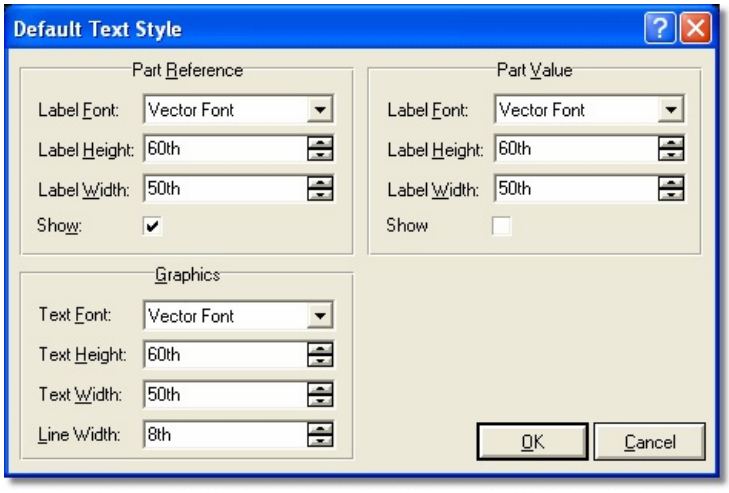
For example, on a two layer board, placing a via on top copper will take you to bottom copper and placing a via on bottom copper will take you to top copper.

The default settings on this dialogue are fine for most boards.

Using the SPACE bar in the main application with toggle the current layer according to the layer pair configuration for that layer.

Text and Graphics

This dialogue allows you to configure font settings and visibility for part reference, part values and 2D Graphics. You can also define the width of any 2D graphic lines used on the board.



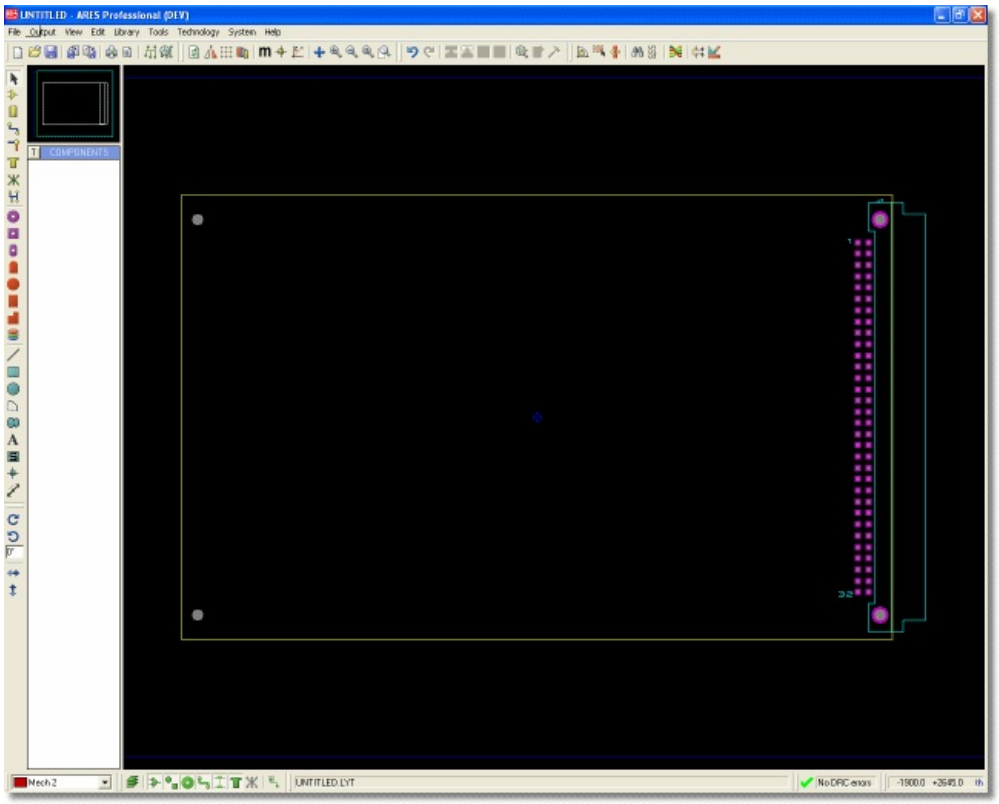
Board Properties

This dialogue allows for the configuration of the global work area as well as the both the board and feature thickness. The latter settings are used in the IDF output from ARES and for the 3D Viewer display.

Defining Board Skeleton

The board skeleton can consist of board edge, 2D Graphics and mounting holes. In the example below, we have set up a simple rectangular board edge with 3mm mounting holes positioned for mini-locking PCB supports and have included the Labcenter company logo.

You can also include components/tracks and other objects in a template. In the example below, we have a eurocard with a placed connector.

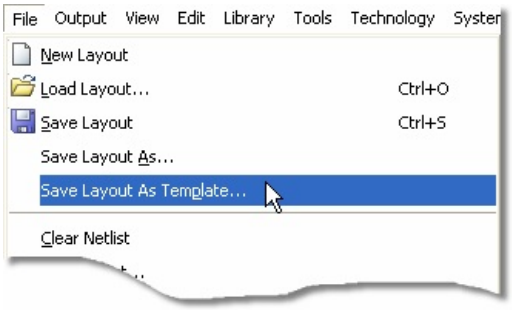


It is crucial to understand that templates that contain components will need to have their annotations matched to the schematic. In other words you need to make sure that the connector (J1) corresponds to the J1 connector on your schematic. If it corresponds to the J2 connector on the schematic then you need to change the annotation on the layout to be J2 before you netlist. Clearly, if you save a template with many components this may involve considerable manual re-annotation on the layout.

⚠ If you are at all unsure about this we strongly recommend that you do not save templates with components or tracks. You can easily strip these objects from an existing layout with the selection filter.

Saving the Template

Having completed the Template configuration we simply use the Save as Template command on the File Menu and provide a sensible name.



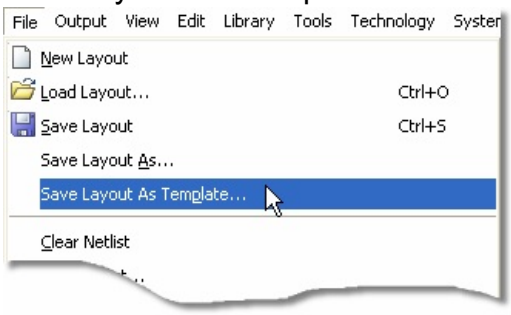
Whenever we start a new layout we will now be offered this Template as an option for the new board.

- More information on Saving Templates can be found here.

SAVING & LOADING TEMPLATES

Saving Templates

Invoking the 'Save as Template' command on the File Menu will save the current board skeleton and technology data on the current layout as a Template for re-use in future designs.



Saving the design as a template

Anything on the layout will be saved as a template when you invoke this command. This may not be what you want - for example, you may wish to use the board edge and mounting holes but not the components as they are specific to the current layout. You can easily strip such objects from the layout using the selection filter before saving as a template.

⚠ Saving a template that contains components and tracking can cause problems. When initialising from that template you **must** re-annotate any of the components on the layout to match the annotation of their equivalent schematic parts.

A couple of examples of removing items from a layout before saving the template are outlined below

To delete all components, zones and tracks from the layout

1. Draw a tagbox around the entire board.
2. Adjust the selection filter to include only loose pads (mounting holes) and graphics.

3. Hit the delete button
4. Remove any other unwanted graphics.

To remove stitching pads from the layout:

1. Use connectivity highlights to select the net that the pads are on.
2. Hit the delete button to delete the net (and therefore the pads).

To save only the board edge

1. Draw a tagbox around the entire board.
2. Select all object types except 2D graphics from the selection filter

3. Hit the delete button
4. Remove other unwanted 2D Graphics.

Saving Technology Data requires the saving of a template. If this is all you want to store you can simply delete the board entire before invoking the save command.

Loading Templates

When you create a new layout you are presented with the option to initialise the layout from one of the Templates. A dialogue form with all of the stored templates is presented both on File - New Layout or on the first netlist from the schematic in ISIS.

You cannot load a Template into an existing design as the board edge, location of graphics objects etc. will almost certainly be different. Instead, you can apply the Technology Data from an existing template into a current design via the Apply From Template command on the Technology Menu.

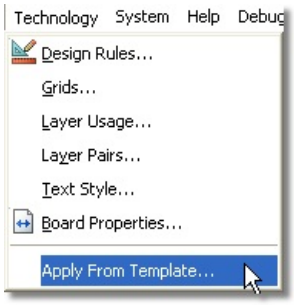
- More information on Applying Technology Data can be found here.

APPLYING TECHNOLOGY DATA

Overview

Technology data can only be saved as part of the template but it is sometimes useful to import only the technology information into another project. This allows you, for example, to re-use a set of design rules in multiple projects, regardless of the physical dimensions of the board.

The Apply from Template command is invoked from the Technology Menu.



Effects on Current Layout

When you apply technology data from a template the basic rule is that anything in the technology data will override its equivalent setting in the current layout. Specifically, the following rules are applied to the import:

- A net class/design rule in the current layout that has an equivalent net class/design rule in the imported technology data (same name) will be overwritten with the information in the technology data.
- A net class/design rule that no equivalent in the imported technology data will remain unchanged.
- The default settings (neck & relief styles and tolerances) will be overwritten with their equivalent settings in the technology data.
- Grid Settings, Layer Pairs, Layer Usage, Text Styles will all be updated from the technology data.
- Board and Feature thickness settings will be updated from the technology data. Work area will remain unchanged as the current layout may be larger than the settings in the technology data.

LIBRARY OVERVIEW

Introduction

As supplied there are 2 symbol libraries (which are shared with ISIS):

- SYSTEM.LIB
- USERSYM.LIB

and several package libraries including:

- PACKAGE.LIB
- CONNECTORS.LIB
- SMTCHIP.LIB
- SMTDISC.LIB
- SMTBGA.LIB
- USERPKG.LIB

You can, of course, create further libraries of your own using the Library Manager

Library Discipline

USERSYM.LIB and USERPKG.LIB are set to read/write; the rest are set to read-only. The idea is that you should only add things to USERSYM.LIB (new symbols) and USERPKG.LIB (new packages). This means that we can supply you with updates to the parts we have defined without risk of overwriting similarly named objects in your own libraries.

If you must change things in the read-only libraries, you can set them to read/write using the Properties command in Explorer under Windows or with the DOS ATTRIB program. The command

```
ATTRIB filename -R
```

sets a file to read/write whilst

```
ATTRIB filename +R
```

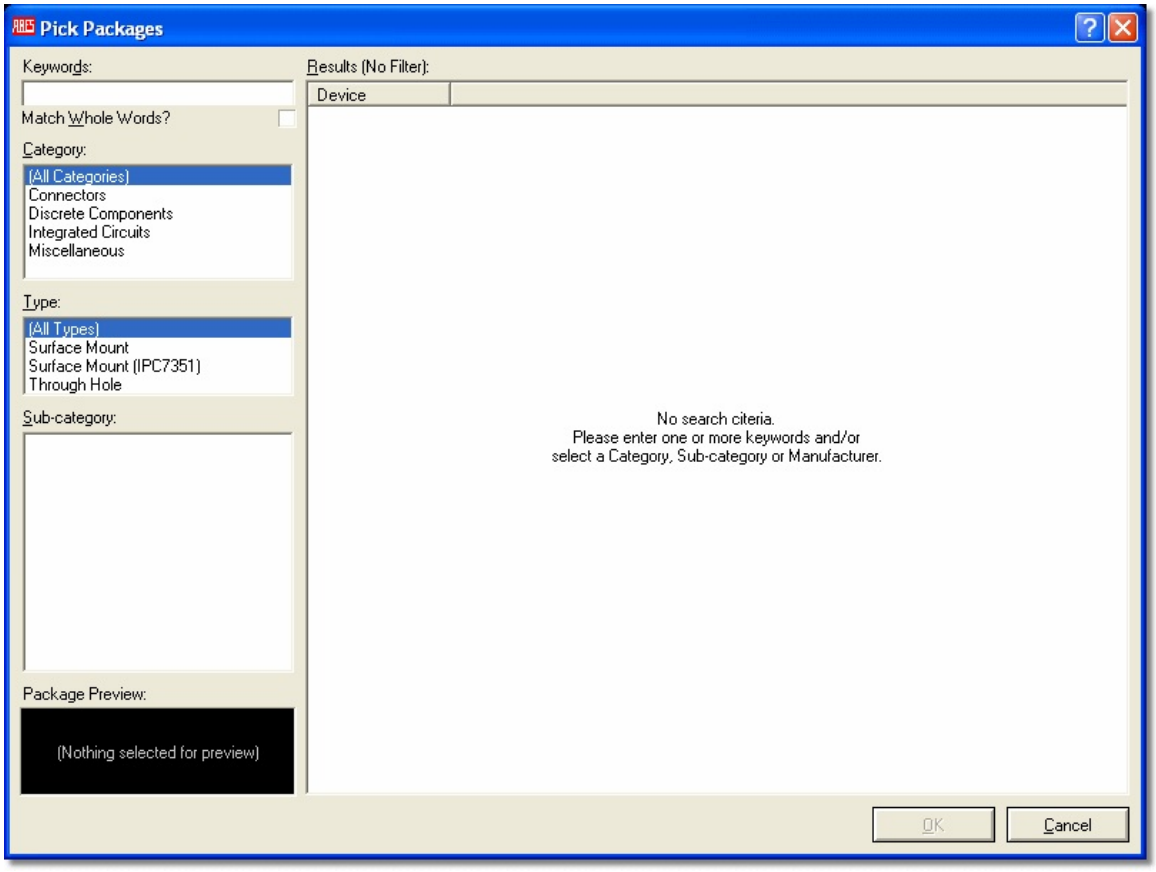
sets a file to read-only. Wildcards may also be used.

The Library Manager also includes an option to toggle the read/write status of libraries.

⚠ Make very sure that you know what you are doing before changing the attributes or writing to a library supplied by Labcenter. Future installations of the software will overwrite these libraries to facilitate update of existing parts and inclusion of new footprints.

The Library Browser

Each part in the package libraries is assigned a Category, a Type, a Sub-Category and a textual Description. This information is displayed by the Library Browser and makes it very easy to search the libraries for the parts you require.



Library Browser dialogue form

To search the libraries for a part using keywords

1. Enter keywords that describe the part into the Description field. The search operates by matching the keywords against the parts' names, descriptions, categories and sub-categories.

   If you enter several keywords, they must all be matched against a part for that part to show in the results. In other words, it is implicitly an AND search. Unless you check the Match Whole Words checkbox, then partial matches are allowed, so 'DIL' will match 'DIL08' and so on.
2. If the keyword(s) result in a lot of matches – for example 'capacitor' will match many parts in the libraries, you can use the Category, Type and Sub-Category fields to refine the search.
3. Pick the part(s) you require by double clicking them in the Results list.

To browse the libraries for a part by category and sub-category

1. Ensure that the Description field is empty.
2. Select the Category and (optionally) Type of the part you are looking for. This will cause the Sub-Category list to show options appropriate for the chosen category.
3. Select a sub-category to refine the search further.
4. If you still have a large number of results, you can type a keyword into the Description field in order to further refine your search.

Libraries from earlier versions of Proteus

Existing user package libraries from versions of Proteus prior to 6.6 will not contain category and sub-category information. Parts from these libraries will appear under the Unspecified category, and the library names will be used for the sub-categories.

Information on how to index your own libraries is provided in the ISIS help documentation (click

here to launch the topic in a separate window) and also in Appendix D of the printed manual.

Updating the Layout from the Libraries

If you pick a package or symbol that is already loaded into the layout, ARES will update it from the libraries on disk.

Note also:

- Where packages are concerned, you should be aware that ARES simply aligns the new part on the old part's origin. This means that the pads may well end up in different positions and some manual re-tracking of the board will be required.
- Where a there are two or more packages or symbols with the same name spread across several libraries, the Pick command will load the newest one. This is particularly helpful if you have changed one of our parts and put in USERPKG.LIB, since your version will be deemed newer than ours.

While generally speaking package update works seamlessly, be aware that in some situations a sensible results is simply not possible. For example, if you remake a part with fewer pins then an overlay replace is not feasible. You should always visually inspect the layout after updating footprints.

The Tidy Command

The Tidy command on the Edit menu performs several functions, one of which is to remove all packages and symbols from the selectors that are not actually in use on the layout.

This saves memory and also simplifies picking the ones you are actually using.

THE PACKAGE LIBRARY

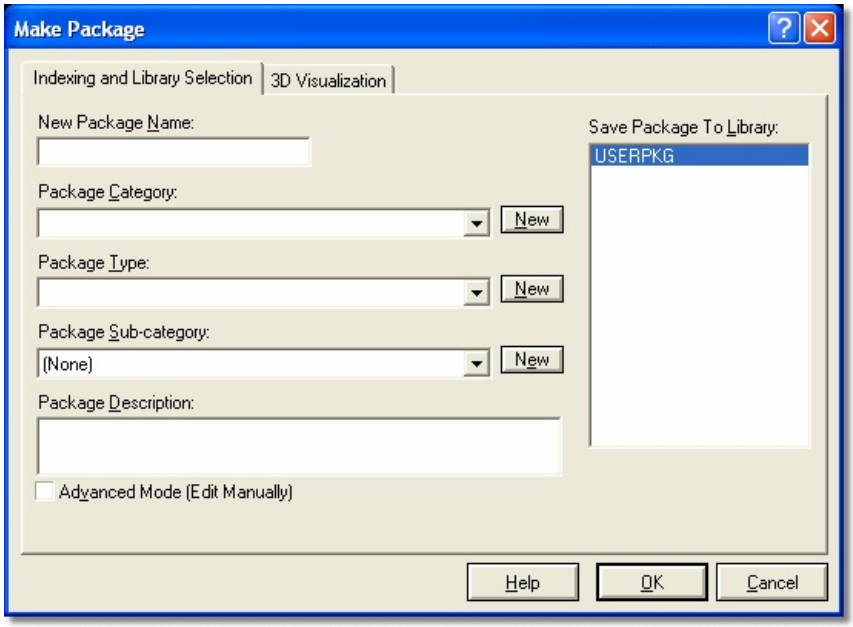
Overview

A package is a collection of pads and silk screen graphics used to site a component on the board. A library of many common package styles is provided and this will save you a great deal of time compared with manual methods of PCB layout. However there will be occasions when you will have to create your own packages, and the steps to follow are set out below.

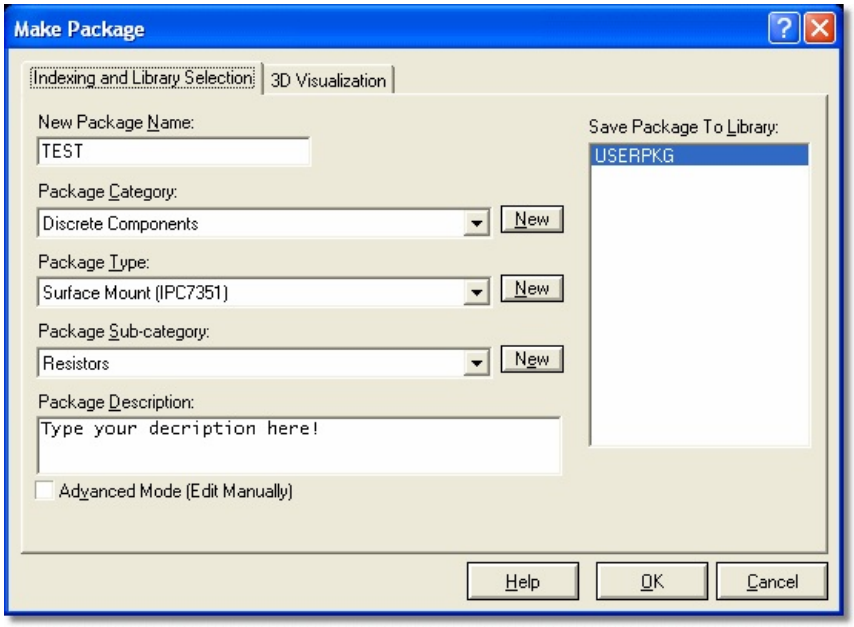
Making a Package

To make a new library package:

1. Select the appropriate pad shape icons and place the pads of the package as required. Use the Layer Selector to place the pads on the appropriate layers. See below for more information.
2. By default, ARES will give the pins numbers from '1' upwards, numbering them in the order they were placed. If this is not what is required, you must either renumber them with the Auto Name Generator, or else edit them manually. See below for more information.
3. Select the appropriate 2D graphics icons and place silk screen graphics as required. A package can have legends on both the component side and solder side screens; these will be swapped if the package is placed with the mirror icon active.
4. Tag all the objects by dragging a box round them.
5. Invoke the Make Package command on the Library menu. The Make Package dialogue form will appear as shown below



6. Enter a name, category, type, sub-category and description for the new package and then choose a package library to store it in - typically this will be USERPKG.LIB. When you are done you can either switch to the 3D Visualisation Tab to create a 3D model for the package or simply commit the parts to the library with no 3D information.



7. In the 3D Visualisation tab enter 3D information according to the provided syntax to present a 3D image of the package – this will be stored and will appear in the 3D Viewer henceforth. The job is greatly aided by a 3D Preview on the dialogue form which will update live as parameters are entered and and adjusted.
8. Click OK to commit the package (and optionally 3D information into the specified library.

Note that the description you enter via the Make Package dialogue form will be used via the Library Browser when searching via keywords. While not of particular importance in our test case you should give some thought when creating a part to ensure that the description contains keywords that you would use to identify the part during a search

The reference or 'anchor point' for a package can be determined by placing the ORIGIN marker. If there is no ORIGIN marker, the reference will be the centre of first pin placed. Markers are accessed by selecting the Marker icon.

It is also possible to define non-default positions for the reference and/or value labels. To do this, place the REFERENCE or VALUE markers at the required position(s).

A practical example of creating a package from scratch can be found in the tutorial section of the documentation.

Choosing the Correct Layers for the Pins

Ordinary pad styles can be placed on single copper layers, or on all the copper layers as defined by the current setting of the Layer Selector.

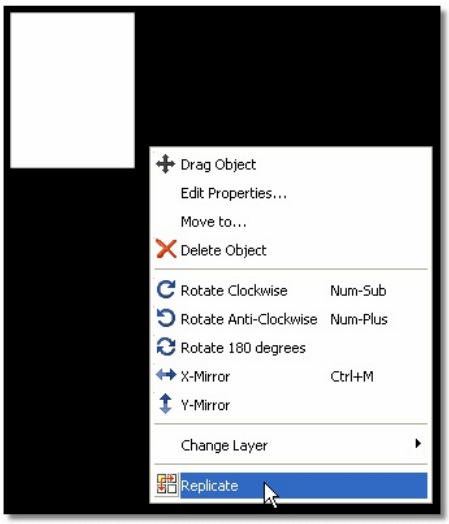
- For ordinary through hole parts, you should placed the pads on all the layers.
- For SMT parts, you should place the pads on the top layer only. If such a part is placed on the Solder Side, the pads will be automatically transferred to the bottom copper layer.
- For doubled sided edge connectors, you should place separate fingers on both the top and bottom copper layers.
- If you need different shaped pads on different layers for a through hole pin, you should use pad stacks as defined under Pad Stacks.

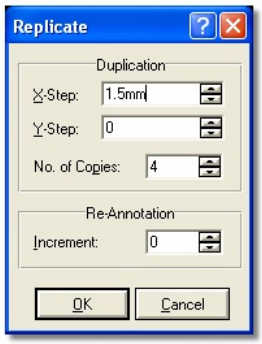
The Replicate Command

ARES provides a command to replicate tagged objects at a specified spacings in X and/or Y. This can be extremely useful when creating new packages as you can use it to lay out rows of pads with the spacings entered directly from the keyboard.

To place a row of pads by replication

1. Place a single pad of the require style at the first position of the row.
2. Ensure no other objects are tagged, and then tag the pad with the right mouse button.
3. Invoke the Replicate command from the Edit menu.
4. Enter the required X-Step (for a row) or Y-Step (for a column). Remember to enter a dimension suffix for these values - e.g. 20th or 1.5mm.
5. Enter the number of copies - this should be one less that the total number of pads required as you have placed one already.
6. Click OK - ARES will replicate the original pad as specified.





This technique is demonstrated in the appendix to the tutorial in the documentation.

You can extend this principle to creating grids of pads by replicating a pre-laid row.

Pin Numbering

When a netlist is loaded, ARES attempts to match pin numbers appearing in the netlist with the pin numbers assigned to the pins of the component packages. Where a match cannot be found, this is an error and either the ISIS library part or the ARES library part must be changed.

When you are defining an ARES library part, there are three ways to control the pin numbering:

- By default, the pins will be numbering from '1' upwards in the order that they were placed.
- If you invoke the Auto Name Generator command on the Tools menu, you can then click on the pads in any order you like to assign incrementing pin numbers.
- If you select the Instant Edit icon, you can click on each pin and key in the pin numbers manually.

Note that trying to use a mix of default and manual pin numbering is likely to lead you to confusion and mistakes.

Other Points

ARES normally chooses locations for the part ID and value; these labels can always be moved after a part has been placed by tagging it, pointing at the label you wish to move, and dragging with the left mouse button depressed. The locations chosen will generally be good enough as is, unless many components are closely packed together.

However, if you wish to force the label to a particular default position, you can use the REFERENCE and VALUE markers, as described above.

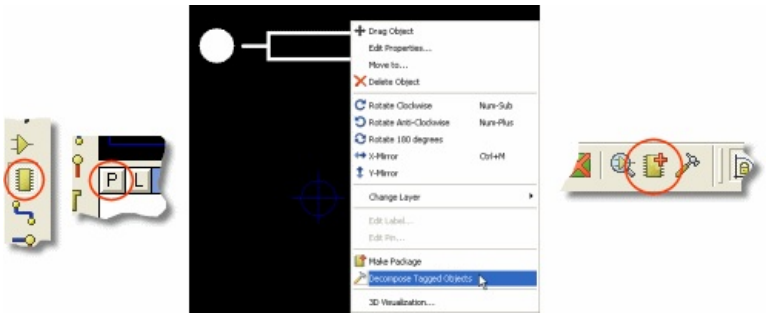
The size of the labels is set by the Text Style command on the Technology menu. This will

apply globally except for labels whose size has been manually edited.

Editing a Package

To edit and existing library package

1. Select the Package icon.
2. Pick the package to be edited from the library (using the toggle on the Object Selector) and place it on the work area.
3. Tag the package with the right hand mouse button and then invoke the Decompose command from the context menu. This will break the package into its constituent pads and 2D graphics and also place an ORIGIN marker at the location of the part's placement reference.
4. Edit the pads and graphics as required.
5. When you have finished you can use the Make Package command to re-store the package either with the same name as before, or as a new part.



Pick and place the package > Decompose and amend > re-make the package

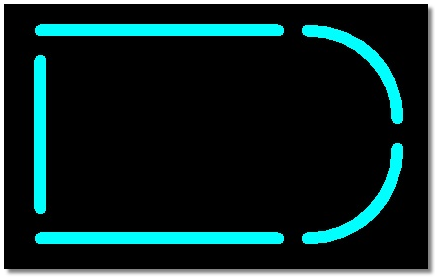
Of course, if the original package came from PACKAGE, which is normally read only, you will have to store it back to USERPKG. We recommend against using Library Manager to re-copy the part to PACKAGE as this can cause problems for you when we issue a new PACKAGE.LIB - how are you going to sort out which parts have been edited by you in order to preserve them?

⚠ Note that the pins will carry the pin numbers from the original package so decomposing a DIL14, sticking two pins on the end, and then re-making is not the way to make a DIL16 unless you are prepared the to re-number all the pins manually.

THE SYMBOL LIBRARY

Overview

A symbol is a group of graphic objects which are treated as a single object. For instance, using 4 lines and two arcs you can form an AND gate symbol.



A symbol may be created by tagging the objects you wish to form it and then invoking the Make Symbol command on the Edit menu. Layers are ignored for this purpose. A form will appear allowing you to name the symbol which will then be stored in the Symbol Library and made available for immediate placement from the Object Selector. If there is already a symbol with the same name, it will be replaced.

ARES quite happily allows a symbol to contain other symbols and/or other graphic objects. This allows you to make, for instance, a NAND gate out of the previously defined AND gate plus a circle.

As with packages, the placement reference for a symbol may be determined by placing an ORIGIN symbol.

ISIS and ARES symbols are essentially exchangeable, and all symbol libraries have the type 'PROTEUS SYMBOL LIBRARY'. However, ARES does not support all of the drawing appearance features of ISIS, so correct rendering of ISIS symbols with complex fill patterns is not guaranteed, especially for CADCAM output where we can say with some certainty that it will not work!

NETLIST FEATURES

Overview

A netlist produced using ISIS or some other schematic capture package contains as a minimum, a list of the components used in the design and a specification of how their pins are to be connected. The ability of ARES to use this kind of information distinguishes it from budget packages which provide what amounts to a computerized version of rub-down transfers and drafting film.

Our own netlist format is called SDF which stands for Schematic Description Format. As well as component name and connectivity information, an SDF file can also detail information concerning the package to be used for each component and the net class to be used for each net. As a result, it is possible to completely specify a PCB with an SDF file (and therefore from within ISIS), save for the physical positions of the components and the detailed routing of the interconnections.

The Load Netlist Command

The Netlist Loader is the means by which you import the data held in an SDF file into the current PCB layout.

A netlist can be loaded with the work area completely empty, with a set of placed and pre annotated components on the work area or, perhaps in order to effect a modification to an existing design, with a completely placed and routed layout loaded.

Using the Netlist Loader on an Empty Layout

With no components placed, all parts specified in the netlist are simply noted and displayed in the Object Selector ready for placement as described in Placing Components.

The Netlist Loader & Existing Components

Where components or annotated packages are already on the board, any that have been given the same names as those in the netlist are linked into the netlist database. They will then play a full part in subsequent netlisting activities such as ratsnest compilation. Components or annotated packages which are not specified in the netlist are tagged. This is a half way house between ignoring them and deleting them one click on the Delete icon will remove them if they really are superfluous.

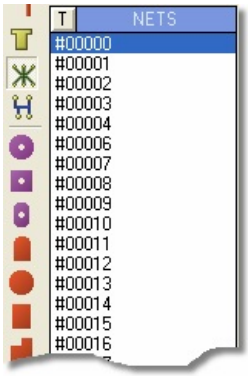
ARES checks to see if the library package specified for a component in the netlist matches that of the placed component on the layout. If the specified package differs, one of two things will happen:

- If the new package has the same number of pins as the existing one, then a replacement will occur such that the new package is placed with its pin 1 lying over the old package's pin 1. The component is tagged to highlight the fact that something has happened to it.
- If the new package has a different number of pins, the component is removed from the layout and added to the Object Selector for manual replacement.

Unannotated packages are ignored by the netlist loader. This makes them a suitable way to place footprints for parts which, for whatever reason, you do not want subject to full rigors of netlist based design. You should also note that because their pins are not specified in the netlist, they can be connected to anything without CRC violations occurring.

The Netlist Loader & Existing Tracking

In the case where there is tracking as well as components on the board, ARES checks the connections made by the tracking to see whether they agree with the netlist. The current net name for the track is shown at high magnification zoom once the track is routed – nets that have not been named in ISIS are given an autogenerated numerical name.



Auto generated net names

Where tracking is found that joins two nets (presumably as a result of a change to the schematic) the traces and vias involved are assigned to the VOID net.

VOID tracking and vias appear in flashing yellow, are ignored by the connectivity scanner and are not printed/plotted. They can be removed in one of two ways:

- By invoking the Tidy command on the Edit menu. This does other things too and can take some time on a large board.
- By selecting the Connectivity Highlight icon, then selecting the VOID net in the Object Selector and finally clicking its toggle to highlight the VOID tracking. Clicking on the Delete icon will then remove them.

VOID tracking shows you which bits of tracking need to be removed after a design change. If, having seen this, you decide against the design change, the procedure is to quit ARES without saving and then restore the schematic to its original state.

New connections specified in the netlist are will automatically appear as extra ratsnest lines.

The Bridge Style and Joining Nets

It is sometimes desired to join nets at a particular point (or points) on the layout - for example, where you have an analog and digital ground and you want to keep the digital return currents off the analog net. ARES allows you to do this by creating and using a trace style called the Bridge Style. Simply drag out a track from a pad on net A to a pad on net B using the bridge style.

⚠ By necessity the bridge style ignores connectivity and DRC conditions (it is after all explicitly used to connect two nets!). It is therefore entirely at the designers discretion as and when to use this – the software will not warn you if you make a mistake when using this style.

If the Bridge style does not appear in the parts bin when Trace Mode is selected you can easily create a trace style (Edit Menu – New Trace Style) and call it bridge.

Problems with Pin Numbers

As the netlist is loaded, ARES pulls in the specified package for each component and then attempts to match the pin numbers used for that component in the netlist against the pin numbering in the library-package. If the netlist references a pin number that does not appear in the library package then it displays an error message box and you must click OK to acknowledge. Problems in this area usually arise from one of the following causes:

- Specifying the wrong package - e.g. a bipolar transistor outline for a regulator.

- Failing to manually number pads which require non numeric 'numbers'. A typical example might be a DIN connector which has pin numbers like A1, A2 etc. Remember that by default, ARES always numbers the pads in the order you placed them.
- Leaving spaces on the ends of ISIS device pin names/numbers or in ARES pin numbers. This can be checked by editing the relevant object and examining closely the state of the baseline of the Data Entry Field on which the text sits. If there are any gaps in it, these designate the presence of spaces.

If you use the Packaging Tool in ISIS when creating new library parts, you should avoid any of the above problems.

Packaging Considerations

At some point during the design process, it is necessary to specify the library-package to be used for each component. With the ISIS/ARES system, this can be done:

- At the ISIS end by storing the package name in the PACKAGE property. This can be done either manually by editing each part in turn, or automatically using the Property Assignment Tool and/or ASCII Data Import facilities. This approach is by far the preferred one as it avoids having to re-enter data each time the netlist is loaded.
- At the ARES end as the netlist is loaded.

In the latter case, ARES will prompt you for the package name to use for each component in the netlist.

A limited facility is provided to automate this process for commonly used parts the file DEVICE.XLT can contain lines specifying device/package pairs such as

```
7400, DIL14
```

To use this feature, you must check the Enable Device->Package Lookup checkbox on the Environment Settings dialogue form. This command resides on the System menu.

You can then add entries to the DEVICE.XLT file either with a text editor or by selecting the Store button on the package input dialogue form.

Connectivity Highlight

Connectivity Highlight mode is selected by clicking left on the Connectivity Highlight icon. You can then:

- Highlight/tag a group of pads, tracks and vias connected to a pad by clicking left on the pad.
- Highlight all pads, tracks and vias connected to a net by selecting the required net from the Object Selector and then clicking left on the 'T' toggle.
- Delete all tagged pads, tracks and vias by clicking left on the Delete icon. Combined with the above this provides the means to rip up all tracking associated with a given net.
- Untag all pads/tracks/vias by clicking right at a point where there is no object or else by invoking the Redraw command.

RATSNEST FEATURES

Overview

The term Ratsnest is used to describe the pattern you get on the screen when the pin to pin connections specified in the netlist are shown as single straight lines rather than copper tracking. Once a netlist has been loaded and the components placed (or vice versa for that matter) the ratsnest gives a good visual impression of the complexity of the routing task that is before you or the autorouter. Furthermore, it gives an indication of the quality of component placement since the presence of lots of long ratsnest lines suggests that some components would be better placed closer together.

Automatic Ratsnest Re-calculation

ARES keeps the ratsnest up to date and optimized at all times, using new technology that is considerably faster than what went before. Add a component or delete a track and new ratsnest lines will appear. Make connections and ratsnest lines will disappear. It's as simple as that.

The term 'optimization' refers to that fact that ARES always displays the 'Minimum Spanning Tree' for each net. This means that the displayed ratsnest lines always represent the shortest interconnection pattern between the pads.

From the Design Rule Manager (Net Class Tab) you can specify a different colour for ratsnest lines of a particular net class. For example, you may have green ratsnest lines for SIGNAL traces and blue ratsnest lines for POWER traces. This is particularly useful as these colours are maintained during part placement, allowing you to make sensible decisions about the best position and rotation for the footprint.

The ratsnest can also be turned off wholesale by using the checkbox on the Layers dialogue on the Display menu. Ratsnest lines related to a particular net class can also be turned on and off using the Design Rule Manager (Net Class Tab) command on the Technology menu.

Dynamic Ratsnest

The ratsnest in ARES is intelligent and designed to help you quickly and easily manually place routes. When you initiate manual route placement on a netlisted design the ratsnest will change to indicate the nearest termination point (said point being highlighted for additional emphasis) to the current location of the mouse. Further, a 'shadow' trace will appear over the area that has already been tracked to indicate the layer and position of the area already tracked. The termination point and ratsnest indicator will update in real time as you move providing you with constant feedback on route termination.

Force Vectors

Force vectors are provided as a further aid to component placement. They appear as yellow arrows which point from the centre of each component to the point where it's ratsnest length (as defined by its current ratsnest lines) would be shortest. Another way of thinking about them is to consider each ratsnest line as a rubber band. The force vector then points to the place where the component would move to if released (strictly speaking this would also depend on the elastic linearity of the rubber bands, but we won't get into that - it's a good analogy!)

One oddity that can arise is that if you move the component to the position marked by the tip of its force vector, the ratsnest itself may change because a better minimum spanning tree may then be found. This can mean that the force vector then points to somewhere slightly different. We do not regard this as solvable; most PCB packages do not re-optimize the ratsnest after each edit, so such effects would not be so obvious - but they are still there.

The force vectors can be turned off by using the checkbox on the Layers dialogue on the Display menu; this setting is persistent across ARES settings.

Ratsnest Mode

Ratsnest mode is selected by clicking left on the Ratsnest icon. You can then

- Specify connections manually - if you are working without a schematic but still wish to use the auto-router.
- Perform pin and gate-swaps (where legal) by dragging tagged ratsnest lines from one pad to another.
- Tag a connection by pointing at it and clicking right.
- Tag all the connections for one net by selecting the net in the Object Selector and clicking left on the 'T' toggle.

In conjunction with the autorouter's ability to route all, tagged or untagged connections, ratsnest mode gives you the means to autoroute the board selectively.

ARES supports manual ratsnest entry in much the same way as you would manually route a track (with the ratsnest icon selected). You can then use the autorouter normally. Note however, that this is a less than ideal way to use the software and where possible boards should always be netlisted from ISIS or another suitable schematic package.

Manual Ratsnest Entry

If you are working without a schematic, it is possible to enter ratsnest connections manually. Connections are specified by clicking left the pads to be interconnected. ARES will automatically create net-names for connections specified in this way.

Note that where component pads are on single layers, you must use the layer selector to select the correct layer for each pad. If a ratsnest line must change layer, then you can press the SPACE bar to toggle between top and bottom whilst placing the ratsnest line.

This is not a recommended mode of working as many features - particularly power planes and design rule management - require netlist information and such functionality will not be available without a schematic netlist.

PINSWAP & GATESWAP

Overview

When used in conjunction with ISIS, ARES supports pin-swap/gate-swap changes to the connectivity whilst the layout is being routed. This means that you can choose to interchange the wiring to like pins, and/or interchange the use of like elements of multi-element parts. A full discussion of how to prepare your library parts to exploit this feature is given in the ISIS manual, but from the ARES side there are two ways of using this feature:

Manual Pin-Swap/Gate-Swap

To perform a manual pin or gate swap

1. Select Ratsnest icon.
2. Click right on the source pin. For a gate-swap, this can be any member of the gate. The ratsnest lines connected to the pin will highlight. In addition, legal destination pins will also highlight.
3. Hold down the left mouse button and drag the ratsnest lines to the required destination pin.
4. Release the left button. ARES will make the change, updating the ratsnest and force vectors as appropriate. In the case of a gate-swap, ARES will move other ratsnest lines automatically.

It is possible to combine a pin-swap and a gate-swap in one operation - for example, swapping input A gate 1 (pin 1) with input B gate 2 (pin 5) on a 7400 will do just this.

⚠ Pin-swaps and Gate-swaps constitute changes to the connectivity of your design. ARES uses the pin-swap and gate-swap data specified in the ISIS libraries to decide what is, and is not, a valid swap. If there are errors in this data, then ARES may well suggest illegal swaps. We will not, under any circumstances, be held liable for any costs incurred or losses arising as result of such mishaps, whether the error be in your library parts or ours or in the software itself. We strongly recommend that you check that the swaps you make are really legal, and that you prototype your PCB prior to the manufacture of large quantities.

Automatic Pin-Swap/Gate-Swap

In the case of a board with a large number of possible gate-swaps (the SHIFTPCB sample is a spectacular example) in can be very hard to find the best arrangement. For these cases we have provided an automatic gate-swap optimizer.

To perform automatic gate swap optimization

1. Invoke the Gate-swap Optimizer command from the Tools menu.
2. ARES will make repeated passes trying the current set of possible swaps. The process repeats until no reduction in ratsnest length is achieved.

You will need to have a PCB Design Level 2 or higher to access this feature.

⚠ The Gate-Swap Optimizer relies entirely on the gate-swap data specified in the ISIS component libraries to decide what is, and is not, a valid swap. If there are errors in this data, then the swap-optimizer is likely to make erroneous changes to the connectivity of your design. We will not, under any circumstances, be held liable for any costs incurred or losses arising as result of such mishaps, whether the error be in your library parts or ours or in the software itself. We strongly recommend that this command be used only if you are

going to prototype your PCB prior to manufacture.

Synchronisation with the Schematic

Whether manual or automatic swaps are to be performed, it is a requirement that the changes can be reflected or 'back-annotated' into the schematic. For this to occur successfully, the schematic must not have been changed as well.

PROTEUS manages this by using the netlist file as a kind of token. If ARES cannot find an up to date netlist, it will not allow changes, and if ISIS makes changes it deletes the netlist.

When ARES does make changes it writes out a back annotation file (extension 'BAF') the next time the PCB is saved. ISIS picks this up the next time it comes to the foreground. If ARES has changes that are not saved, ISIS will not allow changes.

This scheme will prevent the making of simultaneous changes to both schematic and PCB in normal use. It is, of course, possible to circumvent the token mechanism by renaming files, editing and copying back, editing on other machines etc. If you deliberately contrive to modify the schematic and PCB at the same time, then you must live with the consequences! The only cure for such situations is to check carefully that the schematic and PCB are the same and then re-load the netlist into ARES.

Further discussion of pin-swap/gate-swap is appears in the chapter entitled ISIS and ARES in the ISIS manual.

ROUTING CLASSES

Overview

In ARES a net class defines everything about how the nets assigned to it are to be routed. This information includes:

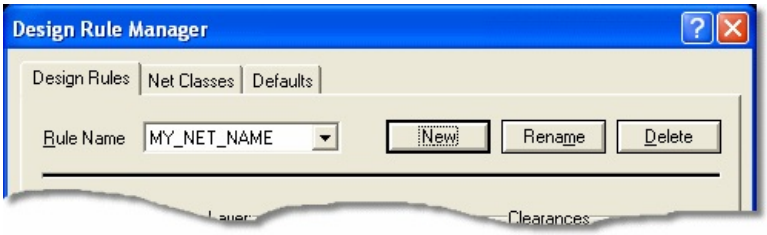
- The trace and via styles to use.
- The via type to use normal, buried or blind.
- Assorted controls for the auto router.

The beauty of encapsulating all these properties in a single entity and then assigning nets to it as opposed to assigning the properties separately to each net is that it greatly reduces the amount of information that need be entered on the schematic.

In older versions of Proteus a net class was termed as a strategy.

Net Classes and the Netlist

ISIS can associate a named property in this case a net class name with a net by placing a net label such as CLASS=MY_NET_NAME. This will then be processed by the ISIS netlist compiler and will appear as a net property just after the net name.



Your net appears in the Net Class field

For example:

```
VDD,2,CLASS=POWER
U1,14
U2,14
```

If you are using some other schematics package, much depends on its ability to process net properties in some manner similar to the above. If so, then SDFGEN will include them in its output in the same manner as above. If not, then you have three choices:

- Use net names of the form such as VDD=POWER. ARES will parse this into a net called VDD assigned to a net class called POWER.
- Edit the SDF netlist with a text editor and add in the net class information by hand. By judicious use of the macro facilities of your text editor, this can be a very convenient and flexible approach.
- Ignore the net class capabilities altogether as far as the schematic side of things is concerned. Given the automatic net class assignment described in the next section, this may be adequate (although it will limit your flexibility) and is certainly the simplest option.

Net Classes and Design Rules

Net Classes are **extremely** important in setting Design Rules for your layout. Once you have named a net class in ISIS and netlisted to ARES, this net class becomes available as an option in the Design Rule Manager. This in turn allows you to control clearances between pads and traces for that net class by creating additional rules as required.

Special Net Class Names

The most common reason for needing different net classes is to distinguish power connections from signal connections in order that the power routes can be made out of thicker tracking. Also, especially when using the autorouter it is often an idea if memory bus connections are given special treatment as they really need to be routed in an orderly, regular fashion.

In order to help with achieving the above distinctions, the net class names POWER, BUS and SIGNAL are given special meaning and certain forms of net name will automatically be assigned to these net classes.

- The net names GND and VCC default to the POWER net class unless they are explicitly assigned to a different one in the netlist.
- Net names of the form D[0] (it is the square brackets that are important) default to the BUS net class. This is now somewhat obsolete (at least as far as ISIS is concerned) since you can place a bus label such as CLASS=BUS directly on a bus segment. Nevertheless, this syntax may be of use with other schematics packages which do not have net properties.
- All other net names default to the SIGNAL net class.

This scheme will enable many boards to be handled without need to explicitly specify net classes on the schematic or in the netlist.

Editing a Net Class

Net Classes are created automatically by the Netlist Loader as their names are encountered. However, you will normally then need to edit them in order to set the various fields to whatever you require.

Net Classes may be edited using the Design Rule Manager on the Technology menu.

BACK-ANNOTATION

Overview

Back annotation is the process whereby changes to the component annotation on the PCB can be fed back into the schematic. This would normally be done if it was felt that a particular order of annotation on the PCB would aid the production department in making the board. In fact, the most natural order for this purpose is if placement path for placing the components in successive order is a minimum.

Manual Re-Annotation

It is perfectly possible to re-annotate the board manually by editing each component reference label as required. PROTEUS uses internal unique identifiers to track 'was-is' data so it does not matter what changes you make, in what order, or when they are eventually read back into ISIS.

The only restriction is that it is not advisable to renumber connectors made from Physical Terminals in ISIS. This is because ISIS cannot re-annotate them - it would involve it finding not only the terminals but also the references to the part name in the *FIELD blocks. The work around is to manually re-annotate in both ISIS and ARES. A warning to this effect is displayed if you attempt such a change.

Automatic Re-Annotation

Should your sole aim be to renumber the components to give a geometrically sensible placement order, then the Component Re-Annotator command on the Tools menu will do this for you.

The process starts at the top left of the board and proceeds for all components picking the next nearest one not yet processed.

Back-Annotation to ISIS

Both changes to the component numbering and also changes to the connectivity caused by pin-swap/gate-swap operations will be automatically picked up by ISIS. This works whether ARES is running or not.

You have a choice of allowing these updates to occur entirely automatically (in which case, your PCB will be saved when you switch to ISIS) or to do it manually, in which case ISIS will prevent you from editing the schematic until the PCB is saved. This choice appears on the Set Environment command on the System menu in ISIS.

Further discussion of this mechanism is provided in the ISIS manual in the chapter entitled ISIS AND ARES.

REVERSE NETLISTING

Overview

ARES can create a netlist from a placed and routed PCB. This has a number of uses:

- As a way to check the wiring where no schematic is available.
- In cases where you wish to re-route a board, without risking changing its connectivity, and do not have the schematic.
- As an aid in reverse engineering designs brought in from Gerbit or elsewhere.
- As a debugging aid where there is some doubt about whether the PCB corresponds with the schematic.

To create a netlist from a PCB

1. Load the PCB in the usual way.
2. Invoke the Save Netlist command from the File menu.
3. Choose a filename for the netlist.
4. Click OK

In cases where the layout file already contains a netlist, the net-names used in this netlist will be used. Otherwise numeric net names will be generated automatically.

AUTOPLACEMENT

Introduction

As well as an auto-router, ARES incorporates an algorithm for automatic component placement. As with auto-routing, this may not perform as well as a human operator in all circumstances but on the other hand it can save an enormous amount of time and effort. In addition, the resulting placement can usually be ‘hand tweaked’ - perhaps more so than can be said of auto-routed tracking. At any rate, we would recommend that most users at least try the auto-placer as a way of quickly achieving a starting point.

The overall system is now sufficiently developed that small-medium complexity boards such as our CPU sample can be converted from schematic to PCB with no human interaction whatsoever!

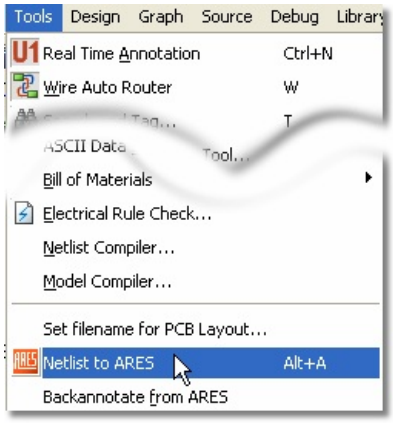
USING THE AUTOPLACER

Overview

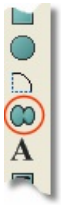
In operation, the placer requires only that the board outline be defined (by placing lines and arcs on the Edge layer) and that a netlist be loaded to specify the components to be placed. A more detailed sequence of events is set out below:

To auto-place a PCB layout

1. Create a schematic in ISIS assigning group properties to those components which you wish to be located physically close to each other in the final layout.
2. Invoke the Netlist to ARES command on Tools menu in ISIS.

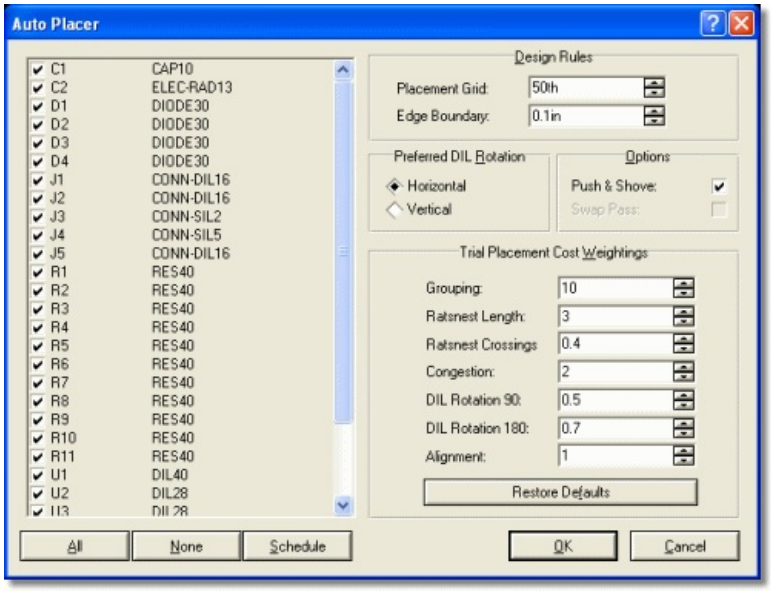


3. Draw a board outline of the correct dimensions using the 2D graphics tools on the board Edge layer. Ensure that the boundary forms a closed polygon. The best way to do this is to use a Path object.

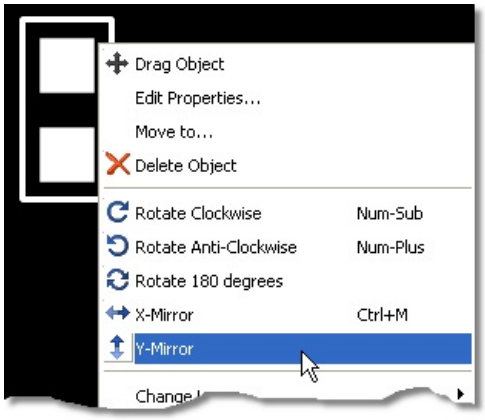


4. Place any components which are immovable on the board. Hand placed components are viewed as fixed, these will be treated as obstacles by the auto-placer.
5. Select Auto-Placer command from the Tools menu in ARES and adjust the various options as you see fit. See below for more information about this.

6. Select those components from the list which you wish to be placed automatically.



7. Modify the Design Rules and the weightings suit the board.
8. Click OK.
9. Move and rotate the components as required and start the auto-placer again.

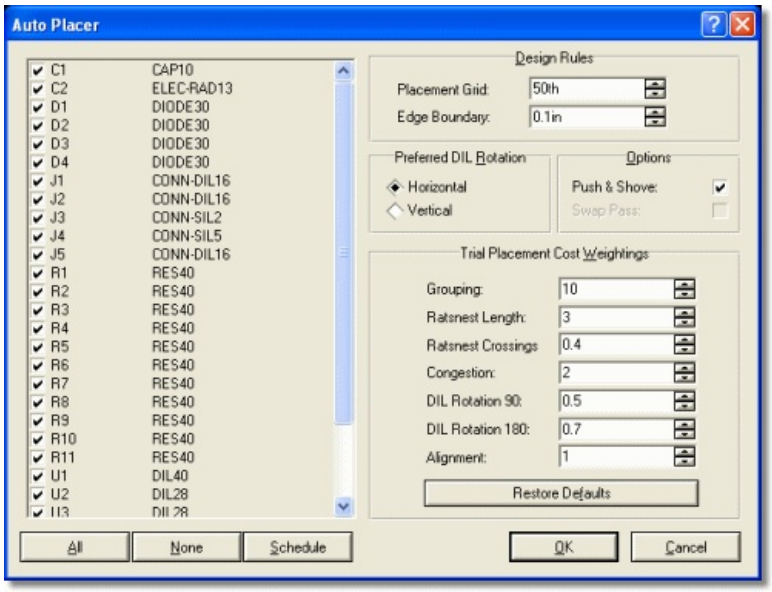


10. Once you have finished with the auto-placer, manually place any remaining components and start routing.

THE AUTOPLACER DIALOGUE

Overview

The auto-placer is highly configurable and as a result the dialogue form may at first appear somewhat daunting. However, it will operate quite happily with the default settings for most purposes. The function of the various fields is discussed below.



The Auto-placer dialogue form

The Component Selector

The left hand pane show a list of all the components still to be placed, when the dialog form is first displayed the components are listed in alphabetical order with all the components selected for placement.

The function of the All and None buttons should be fairly obvious; the Schedule requires a little more explanation. By the default, the components are presented in alphanumeric order but by pressing the Schedule button they are then presented in the order that they will be placed by the auto-placer. In this case, components not-selected for placement always appear at the bottom of the list.

Design Rules

The Placement Grid defines the step size used by the auto-placer when trying to find a position for each component. This should always be set to a multiple of the grid size you are going to use for auto-routing. Values other than 100th, 50th or 25th would be highly unusual.

The Edge Boundary defines the minimum distance to be allowed between any component and the edge of the board as drawn on the edge layer.

Trial Placements and Cost Weightings

Auto-placement is achieved by trying the next component at various positions on the board in an attempt to minimise a number of key factors. The relative importance of these factors is set by you, the user, by moving the sliders to the right (more important) or left (less important) for each of the seven categories.

Grouping

It is often necessary to force two (or more) components to be placed together (such as de-coupling capacitors) and this can be achieved in one of two ways. Firstly you can manually edit a placement to move the components to the desired location, however this rather negates the use of auto-placement. Secondly you can give the auto-placer a helping hand by telling it which components are to be grouped together.

A particular problem arises with decoupling capacitors since the auto-placer will, other things being equal, tend to place all of them in one place since this will give the most routable solution! To prevent this you can use the GROUP property in ISIS. If the decoupling capacitor C1 is intended to decouple the chip U1, then you should add the property

```
GROUP=U1
```

to the capacitor. Then, when the placer comes to place C1, it will give high costs to positions which are far from U1 and a placement close to the device will result.

If a number of components are to be grouped together they may all be given a common group name, rather than a component name, i.e. to group U1, U2, U3, C1 & C5 together you could add the property “GROUP=TIMER” to each component.

To add properties to a large number of components use the Property Assignment Tool in ISIS.

Ratsnest Length

Arguably the most important weighting factor in terms of routability, this assigns importance to the minimisation of the length of inter-connectivity of components before routing. Note however that in our experience this should be balanced with some knowledge of the number of ratsnest crossings. A board with short total ratsnest length and a large number of crossings can often be unroutable.

Ratsnest Crossings

This weighting defines the importance of minimising the number of ratsnest lines which cross. As a component is moved around the board the ratsnest length and number of crossing changes, both of these factors will affect the overall routability of the final placement. It is important to get a good balance between this and ratsnest length.

Congestion

Small components such as resistors and capacitors may be encouraged to avoid highly congested areas of the board by increasing the importance of the congestion weighting. Note that when a small component has been associated with another component via the GROUP property congestion is ignored for that component.

DIL Packages

Components which are allocated DIL packages are treated as a special case. It is seen as desirable to align DIL packages such that they all have the same orientation. To this end a number of controls on the dialogue are tailored to the placement of DIL packages. Preferred DIL rotation is by default horizontal but the shape of your board may dictate that vertically placed DIL packages make better use of the available space. Two further controls in the cost weightings group on the dialogue form define how heavily penalised a component will be if the package is placed with either 90 or 180 degree rotation from the default. If you are not concerned about the orientation of your DIL packages then set these costs to zero.

Alignment

A successful PCB layout is judged not only on routability but also on aesthetics. One key factor in whether a PCB looks right is whether components are aligned. This weighting factor stresses the importance of aligning component edges. Increased alignment can sometimes also aid routing since if like components are aligned then it follows that their pins are also aligned.

Options

Push & Shove

The auto-placer recognises two types of component, those initially placed by the user which are deemed to be fixed and immovable and those placed by the auto-placer which may or may not have been moved by the user. The later will be pushed and shoved around the board where possible in an attempt to best place the remaining components. This shuffling can become restrictive and counter-productive on boards where the pre-placed components are in the centre of the board. In such cases Push & Shove should be disabled.

Swap Pass

Since the components are placed sequentially, it may well be that further improvements can be made to the placement by interchanging like package styles once all the parts are down. This process can be enabled by selecting the Swap Pass checkbox.

OCCUPANCY DEFINITIONS

Overview

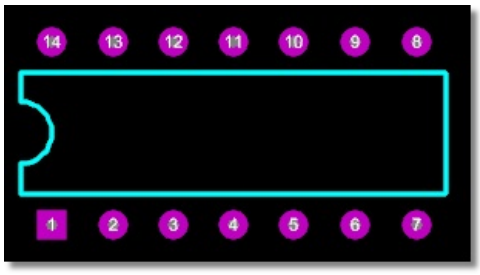
Each component that is placed on the board is treated as an obstacle for the remaining components to be placed. By default the area which is occupied by a component is assumed to be the smallest box which contains all silk screen graphics and component pins. Any two such boxes may not touch or overlap.

It is quite feasible that you will wish to modify the area taken by an object such that, for example, a DIL package has additional space on the sides occupied by the pins, or that a connector has increased occupancy on one side only. To achieve this ARES provides an Occupancy layer onto which 2D graphics may be drawn in much the same way as silk screen graphics are added.

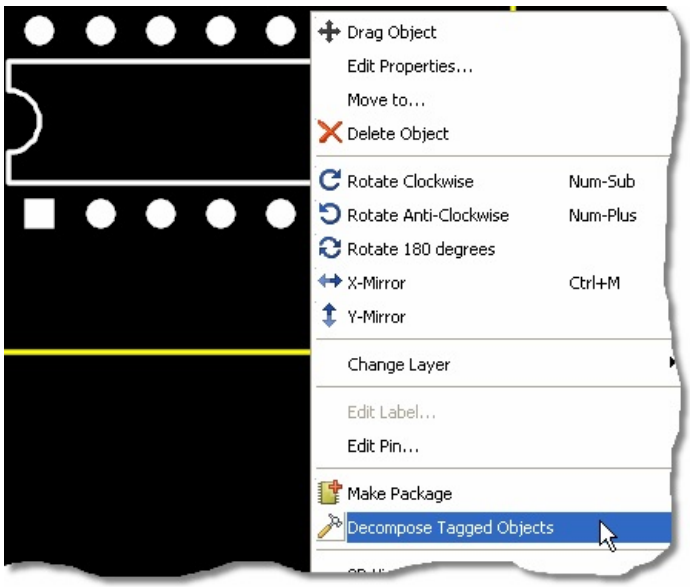
Occupancy graphics are not normally seen but may be examined and edited by first decomposing the package, amending or adding graphics as necessary and then invoking the Make Package command from the Tools menu.

To add an occupancy definition to a package

1. Place a copy of the package on the layout.

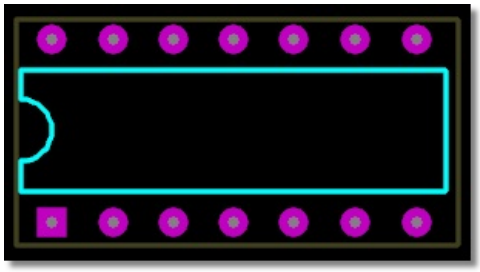


2. Tag it and then invoke the Decompose command on the Edit menu.

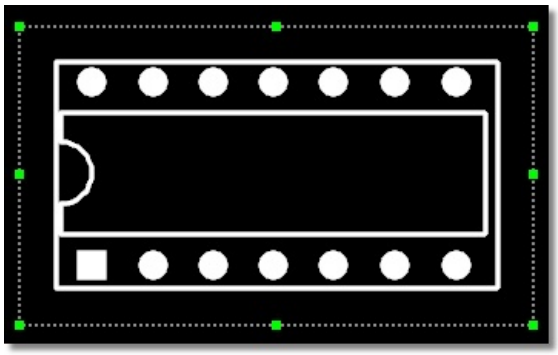


3. Select the Occupancy layer in the Layer Selector.

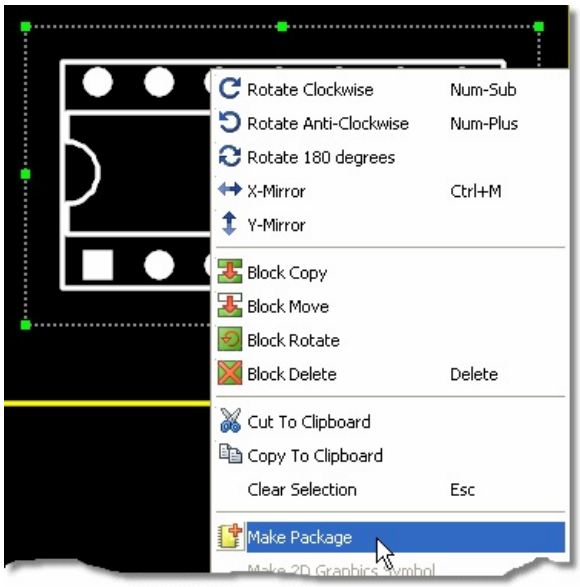
4. Place either a box or a circle to define the occupancy.



5. Tag all the elements of the package.



6. Use the Make Package command to store your handiwork back the library.



The current implementation does not support the use of path objects in the context. Multiple boxes and circles are allowed at present, but this may be dropped when we code support for path objects as testing for multiple objects carries a speed penalty.

Many of the newer library parts and in particular the IPC-7351 standard footprint libraries come pre-supplied with occupancy definitions as per the IPC standards.

AUTO-PLACEMENT

Limitations

The auto-placer is another very useful tool in the armoury provided by PROTEUS but there are a number of limitations of which you should be aware:

- It can place components on the top of the board only. Boards requiring double sided placement are beyond its scope at present.
- Whilst the auto-placer can operate on boards on which some components have been pre-placed, our experience is that the more of these there are, the less well it performs. If possible, it is better to let in place everything and then move things about afterwards. Pre-placing components in the middle of the board will cripple its performance because this interferes with its ability to shuffle the components about as placement proceeds. If you must have pre-placed components in the middle you may find it does a little better if the Push & Shove option is disabled but in general it will do badly.
- Boards which are decidedly non-rectangular also interfere with the Push & Shove mechanism and again the placer will perform badly in such cases. Boards with small cut-outs etc. should not present a problem, however.

INTRODUCTION

Overview

The ARES auto router is the one of the most powerful tools in the Proteus system and is capable of saving you an enormous amount of time and effort. The two principle technologies applied in the router are discussed below.

Shape Based Auto Routing

The ARES Router features cutting-edge shape based algorithms for Shape-Based Autorouting. By contrast to traditional gridded maze autorouters, a shape-based approach allows for more efficient use of routing area and is more suited to handle complex design rules requirements of high density PCB designs.

Conflict Reduction Algorithms

In addition to its shape-based architecture, the ARES router places routes with conflicts and then uses a multi-pass cost-based conflict reduction algorithm to find a routing solution adapting to the natural flow of the nets. On the initial passes the router uses a relatively low cost for accepting a routing path solution with crossing and/or clearance conflicts. Costing are then slowly increased on subsequent passes until all conflicts are removed. By working in this way (allowing nets to be routed with conflicts, autorouted wires will initially start with an ideal path and the final routing solution will closely mimic manual routing in terms of optimum wire length and number of vias. This adaptive routing technology is a proven approach to systematically reach high completion rates on complex high density PCB Designs.

Modes of Operation

Despite the considerable internal complexity of the router, it is extremely easy to use in practise. There are three internal modes of operation, which can be thought of in simple terms as fully automated, scriptable and interactive. A fourth option exists allowing users with an installed and licensed copy of the ELECTRA router to directly invoke this application. All of these modes are discussed in more detail in subsequent topics.

The Auto-Router and Design Rules

The ARES auto-router is truly world class and can save you valuable hours of development time. However, it can only route with the guidelines you provide it and it stands to reason therefore that the more information you give the better the routing solution. Fortunately, ARES also includes a comprehensive and flexible design rule system where you can specify a series of board constraints and clearances, trace widths and via styles for named net classes (e.g power, signal) and even the layers on which you wish given net classes to be routed on. All of these rules can be created and edited from the Design Rule Manager on the Technology Menu and will be honoured by ARES when routing the board.

Whilst ARES will quite happily route a board with the pre-supplied global design rule we strongly recommend you make use of the Design Rule Manager to specify constraints and information relevant to your particular project.

More information about creating design rules and the design rule manager can be found here.

Live Status Reporting

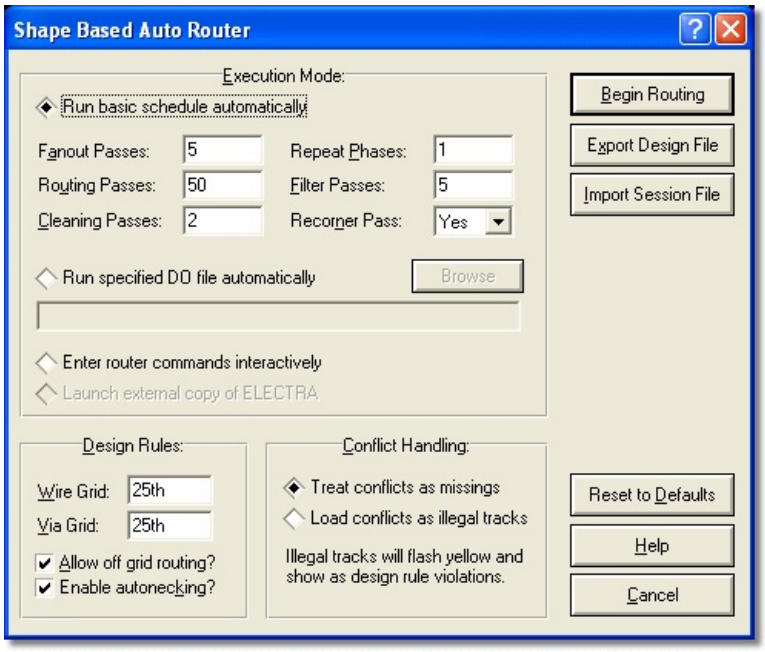
Once you have specified the required design constraints and invoked the autorouter the status bar at the bottom of the editing window in ARES will report on the current routing percentage completed and provide additional information on routing behaviour. Specifically, the type of

routing pass in progress, the current pass, the number of conflicts and the percentage completed.

AUTOROUTER DIALOGUE FORM

Overview

All autorouting operations – regardless of the mode of operation - start by invoking the Autorouter command on the Tools Menu in ARES. This topic discusses the various configurations on the dialogue form.



Note that the despite it's seeming complexity, the basic operation of the router is extremely simple. Many configuration options are provided for additional flexibility and the default values are sufficient for all but the most complex of boards. Please see the examples section for more practical usage guidelines or working with the autorouter interactively for more information on this mode

Execution Modes

The main section of the dialogue form allows you to choose one of four execution modes; Automated Routing, Scripted Routing (run a script file), interactive routing (type routing commands) or, if you have a copy of ELECTRA, to launch the ELECTRA Autorouter and route the board externally. Some of these radio buttons may be disabled based on the following rules:

- If you have a PCB Design Starter Kit, a PCB Design Level 1 or a PCB Design Level 1+ then the Scripted Routing mode and the Interactive Routing Mode will be disabled.
- If you do not have a license for the ELECTRA autorouter or ELECTRA is not installed on your computer then the ELECTRA Mode will be disabled.

Several additional options are available in the fully automated mode. Whilst in most cases the default settings will suffice these options provide extra flexibility for more complex or densely packed PCB's.

Fanout Passes

These passes will autoroute small escape wires with a via from SMD pads, thereby allowing easier connection to the pads during the main routing passes.

Routing Passes

The number of principle routing passes. There is only marginal benefit in increasing this beyond 50; for complicated or densely packed boards we recommend increasing the repeat phases rather than the number of routing passes.

Cleaning Passes

Cleaning passes will basically rip-up and reroute each net with high via cost and the highest cost for making clearance and crossing violation. The default value is adequate for almost all boards.

Repeat Phases

This allows you to specify the number of times the routing passes and cleaning passes will repeat. The ARES Autorouter is an adaptive shape based router so repeating the route/clean passes will typically help achieve higher completion. The default value is adequate for most boards but increasing the repeat value will have benefit in more complex or densely packed layouts.

Filter Passes

The filter pass will be performed after the router has repeated the route/clean passes the specified number of times. This will simply rip any tracks that have conflicts at the time of invocation. The default value for this is fine for almost all layouts.

Recorner Pass

This pass is performed last and changes 90 degree wire corners to 135 degrees to reduce track length.

Routing Rules

These options provide additional guidelines to the router and indicate the default behaviour for the routing algorithms. As with the scheduling options detailed above, the default values will suffice for the vast majority of boards.

Wire Grid

Specify the default wire routing grid. This advises the router as to the preferred grid for placing tracks but does not actually constrain it to routing on that grid.

Via Grid

Specify the default via routing grid. This tells the router the preferred grid for placing vias but does not constrain it to routing on that grid.

Allow Off Grid Routing?

This option, defaulted to on, determines whether the router is constrained to routing on grid (grid based routing) or is able to route off grid when necessary. We strongly advise leaving this option enabled as disabling this option will adversely affect router performance in most cases.

Conflict Handling

The ARES router works on a costing basis so it first places all tracks according to their ‘ideal’ position and then increases the costing of illegal tracks until there are no conflicts remaining. This means that in the unlikely event that the router does not achieve completion there will be illegal tracks placed on the boards and these options allow you to configure how you would like ARES to handle such tracks.

Treat Conflicts as Missings

In this case when the router is aborted or stops any tracks that cross other tracks or are deemed illegal are ripped from the layout and the connection will revert to a ratsnest line in ARES.

Treat Conflicts as Illegal Tracks

When selected, all conflicts will be left as illegal tracks (showing design rule violations and flashing in yellow). Note however, that is possible (particularly if the user aborts the router) that you can have a very large numbers of conflicts early on during the routing process and trying to sort out such a tangle in ARES is more or less impossible.

Command Buttons

The Begin Routing button is the button that invokes the router and commences the auto-routing of the board. Two additional buttons for exporting a design file and importing a session file are provided for users with the SPECCTRA™ Autorouter, whilst the Reset to Defaults button will reset all the configurable options to their initial values at installation. Finally, the Cancel button can be used to return to the ARES Editor without invoking the autorouter.

BASIC OPERATION

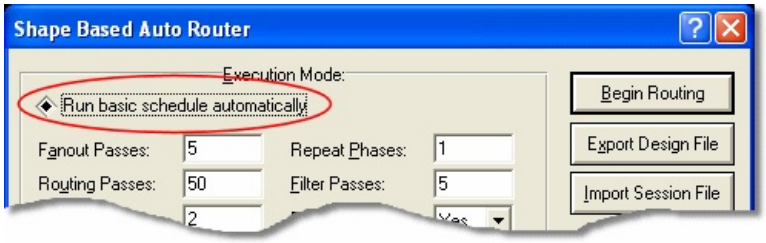
Overview

While the ARES Router is both complex and flexible it is also extremely easy to use. In this topic we look at some basic examples of how we might invoke the router in the different execution modes. Please see the appropriate sections of the documentation if you need more information on controlling router operation

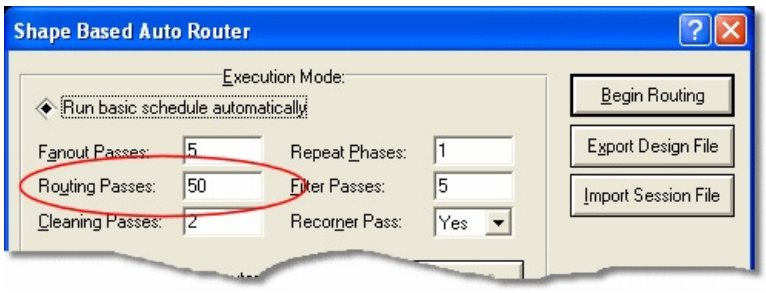
To invoke the router in fully automated mode

1. Launch the Auto-Router Command from the Tools Menu in ARES.

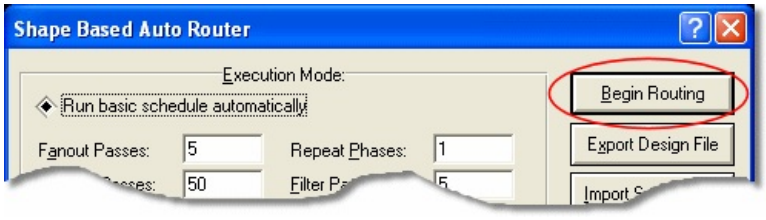
2. Select the Run Basic Schedule Automatically radio button as the Execution Mode.



3. If desired, adjust the number of passes from the default values.



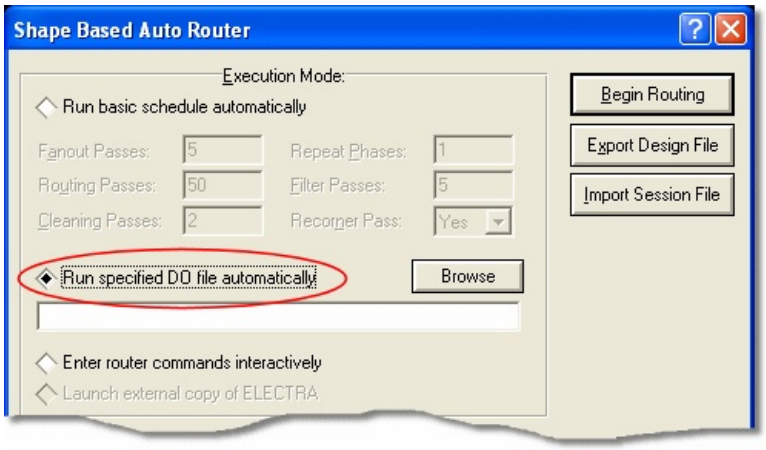
4. Use the Begin Routing button to autoroute your board.



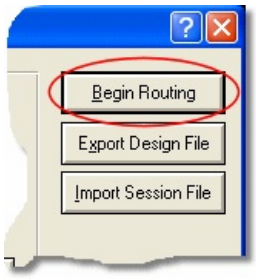
To invoke the router in scriptable mode

1. Launch the Auto-Router Command from the Tools Menu in ARES.

2. Select the Run Specified DO file Automatically radio button as the Execution Mode.



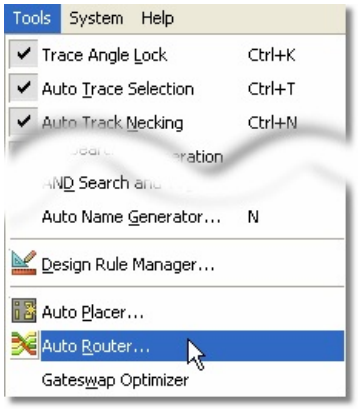
3. Browse for a router script (DO File) via the Browse button or type the location directly into the Edit Field.
4. Use the Begin Routing button to autoroute your board.



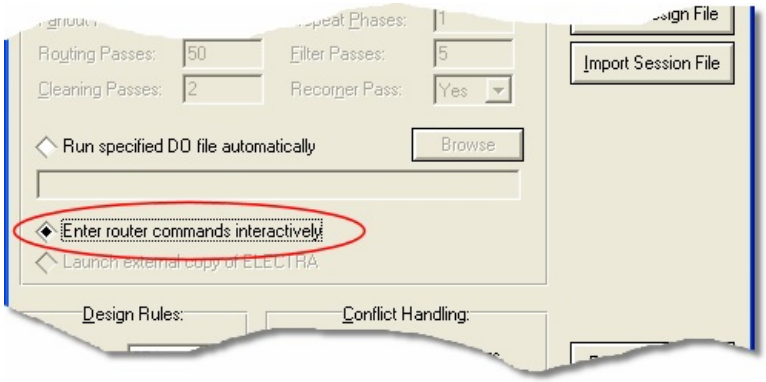
This option is only available with the advanced features set PCB Design Packages (PCB Design Level 2 and higher).

To invoke the router in interactive mode

1. Launch the Auto-Router Command from the Tools Menu in ARES.

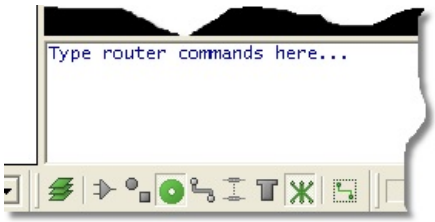


2. Select the Enter Router Commands Interactively radio button as the Execution Mode.



3. Use the Begin Routing button to autoroute your board.

4. A command window will appear at the bottom of ARES allowing you to type routing commands either globally or based on selected areas of the board.



This option is only available with the advanced features set PCB Design Packages (PCB Design Level 2 and higher).

CREATING ROUTING SCRIPTS

Overview

The ARES Router is capable of running a pre-supplied script and will schedule routing operations based on the commands in the script. This can be particularly useful if you find that a particular sequencing of commands helps to achieve completion or produces a better result and you want to use this sequence repeatedly for different projects. Finally, for very complex boards this method of execution is ideal where, for example, you wish to let the router find the best solution overnight or over a period of several hours.

Creating and Modifying Scripts

Labcenter provide a pre-supplied scripting file that, whilst basic, will successfully achieve completion on most standard PCB's. The intention is that this file be copied and then modified or extended as required to suit the needs of your project(s).

The contents of the pre-supplied basic script are copied below:

```
#==================
# BASIC DO file
#==================
bus diagonal
fanout 5
route 20
clean 2
route 25 16
clean 2
filter 5
recorner diagonal
```

Note that comments can be entered by preceding the line with the hash character.

More information on each of the commands used can be found in the command reference.

Once you have adjusted the script as required simply save it on your computer, specify the execution mode and location of the script via the Autoroute command on the Tools menu and invoke the router. This procedure is outline in more detail in the basic operation topic.

INTERACTIVE AUTO-ROUTING

Overview

Working with the router in interactive mode maximises the flexibility you have over the routing process. It allows you to easily route parts of the board gives you full control over the order and priority of routing tasks. You do however need to have an understanding of the basic command set and of the different routing phases in order to maximise the use of this mode of operation. It is therefore most likely to be of use in more complex designs or for users comfortable with directing the flow of the router.

Interactive mode is only available to users with an advanced feature set version of ARES (PCB Design Level 2 or higher)

Once you have invoked the autorouter in interactive mode you will see that a command window is launched at the bottom of the ARES Editing Window. It is important to understand that at this point you are in autorouting mode and therefore that any attempt to change the layout will exit the routing mode before applying your commands. Similarly, if you switch modes (for example, to component or track mode) then you are implicitly leaving autorouting mode and ARES will therefore remove the command window accordingly.

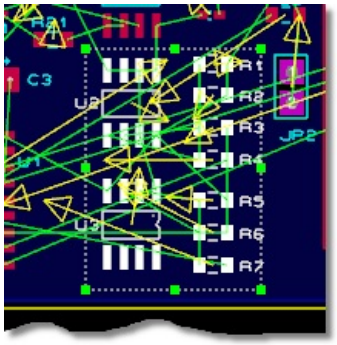
Basically, when you have entered the interactive routing mode you can:

- Type routing commands to control the autorouter.
- Use the mouse to select an area or net which you wish to route.
- Use the object selector to select a net.
- Use the mouse or keyboard to navigate the layout in the normal way.

Other operations will cause the routing mode to exit before taking place.

Specifying Nets

By default, any commands you type in the command prompt will action on the entire board. For example, if you type ‘route 20’ then twenty routing passes will occur on the entire board. It is quite common however to wish to route a particular area of the board (e.g. around a small pitch SMT footprint) prior to the global routing process. You can do this in ARES simply by dragging a selection box around the area of the board that you wish to route.



Selecting an area to route

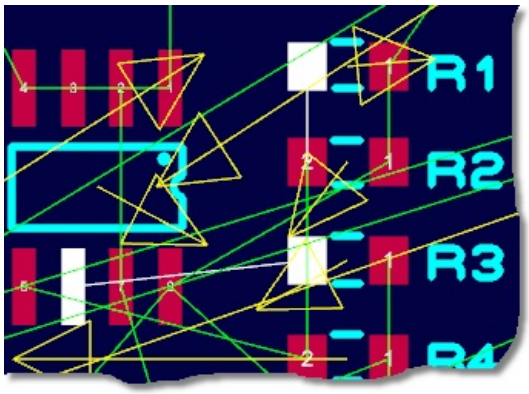
Having created the selection area you then have two options:

1. Limit the selection only to nets entirely enclosed by the selection area.
2. Extend the selection to all nets where a connection on that net is partially enclosed by the selection.

You can toggle between these two options via the track selection mode icon on the Selection Filter

Track selection mode

Alternatively, you may wish to limit the initial routing to affect only a specified net. You can do this by simply clicking the mouse on a pad on that net – this will highlight all of the ratsnest connections on that net and any subsequent routing commands will apply only to the selected objects.



Single net selection

If a selection exists the router commands will be limited to operating only on the selection. This means that, if you want to revert to routing globally, you need to click the mouse on an empty area of the layout to deselect all currently selected objects

Routing Phases

In interactive mode you drive the router by indicating what you want to route (via selection) and then typing commands to direct the routers progress. In order to successfully do the latter it will be useful to look at the three phases of a typical routing schedule.

Preparation

This is the first phase where commands are issued to handle particularly difficult routing objects. A good example here is the 'fanout' command that should be used before routing to make access to fine pitch SMT footprints easier.

Routing

This is obviously the main phase and we typically use both the 'route' and the 'clean' commands here. It is important to remember that the router is adaptive so if you don't reach completion repeat the process – you are not doing the same thing again!

Tidy Up

Once we are finished with the routing phase we enter commands to tidy up the board. This is where we might use the 'filter' command if we didn't reach completion and where we would use the 'recorner' command to minimise track lengths or avoid solder traps.

- A good example of the type of command set that you would use for a typical routing exercise is that used in the standard pre-supplied script which you can also find here
- Command Syntax and examples of usage can be found in the command reference.

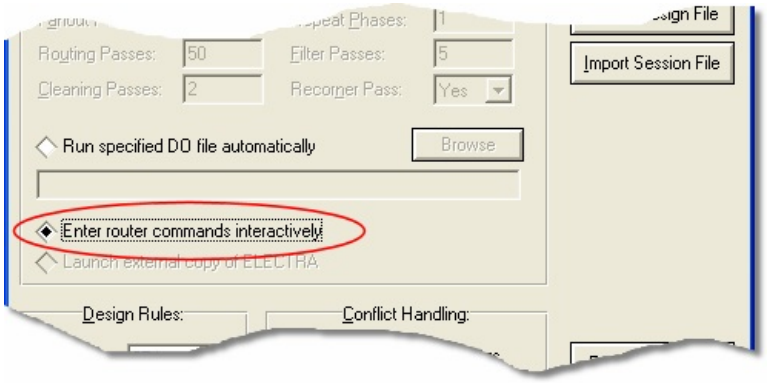
Example: Routing in Interactive Mode

This example describes how you would typically route a board in interactive mode, assuming that you have configured design rules, layer sets and net classes through the Design Rule

Manager.

To route a layout interactively

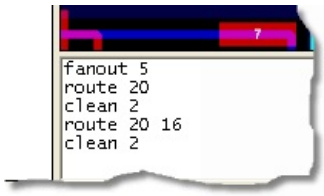
1. Launch the router in interactive mode.



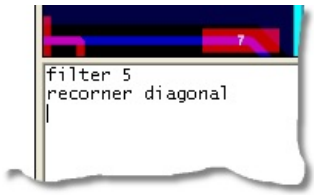
2. The preparation phase consists of the bus command followed by the fanout command to provide access to fine pitch SMT parts.

3. The initial routing phase then consists of the route command (single parameter for number of passes) followed by the clean command to tidy up.

4. Subsequent routing phases consist of the route command (two parameters - number of passes and start pass) followed by the clean command to tidy up.



5. Once we are finished routing the tidy up phase consists of the filter command (if we didn't achieve completion) and possibly the recorner command for solder trap avoidance.



The router is adaptive so re-using the same route commands in subsequent passes does not simply repeat the same routing process.

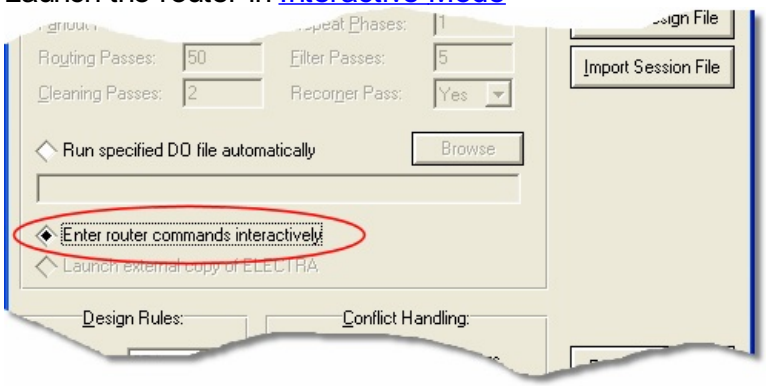
Example: Routing a Net on a Particular Layer

ARES provides a User Interface for this via the design rule manager (Technology menu) where you can specify the routing layers for any given net class. However, you can directly instruct the router to do this in interactive mode. The layers in ARES are identified for routing purposes as follows:

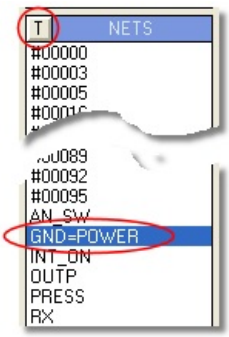
Layer Name	Designator		Layer Name	Designator
Top Copper	TOP		Inner 8	I8
Inner 1	I1		Inner 9	I9
Inner 2	I2		Inner 10	I10
Inner 3	I3		Inner 11	I11
Inner 4	I4		Inner 12	I12
Inner 5	I5		Inner 13	I13
Inner 6	I6		Inner 14	I14
Inner 7	I7		Bottom Copper	BOT

To route a net on a particular layer

1. Launch the router in Interactive Mode



2. Tag the net (e.g. GND) by selecting it in the parts bin and then clicking on the 'T' button at the top.



3. Use the cost command to forbid routing on one or more layers.

4. Use the route command to route the net on the remaining layer.

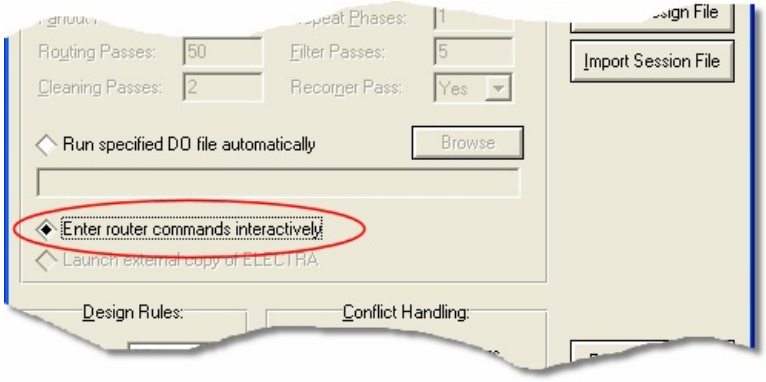
5. Use the cost command again with value -1 to reset to default.

Note that restricting the layers available for routing makes achieving completion much more difficult.

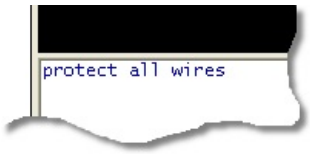
Example: Protecting existing tracking

To protect tracking when routing interactively.

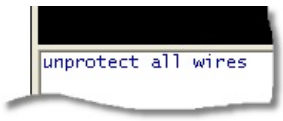
1. Launch the router in Interactive Mode



2. Use the protect command with 'all wires' prior to routing.



3. Use the unprotect command when you are finished.



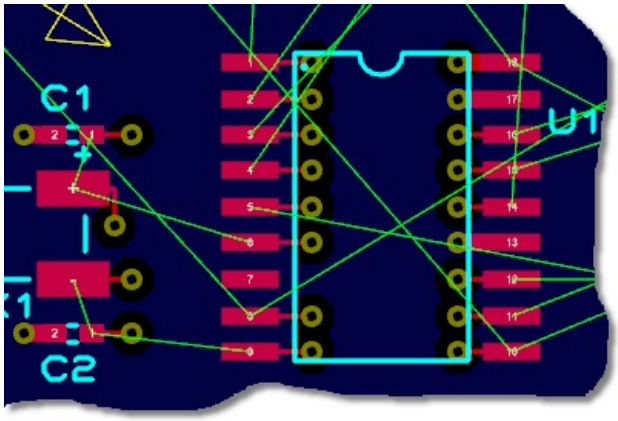
The lock command can be used instead of the protect command if you want to prevent the router connecting to tracks already placed as well as prevent the tracks being moved/deleted.

Example: Controlling Fanout on SMT Chips

Fanouts are escape wires from SMT pads which then allow the router to make connections. Typically, the fanout is one of the first commands to be executed when routing but interactive mode gives you far more control over what happens in this phase.

To execute the default fanout

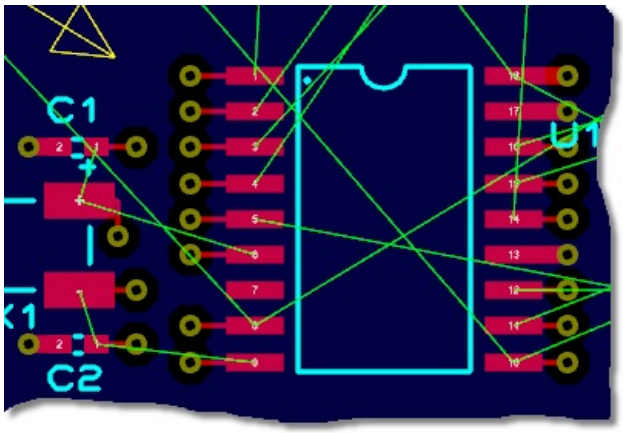
```
fanout 5
```



Fanout 5 where the vias are in side the IC

To force fanouts to be outside the IC

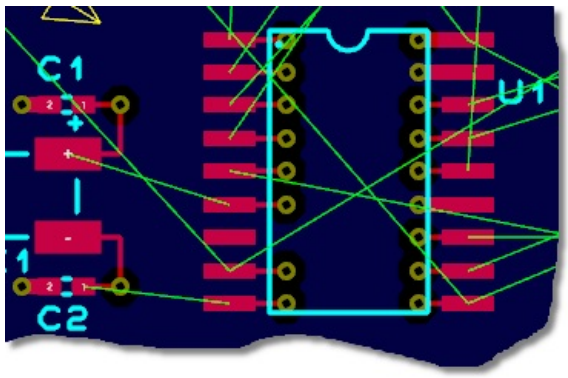
```
fanout 5 (direction out)
```



Now note that the vias are outside the IC

To allow via sharing on a fanout

```
fanout 5 (via_share on)
```

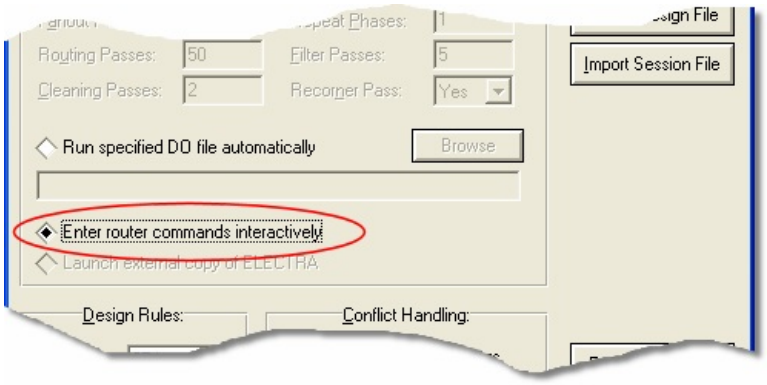


Note that C1 and X1 now share the same via

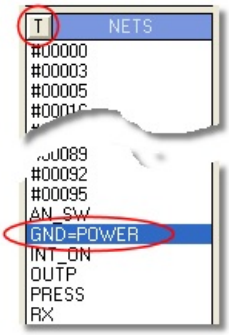
Example: Setting the Maximum Number of Vias for a Track

To specify a maximum of two vias to be used when routing the GND net.

1. Launch the router in Interactive Mode



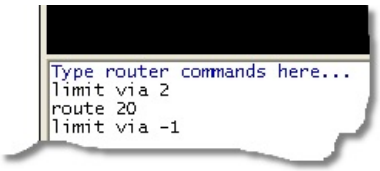
2. Tag the net (e.g. GND) by selecting it in the parts bin and then clicking on the 'T' button at the top.



3. Use the 'limit via 2' command to limit the number of vias

4. Route the net.

5. Reset to defaults by using -1 with the limit command.



COMMAND REFERENCE

The following is a description of the command set exposed to the user that will execute routing operations in ARES. These commands can be typed in directly when routing in interactive mode or used to construct a custom routing script when working in scriptable mode. Typically, you might only use the first six of these commands for interactive mode whereas you may well need to use some of the more advanced commands when in scripting modes. Each command includes a description of usage, the command syntax and an example.

The BUS Command

This command invokes a bus routing pass only on regular array of pins with collinear connections where the pins share a common X or Y coordinate. This is particularly effective on memory arrays. In this mode, the router will not generate conflicts, so rules must allow for sufficient space. By default traces are routed orthogonally, unless the diagonal option is specified. Typically, this command should be the first issued in a routing script.

Basic Syntax:

```
bus [diagonal]
```

Example:

```
bus diagonal
```

The FANOUT Command

This command autoroutes small escape wires with a via from SMD pads. Recommended for use at the start of routing where there are more than two signal layers. Not recommended for use with two layer boards as it will block routing spaces.

Typically, this command should be issued immediately after the bus command and prior to any route commands.

Basic Syntax:

```
fanout <number of passes>
```

Example:

```
fanout 5
```

The fanout direction can be also be set so that fanout vias are added inside SMD components and/or outside. Fanout can be limited by pin type, for example pins connected to power nets only.

Full Syntax:

```
fanout [<passes>]
[(direction[in_out|in|out])]
[(pin_share [on|off])] [(via_share [on|off])]
{[(pin_type [active|signal|power|unused|all|single])]}
[(max_len <positive_dimension>)]
```

Example: Depth for blind and buried vias

```
fanout (depth opposite 2) (share_len 500)
fanout 5 (pin_type signal) (via_share on)
```

Example: Fanout using a grid of 25

```
fanout [via_grid 25]
```

The ROUTE Command

This command tells the autorouter to perform a specified number of routing passes and is the basic command used when routing. Typically you would use this command iteratively with the clean command to acheive completion. The start_pass parameter should only be used on the second or subsequent routing phases and sets the initial costing of the first routing pass of that route set.

Basic Syntax:

```
route <number of passes>, <start pass>
```

Example:

```
route 20 (first routing phase)
route 20,16 (subsequent routing phases)
```

The CLEAN Command

The clean command will basically rip-up and reroute each net with high via cost and the highest cost for making clearance and crossing violation. This command is typically used iteratively with the route command perform some routing passes, clean, perform some more routing passes, clean again, etc. However, when full completion is reached the clean command can be used to minimise vias.

Basic Syntax:

```
clean <number of passes>
```

Example:

```
clean 2
```

The FILTER Command

At the end of a routing session you may end up with wires/vias in conflicts (crossing or too close to each other). The filter command will remove these conflicts by unrouting the minimum number of connections that are currently in conflict

Basic Syntax:

```
filter <number of passes>
```

Example:

```
filter 5
```

The RECORNER Command

The recorner command changes 90 degree wire corners to 135 degrees. It is performed on wire corners exiting pins and vias as well as bend and slant wire configuration. This command is used after completion is reached to reduce track length. This command is typically issued last.

Basic Syntax:

```
recorner [diagonal]
```

Example:

```
recorner diagonal
```

The CHECK Command

This command can be used to run a DRC (design rules check) and visually tag the violations. This is used in particular when a rule is changed. A check is automatically invoked after every routing pass. The total number of violations is shown on the status line and visual feedback is added to the layout view to indicate conflict locations.

Basic Syntax:

```
check
```

Example:

```
check
```

The CIRCUIT Command (Advanced)

This command is used to schedule the routing order priorities, specific vias to be used and allowed routing layers amongst nets and net classes.

Basic Syntax:

```
class <class_id> | net <net_id> circuit {<circuit_descriptor>}
<circuit_descriptor> ::=
[(priority <positive_integer> |
(use_via {<padstack_id>}) |
(use_layer {<layer_name>})]
```

The value of priority ranges from 0 to 255, the default is 10.

The `use_via` rule assigns one or more via padstack to a class or a net. If more than one padstack is defined, the autorouter will favor the smallest padstack in size.

The `use_layer` rule assigns routing layers where nets and classes must be routed. Note that the use_layer rule will override a layer unselection .

Example: Routing a net/class to specific layers

```
net sig1 (circuit (use_layer L1 L2))
class fast (circuit (use_layer M1 M2))
```

Example: Assigning Routing Priority

```
net sig1 (circuit(priority 200))
```

Example: Routing a defined class to specific layers and using specific vias

```
(class special net1 net2 net3 net4 net5
(circuit
(use_via via.VIA1__1_10.bv1_10 via.VIA3__4_5.bv4_5
 via.VIA3__6_7.bv6_7)
(use_layer SIGNAL_1 SIGNAL_3 SIGNAL_4 SIGNAL_5 SIGNAL_6 SIGNAL_2)
)
(rule
(width 0.007)
(clearance 0.007)
(clearance 0.001 (type via_via_same_net))
(clearance 0.001 (type smd_via_same_net))
)
)
```

The COST Command (Advanced)

This command sets the internally defined costs to a fixed value. By default, some of the cost values get internally modified during autorouting. It is not recommended to change cross and squeeze costing specifically.

Cost values can range from 0 to 100. A value of -1 will reset the cost value. Predefined cost description values can be used:

Cost	Value
Forbidden	100
High	50
Medium	25

Low	8
Free	0

The following costs can be set:

Option	Description
Cross	Crossing conflict
Squeeze	wire to wire clearance conflict
via	wire to via clearance conflict
way	cost of routing in non-preferred layer direction
off-grid	cost of routing off-grid if specified
off-center	cost of entering/exiting an SMD pad off center.
side exit	cost to exit SMD pads on long side
layer	If type is length it is the cost of using the layer; if type is way it is the cost of routing on non-preferred direction.

Basic Syntax:

```
cost <cost-type>
cost <cost_type> cost_descriptors>|[(type[length|way])] |-1]]
```

Examples:

```
cost layer TOP forbidden
cost layer I1 high (type way)
cost via high
```

The DIRECTION Command (Advanced)

Changes preferred routing direction by layer.

Basic Syntax :

```
direction <layer name> [horizontal|vertical|orthogonal|off]
```

The GRID Command (Advanced)

Specifies wire and via grid spacing.

Basic Syntax:

```
grid [via<positive_value>[<via_id>] |
wire<positive_value>[<layer_name>]
```

Examples:

```
grid wire 8.333
grid wire 5 layer 1
```

The LIMIT Command (Advanced)

The limit command sets absolute limit values to be applied to each connection. Control is provided to limit maximum allowed number of intersecting wire, number of vias per connection, number of bends, and the maximum distance of non preferred (wrong-way) routing. The range of limit for <positive_integer> is 0 through 255. You can set limit values, perform some routing passes, and return to the default system values by executing a limit command with a value of –1. If you don’t supply limit values, computed default values are used by the autorouter.

Basic Syntax:

```
limit [cross [<positive_integer> | -1] |
```

```
via [<positive_integer> | -1] |
bend [<positive_integer> | -1] |
way [<positive_dimension> | -1]]
```

Examples:

```
limit via 2
limit way 200
limit way -1 (resets limit to default)
```

The LOCK Command (Advanced)

Prevents the router from changing or deleting locked wires. Note that the router is not allowed to connect to locked wires

Basic Syntax:

```
lock net <name>
```

The PROTECT Command (Advanced)

Prevents router from changing or deleting protected wiring. This option may be useful to protect pre-routed ground and power nets. Note that the router is still allowed to connect to protected wiring.

Basic Syntax:

```
protect all wires
```

The RULE Command (Advanced)

Rules can be set globally (PCB) or specifically to layers, net classes or nets. Note that in the general case this is handled by the design rule system in ARES so this command is only really applicable when working in scriptable execution mode.

There are two categories of rules type; clearances and wiring rules are as shown be the rule descriptor below:

Basic Syntax:

```
rule [pcb|layer<layer_name>|class <class_id>|
net<net_id>|]{<rule_descriptors>}

<rule_descriptor> ::=
[(clearance <positive_dimension>
[(type {<clearance_type>})])] |
[(junction_type [term_only | all])] |
[(limit_bends [<positive_integer | -1])] |
[(limit_crossing [<positive_integer | -1])] |
[(limit_vias [<positive_integer | -1])] |
[(limit_way [<positive_integer | -1])] |
[(max_total_vias [<positive_integer | -1])] |
[(reorder <positive_integer)] | [(tjunction [on | off])] |
[(via_at_smd [off | on [(grid [on | off])] [(fit [on | off])]] |
[(width <positive_dimension>)]
```

In ARES you should use the Design Rule Manager User Interface (Technology Menu) to specify rules and then let ARES communicate with the router to enforce them.

The SELECT Command (Advanced)

This command enables the autorouter to use specific layers and vias for routing. It will also allow for selecting a net for routing. Note that names are case sensitive.

Basic Syntax:

```
select[layer|via]{<id>}|[all[layers|vias]
 |comp{<comp_name>}|[net]{<net_name>}
```

The UNLOCK Command (Advanced)

Unlock existing tracks.

Basic Syntax :

```
unlock net <name> | [all {layers|vias}]
```

The UNPROTECT Command (Advanced)

Unprotect existing tracks and allows the router to reconsider them.

Basic Syntax :

```
unprotect all wires
```

The UNSELECT Command (Advanced)

Disallow the autorouter the usage of specific layers and vias for routing. This command can be used to reduce the number of layers from that originally specified. Note that names are case sensitive.

Basic Syntax :

```
unselect [layer|net|via]{<id>} | [all  {layers|vias}]
```

Example :

```
unselect layer s1 s2
```

HINTS & TIPS

Overview

Most auto-routing exercises proceed along the following lines:

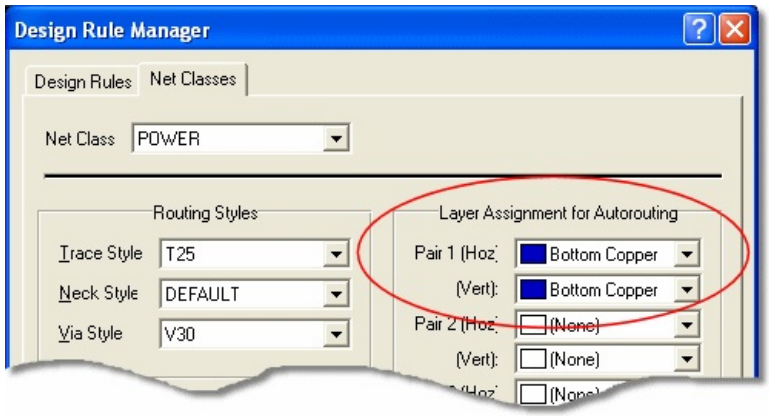
- Load the netlist and place some or all of the components.
- Set up design rules and configure net classes to define track and via styles for routing.
- Run the auto-router in fully automated (or scripted) mode.
- Unless complete routing occurs, examine why/where the routing has become congested and move components around accordingly. To remove tracking placed by the router, you can use the Delete icon in Route Placement mode.
- When you are satisfied with the component placement, either switch to interactive routing mode for complete control or repeat the fully automatic execution with a higher repeat count.

If you reach 95-99% completion and are struggling to fully complete routing you may want to investigate the problem connections. The easiest way to do this is to use the Connectivity Rules Checker on the Tools Menu in ARES. Double clicking on a list item will zoom the board to that connection, allowing you to easily investigate the problem area.

Apart from the above, the following sections form answers to commonly asked questions about using the auto-router.

Single Sided Boards

To route single sided boards, you need to configure the net classes so that all passes specify the single layer on which you want to route. This will force ARES to route both horizontal and vertical traces on the underside of the board.



Setting the POWER class to rout on a single layer

The number of iterations of the router for single sided boards will be considerably higher than that for boards with multiple signal layers (the removal of via capabilities vastly complicates the routing problem) but, with good component placement, completion of most boards can still be expected.

Routes which are not completed will be left displayed as ratsnest lines; some manual route editing will be required to convert them to be sensible wire links.

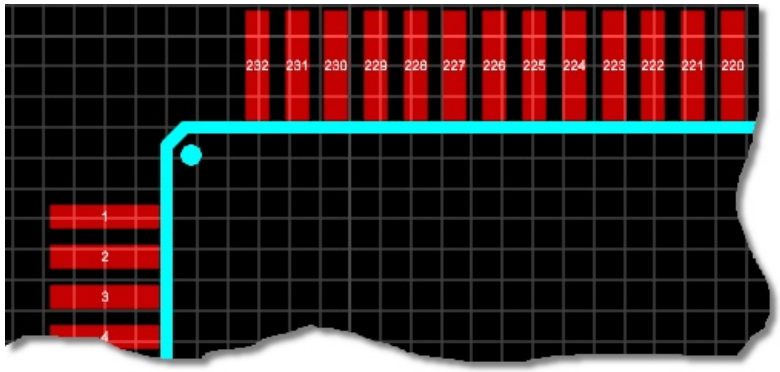
Avoiding Using Component Pads as Through Holes

For small scale prototype, hobby or educational production it can be useful to produce boards in which the pins of some components (e.g. ICs, electrolytic capacitors) are not used as through holes.

The way to achieve this is to redefine the component library parts to have pads on the Bottom Copper layer only. The router will then only route to these pads from the underside.

Surface Mount Components

Surface mount components, especially ICs, present something of a problem to a traditional grid based router because the pin spaces are rarely 25 or 50 thou. Thus tracks running on the grid cannot connect directly to the pads because the centres of the grid squares do not coincide with the pad centres.



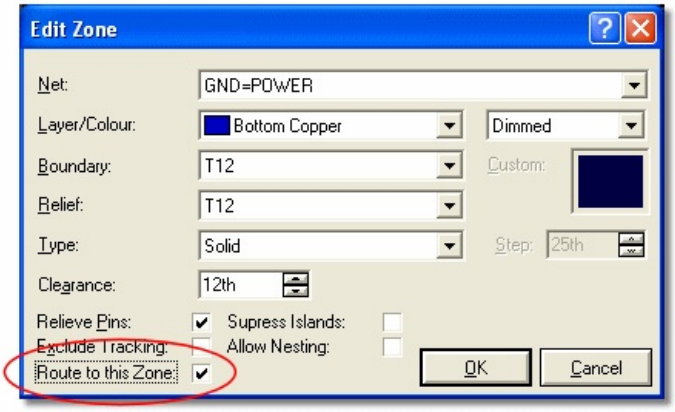
Notice that the centre of some of the pads are not on a grid point, PIN1 is, PIN2 isn't

To overcome this, you should always run one or more fanout passes before routing to maker it easier for the router to make connections during the routing phase.

You should also be careful regarding placement of objects such as decoupling capacitors – if you 'block the entrance' to the pads (and particularly if you are routing single sided) then the router clearly cannot make the connection

Routing to Power Planes

It is important that all power planes to which you wish to route are marked as 'route to this zone'. This option can be found on the edit zone dialogue form for the power plane and is enabled by default for planes generated via the Power Plane Generator command on the Tools Menu. Note however that split planes (planes generated via the zone icon – PCB Level 2 and higher required) do not have this option checked by default.



Check the Route to this zone box

Despite costing the crossing of a split plane highly, it is possible that a zone is splintered during routing by a track on a different net. This can usually be resolved by adjusting the layer set of a net class or by routing interactively where you have more control. Failing all else it may be necessary to correct this after routing or to simply place the zones after routing in the area concerned is complete.

INTRODUCTION & BACKGROUND

Overview

The way in which a PCB design program handles power planes has become much more important in recent years, mainly as a result of the new legislation on electromagnetic compatibility. There are at least three ways in which PCB software can implement power planes:

Grid Based Power Planes

Many mid-to-high end PCB design programs perform some kind of flood fill operation to produce power planes. Essentially, the target area is divided into squares and tests are made to see which squares can be filled with copper, and which are occupied by other objects.

This approach, although relatively straightforward to implement, suffers from a number of problems:

- Because a rectangular grid is used, any diagonal boundaries invariably come out in a staircase pattern, which is non-ideal for RF boards in particular. In addition, the power plane cannot 'squeeze through' narrow or non-orthogonal gaps and so optimum connectivity is not always achieved.
- If any of the pads to be connected to the ground plane are off grid, and if the ground plane is hatched, it is difficult to ensure that they are properly connected.
- In order to maintain the connectivity database, it is usual to represent the ground plane hatching as ordinary tracks. Not only does this use large amounts of memory and slow down the software, but it also makes it difficult to edit or remove the ground plane after it has been generated.

The last two problems could probably be overcome by sufficient coding but the first point is very intractable.

Negative Image Power Planes

Another approach is simply to draw enlarged copies of all the pads and tracks which are not to connect to the ground plane, and say that the result is a negative image of the power plane. Although used in some quite expensive software, this approach is badly flawed:

- The software cannot check if full connectivity is actually achieved - it only needs one track running right across the ground plane area to break it into two unconnected regions. If you are not allowed tracking on ground plane layers, then this usually forces the use of a multi-layer board.
- Even if connectivity is achieved, it is possible for 'slivers' to exist where the enlarged obstacle images nearly touch - perhaps leaving just 1 thou of copper between them.

Polygonal Gridless Power Planes

Although by far the hardest approach to implement, this scheme has none of the disadvantages of the other two and, so far as we know, no significant ones of its own either.

In essence, the idea builds on the negative image approach:

- The process starts by generating polygonal boundaries surrounding all the pads and tracks within the target area. We call these polygons Holes.
- The holes are 'added' together in the sense that where two or more overlap they are replaced by a single hole which closes them both. This process continues until a much

smaller number of many-edged holes remain.
- These holes are subtracted from the original boundary of the target area. This boundary is also a polygon, so you can have any shape of area you like filled with copper. It is at this stage that the software can find out if the connectivity is complete or otherwise. If a hole cuts right across the boundary, then the boundary is split into two separate regions of copper.
- Where holes are created by pads which are on the same net as the ground plane, the software checks to see if and how a thermal relief can be placed, and amends the connectivity database accordingly.

With this much implemented, there still remain two problems:

- Where two holes nearly touch, 'slivers' of copper can exist which give theoretical but unmanufacturable or unsatisfactory connectivity.
- If the power plane is to be drawn with a Gerber photoplotter or a pen-plotter, only a pen of some minimum thickness can be used to draw it. If such a pen is used to draw the boundary of a polygon, then it will actually end up enlarged by half a pen-width, and this could violate the design rules. In addition it is impossible to draw very pointed corners with a 'fat' pen.

Both these problems can be solved by starting out from the premise that all the boundaries will be drawn with a pen of specified thickness. Then, all the holes are computed as enlarged by half this pen width over the their nominal size. Thus the second point (above) is taken care of at the hole generation stage. But in addition, no polygon boundary can be less than a pen width in thickness and so no slivers are created either.

Ground Planes Without a Netlist

One issue that cropped up with annoying regularity in technical support for older versions of ARES was a requirement to produce a ground plane for a PCB which had been produced without a netlist. The problem with this is that of knowing which pads are actually connected to ground. In some older versions- which performed a flood fill from the pads on the specified net to generate a power plane - this problem appears more or less insurmountable.

Note that there is a selector checkbox on the Edit Pin dialogue form so that you can select individual pads to have thermal reliefs, or else connect solidly to the power plane.

USING POWER PLANES

Overview

There are two alternative ways of using the polygonal power plane functionality:

- Using the Power Plane Generator command on the Tools Menu.
- Using the Zone icon you can place rectangular or polygonal regions of copper which can form arbitrarily shaped ground planes occupying any chosen region of the board. This option requires Proteus PCB Design Level2 or higher.

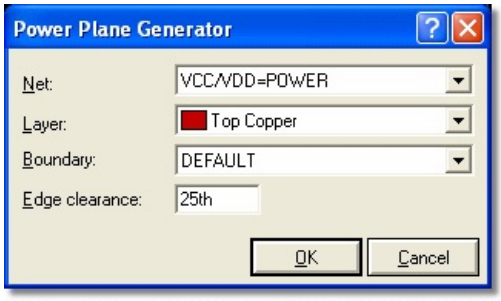
In both cases, the zones can be assigned to be hatched or solid, and computed with boundary tracks of specified thickness.

The Power Plane Generator Command

This command provides the simplest way to create a power plane and causes the generation of a single zone object occupying the entire area of a given layer of the board.

To use the Power Plane Generator command:

1. Choose a net for the power plane. If no net is specified, the power plane will connect to any pads which are marked for Thermal or Solid connection.
2. Choose a layer for the power plane.
3. Choose a boundary style for the power plane.
4. Specify a clearance from the board edge or use the default.



ARES will then generate the power plane, and update the ratsnest display to show the effect on connectivity.

Zone Placement Mode

The most flexible and powerful way to use the polygonal power plane functionality is to use the zone placement mode This option requires Proteus PCB Design Level2 or higher : This option is not enabled in ARES Lite.

To manually place a power plane:

1. Select the Zone icon.
2. Select the required layer using the Layer Selector.
3. Select the boundary trace style for the zone from the Object Selector.
4. Either:
   - Drag out a rectangular power plane by clicking and dragging from one corner to another with the left mouse button depressed.
   Or:
   - Mark a polygonal power plane by clicking left at each vertex. In this mode, you can also hold down the CTRL key to place curved sections of boundary.
5. ARES will then pop up the Edit Zone dialogue enabling you to choose a net and fill style

for the zone.

6. When you click OK, ARES will generate the power plane and update the ratsnest display to show its effect on connectivity.

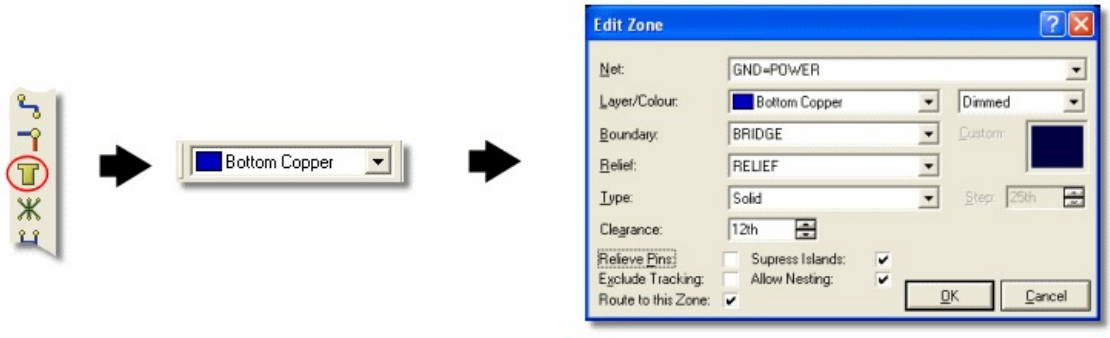
Regardless of the mode of zone creation you can increase the transparency of the zone (dimming) from the Edit Zone dialog as described below.

Editing a Power Plane

Power planes are held as zone objects which behave similarly to other objects in ARES.

To edit a zone:

1. Select the Zone icon.
2. Select the zone's layer using the Layer Selector.
3. Click right then left somewhere on the zone boundary to tag then edit the zone.



Editing a Power plane

Context Sensitive Help is available for all the fields within the Edit Zone dialogue form.

Deleting a Power Plane

This is very simple since the zone is a self contained object 'owning' all the copper regions within its boundary:

To delete a zone:

1. Select the Zone icon.
2. Select the zone's layer using the Layer Selector.
3. Click right twice somewhere on the zone boundary to delete the zone.

Automatic Regeneration of Power Planes

If you place, move or delete tracking or vias onto a board containing one or more power planes, ARES detects whether the power plane needs to be regenerated and then either:

- Proceeds to recompute the internal boundaries of the zone immediately. On a fast PC (e.g. Pentium 4) and a board of low to medium complexity this real time update mode of working can be quite viable with regen times of the order of 1 or 2 seconds. On Windows 98 and Windows NT, the regeneration of complex zones is run in the background such that you can carry on editing; the zone will redraw as soon as the regen process completes. For long and complex regenerations, ARES draws the zone in hatched form whilst its regeneration is pending. The Background Regen Threshold field on the Set Zones dialogue form determines the number of holes required in a zone for it to be deemed complex and thus processed using background regens.

or:

- Redraws the zone in hatched form to show that it is invalid. In this case, all such invalid zones can be regenerated by invoking the Regen command, key 'R'. This mode saves much frustration if the zone regen times are rather long, as will be the case on slower

PCs or with complex boards.

You can toggle auto regeneration on and off using the Auto Zone Regen command on the Tools menu, default keyboard shortcut key 'Z'.

Quick Redraw Mode

When a zone is drawn 'properly', all the external and internal polygon boundaries are drawn in tracking of the specified style. However, this can be quite time consuming and so a Quick Redraw option is provided in which

- The polygon boundaries are not drawn at all.
- Thermal relief segments are drawn as single pixel lines.
- If the zone is hatched, then the hatching lines are drawn as single pixel lines.

The quick redraw mode, and also the zone auto-regen feature are editable from the Set Zones command on the System menu. Both settings can be saved in the configuration file by invoking the Save Preferences command.

Quick redraw is disabled by default

Auto-Routing & Power Planes

In general, it is best to place power planes prior to auto-routing. There are two principal reasons:

1. The power plane will make connections to any through hole pads within itself, and this will remove the need for the autorouter to do so.
2. In the case of SMT components, the automatic via placement pass (PPGVIA) is activated by the presence of the zone(s), and so it is essential that they be placed prior to routing, and that their Route to this Zone checkboxes are enabled.

In cases where the power plane is on a layer shared with other tracking, the power plane may become split by tracking placed by the router. In this case, some ratsnest lines may be left at the end of the routing pass, and the best thing to do is to re-run the auto-router in order to complete these by conventional tracking.

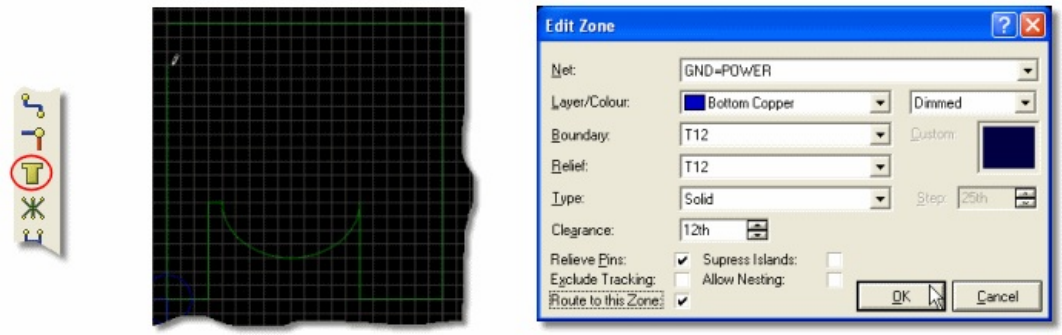
POWER PLANE EXAMPLES

Defining a Power Plane on Part of a Layer

This procedure requires PCB Design Level 2 or higher.

To define a partial power plane

1. Select the Zone Icon
2. Left click the mouse consecutively at each boundary corner of the area you wish the zone to cover (hold down the CTRL key to define a curved segment).
3. When complete, assign the zone to a net and configure from the resulting dialogue form



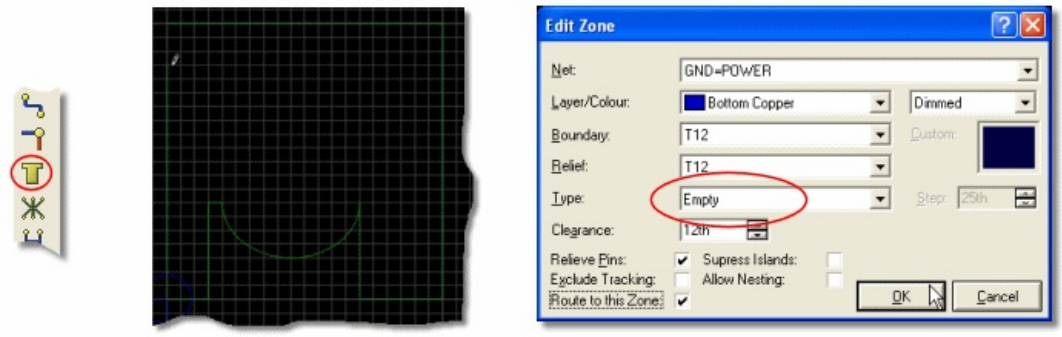
Placing a Power plane

Defining a Keepout Area inside a Power Plane

This procedure requires PCB Design Level 2 or higher.

To specify a keepout area inside a power plane

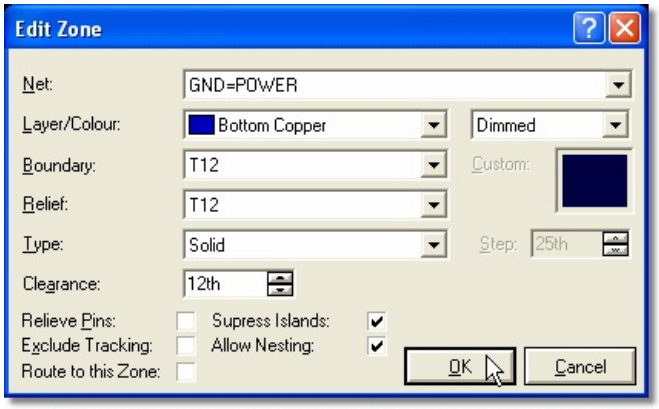
1. Select the Zone Icon
2. Left click the mouse consecutively at each boundary corner of the area you wish the keepout to cover.
3. When complete, set the zone type to be EMPTY from the resulting dialogue form



Setting the Zone type to be EMPTY

Controlling the Copper Flow on a Power Plane

The level of copper coverage is controllable via two options on the Edit Power Plane dialogue form, namely 'Allow Nesting' and 'Suppress Islands'.



Setting Supress Islands and Allowing Nesting of the zones

Allow Nesting

When selected the zone will 'jump' over obstacles such as tracks on other nets which would otherwise have prevented the copper flow from entering into an area.

Suppress Islands

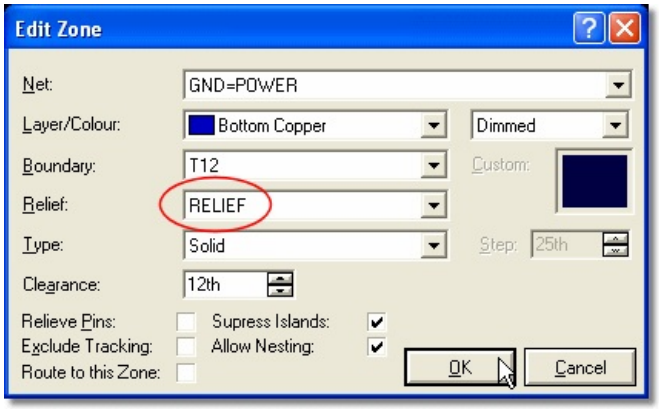
When selected all distinct areas of copper which do not connect to a pad on the specified net will be removed.

It follows that maximum zone coverage can be obtained by de-selecting the Suppress Islands option and enable the Allow Nesting option. However, a more common configuration would be to have both Allow Nesting and Suppress Islands enabled.

Specifying the thickness of Thermal Relief used in Power Plane Connections

To define the thermal relief for a power plane

1. Edit the power plane
2. Change the relief style combo box to the track thickness you wish to be used for relief connections.



Setting the Relief of the zone to be of style 'Relief'

Note that if the relief style is thicker than the width of a SMT pad then no connection will be made; otherwise there is a danger for design rule violations between the relief track and the surrounding pads on the IC.

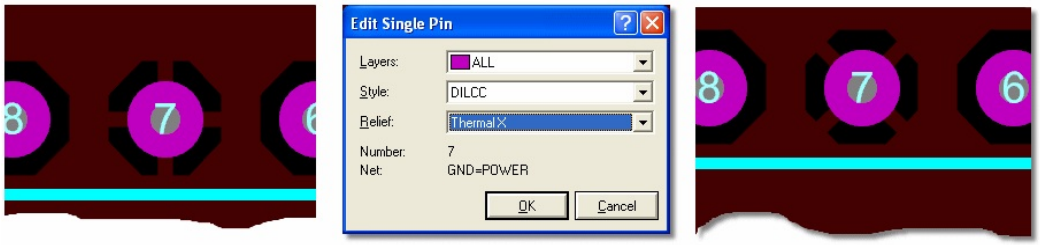
The relief style cannot be thicker than the boundary style otherwise the relief may 'stick out' and cause a short.

Changing the Topology of the Relief Stems on a Pad

It is sometimes useful to change the relief stems from '+' to 'x' in order to maximise connectivity to the zone.

To change the direction of the relief stems for a pad

1. Edit the pad on the layout
2. Change the relief field to 'Thermal X'.



Specifying the Clearance for the Power Plane

To define the thermal relief for a power plane

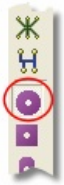
1. Edit the power plane
2. Change the clearance field to specify the clearance you want for objects not on the same net as the power plane.

Setting the clearance of a Power Plane

Stitching two Power Planes together

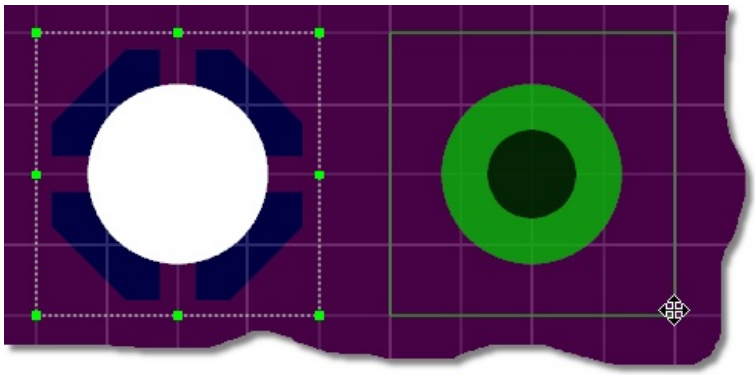
To stitch two power planes together

1. Select through hole pad mode and place a loose pad



2. Edit the pad and assign to the same net as the ground plane

3. Draw a tagbox around the pad and use the block copy command to duplicate as required.

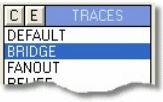


Joining two Power Planes at a Specific Point

To bridge two power planes at a specific point

1. Select track mode and then BRIDGE track style.

2. Place a track between the two zones. There is a pop up box that will warn you thatr the BRIDGE style of track is not electronically connected within ARES.



⚠ Note that the BRIDGE track style is - of necessity - outside of the netlist and therefore ARES will not warn you if you make a mistake. Use with caution.

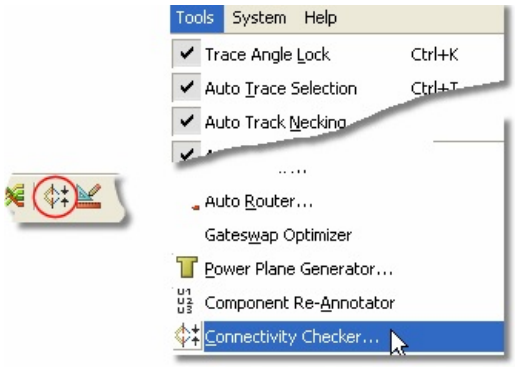
ℹ If you do not have a BRIDGE track style simply create your own trace style and name it BRIDGE.

CONNECTIVITY RULES

Overview

When you have completed routing a board it is useful to check whether it really corresponds with the netlist from which it was created. It is important to understand that ARES relies on exact correspondence between track ends and pad centres to establish connectivity, and that tracks which run onto pads, but do not terminate properly at there centres will show both as missing connections (because no connection is detected) and physical design rule errors (because a tracks is in contact with a pad, but not connected to it).

The Connectivity Rules Checker can be launched at any time from either the icon on the toolbar at the top of ARES or directly from the Tools Menu.



Launching the Connectivity Rules Checker

Basic CRC Functions

The CRC check shows:

- Each pair of pins joined by a ratsnest line is listed..
- Any extra (non specified) connections to each net are highlighted.

If you click on the entries in the listing, ARES will tag the associated tracking in a similar fashion to the Connectivity Highlight function.

Advanced CRC Functions

The CRC check also:

- Completely re-initializes the netlist management part of the layout database. Any floating tracking - that is tracking not connected to anything - is detached from netlist management whilst any detached tracking that has become connected to a net is brought under netlist management.
- Traces and vias involved in any extra connections are assigned to the VOID net. Further information on VOID tracking is given in The Netlist Loader & Existing Tracking.

DESIGN RULES

Overview

ARES has an in-built verification system for testing and warning you when the clearance between objects on the layout is less than required for your design. It is highly recommended that configuring and setting this up is one of the first things you do when laying out a new PCB; not only is it invaluable in avoiding errors but it is also harnessed by both the auto-routing and the manual routing engines to honour trace widths and clearance specifications. The system introduces four main concepts; net classes, design clearances, design rules and the design viewer.

Net Classes

A net class is simply a net or group of nets that have common electrical characteristics (e.g. Mains Power). Once specified, you can control trace widths, via styles and layer sets for a net class as well as controlling clearances by creating net class specific design rules.

> In older versions of Proteus a net class was termed as a strategy.

Design Clearances

There are a number of physical design clearances that can be configured to provide real-time checking during the board layout. Specifically, the following clearances are individually supported:

- Pad-Pad Clearance.
- Pad-Trace Clearance.
- Trace-Trace Clearance.
- Graphics-Net Clearance.
- Edge-Net Clearance.

Whilst the first three are self-explanatory, the last two clearances are worthy of further mention.

Graphics-Net clearance allows you to set the minimum clearance between a silkscreen graphic and any other object on the board. This is useful as the silkscreen graphic approximates the dimensions of the physical package.

Edge-Net clearance allows you to set the clearance between the board edge and any other object on the board.

Design Rules

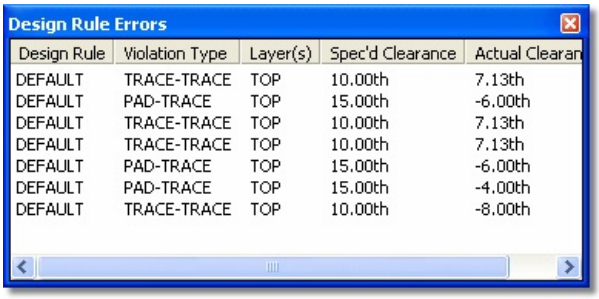
Design Rules are essentially configurable conditions under which the specified clearances are applied to the layout. In the simplest form this defaults to a set of clearances applied to all layers and all net classes on the layout; in other words, the manufacturing tolerances for the board. ARES however, is far more powerful than that and will allow you to specify rules on a highly granular basis as required. For example, you may wish a different set of clearances to apply to a particular layer, or to a given net class.

Design Viewer

The Design Viewer is a live display (located by default towards the bottom right of the ARES application) that will monitor and display the number of violations of the current design rule and clearance set specified for the layout.

The Live Design Rules Checker

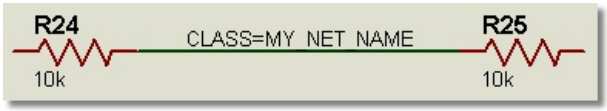
Clicking on the display will then launch a popup listing showing more detailed information, including the design rule in question, the layer, the violation type and the clearance information. If you click on an entry in the list, ARES will highlight the associated design rule flag on the layout, and if you double click an entry, ARES will both highlight it and zoom to its location on the board.



The Design Rule Checker form

Configuring Net Classes

When designing a PCB you invariably need to use different trace widths for different types of connections. You also almost certainly wish to have different clearances and may well want to route on different layers for different electrical 'blocks'. All of this is handled by designating groups of connections with the same electrical characteristics as a net class. This is done in the schematic by placing a wire label on any wire of a net with the syntax 'CLASS=xxxx' to denote that net as belonging to a named net class.



Labelling a wire with a Class

ARES will then pick up all named net classes and allow you to configure characteristics of the class via the Net Classes tab in the Design Rule Manager.

Settings for a particular net class will only have an effect on your layout if your net class is also named / specified it in the schematic capture package. You do this by adding a wire label of the form 'CLASS=xxx' in ISIS

Once you select the net class from the combo box at the top there are four main areas where you can configure options

Routing Styles

These options allow you to specify the trace style, neck style (for necking in to SMT connections) and the via style which will be used when routing the currently selected net class.

Via Type

Specify the via type for the currently selected net class. Obviously, this is tied in with the layer assignments for autorouting selectors such that if you select a top blind via and route from top copper to inner 1 the via is constrained to those layers. Equally, it is meaningless to specify a buried via with a routing pair of top copper and bottom copper!

Ratsnest Display

It is often useful – particularly with more complex designs – to specify a different colour of ratnest lines for different net classes to differentiate them on the board. Similarly, to reduce clutter you may wish to hide the ratsnest lines for a particular net class. These options allow you to do either or both.

Layer Assignment for Auto-Routing

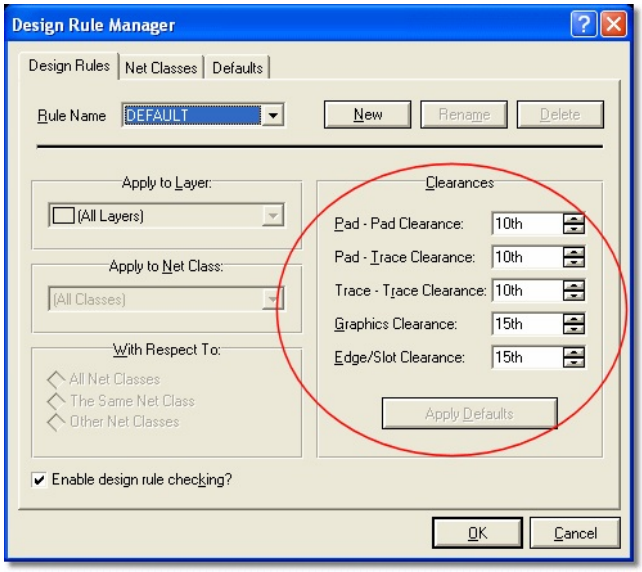
Each net class can involve up to 8 layers on which connections assigned to it will be routed by the autorouter. The (H) layer is for predominantly horizontal routes and the (V) layer for predominantly vertical routes. Single sided routing can be achieved by specifying the same layer for both H and V routes.

In addition to instructing the router where and with what styles to route the net class you can also enforce constraints in terms of clearances specific to that net class. This is done by adding and defining net class rules on the Design Rules tab of the dialogue form. This is discussed in more detail in the following topics.

Setting Design Clearances

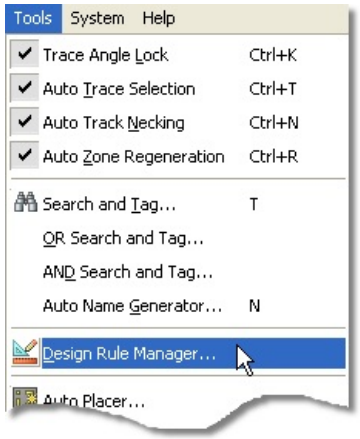
Design clearances can be set in two places; you can set the global default clearances or you can set clearances within a specific design rule.

The global design clearances can be thought of as the manufacturing tolerances for your layout and form the basis for the Default Design Rule; you can change these numbers, launch the Design Rule Manager from the *Technology* menu and set the rules under the DEFAULT Rule on the Design Rules tab.



The Default Design Rules form

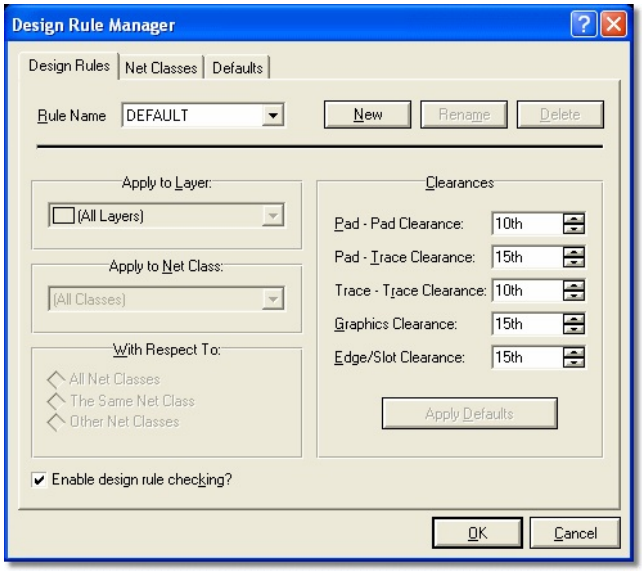
Often it is the case however, where your design requires different clearances based on location (design rules per layer), based on electrical considerations (design rules per net class) or a combination of both. ARES provides you with a method to specify any number of such design constraints via the Design Rule Manager command, located on the Technology menu.



Launch the DRM from the Technology menu

All such clearances are conditional and therefore form the basis of additional design rules. Put another way, you need to add a new design rule for each board requirement that differs from the manufacturing tolerances (the Default Design Rule). This is discussed in more detail in the next topic.

Setting Design Rules



Setting specific rules in the Design Rule Manager

The Design Rule Manager is the place where you can add additional rules specific to the current project. As can be seen from the screenshot above the dialogue has six areas for configuration.

Design Rules

This is where you create, rename and delete design rules or select an existing rule for editing. The Default rule is the global tolerance rule as discussed previously and is therefore permanent (cannot be deleted) and locked to all layers and all net classes.

Apply to Layer

After creating a new rule you can specify that the rule applies to all layers or to a specific layer of your choice.

Apply to Net Class

When creating a new rule you can specify that a design rule applies to a specific net class or to all net classes from this selector. Note that the default rule is locked to all net classes and that

only named net classes will appear in this selector.

For more information on defining net classes click here.

With Respect To

If you have defined a rule which applies to a specific net class you can then further refine the rule by specifying whether the rule applies only within the same net class, whether it applies only with respect to other net classes or whether it applies to both same net class objects and other net class objects.

Note that if you select either Same Net Class or Other Net Classes the graphics clearance and edge/slot clearance fields will be disabled as these objects are not specific to a net class and their meaning in such a case is ambiguous. These clearances are ignored for the purposes of such a rule.

Enable Design Rule Checking

This checkbox allows you to turn of the live checking for the currently selected rule. We recommend however that you leave this option selected.

Clearances

This is where you specify the required minimum clearances for the rule you are creating. The Apply Defaults button simply resets these clearances to those specified in the Global Rule (*Technology* menu - Design Rules).

Design Rule Hierarchy

Given that you can create several design rules that potentially 'overlap' each other, it is very important to understand how the design rule system works.

ARES supports a hierarchy of rule types in increasing order of precedence. A rule in a higher priority level of the hierarchy will take priority over an overlapping rule that is in a lower priority level of the hierarchy. Where two rules are in the same level of the hierarchy and they overlap the rule with the highest specified clearance in a given situation will 'win'.

The simplest way to understand this is that a more specific rule takes precedence over a more general rule; for example, a rule confined to a specific layer will take precedence on that layer over the default rule which governs the entire board.

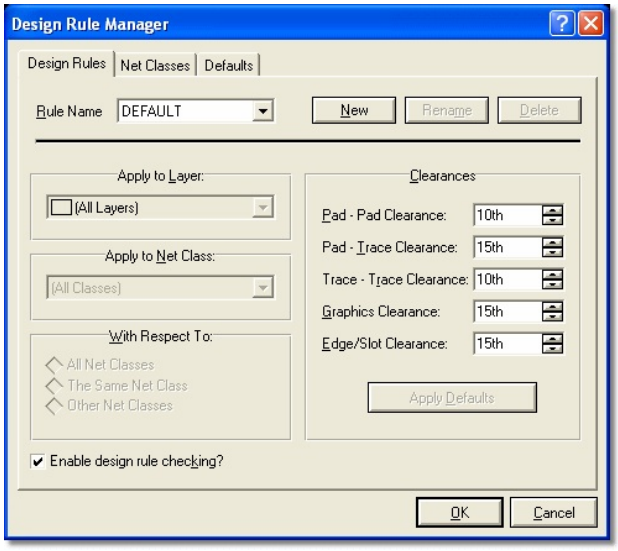
The following is an explanation of the rule types, starting from the lowest priority level of the rule hierarchy and working up to the highest priority level of the hierarchy.

PCB Rule

These are rules that apply to all layers on the board and to all electrical net classes. There is only one PCB rule and that rule is the pre-supplied default rule.

The default rule cannot be deleted for obvious reasons and cannot be changed to apply to specific layers or net classes.

The PCB rule is at the lowest level of the design rule priority tree, meaning that any additional design rules will take precedence on the layer or net classes that they affect.

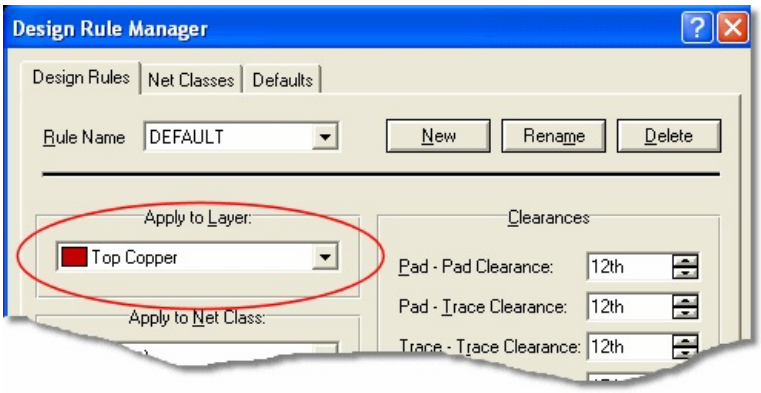


PCB Rule (Default setting)

Per Layer Rule

A Per Layer Rule is a rule with clearances that apply to all net classes but is confined to a single layer of the PCB.

Per Layer rules take precedence over the PCB rule but are lower in precedence to all other rule types.



Applying rules to a particular layer

Per Net Class Rule (All Layers)

A Per Net Class Rule is a rule with clearances that apply to a given net class on all layers of the PCB.

This rule will take precedence over both the PCB rule and any Per Layer Rules.

Applying rules to a all Nets

Per Net Class Rule (Per Layer)

This is a rule where the clearances apply to a specific net class only on a specific layer.

This type of rule will take precedence over all the aforementioned rule types and in particular will take precedence over a Per Net Class (All Layers) rule.

Applying rules to a particular Net

Inter Net Class (All Layer)

An Inter Net Class is a rule where the clearances apply to a specific net class but with respect to either itself or to all other net class. In other words, a rule such that the clearances apply only between traces and pads on the same net class or if you set up a rule where the clearances apply only between traces and pads on one net class against traces and pads on any other net class.

In this case, the Inter net class rule applies to all layers; this is the second highest priority rule.

Setting the DRC to be with respect to All Layers

Inter Net Class (Per Layer)

Exactly as above except that the inter net class rule applies to a specifically designated layer.

This is the highest priority rule and will take precedence over any other rule type that overlaps.

Setting the DRC to be with respect to per layer

While this may sound complicated it is in fact quite intuitive in use. We recommend you read through the examples provided in the section below to make sure you understand how the system works.

Design Rule Examples

This section of the documentation provides some examples to re-inforce the contents of the previous topics. In the following discussion assume that we have drawn the schematic, netlisted the design and now need to create a rule set to determine the design rule constraints of the board.

To create a rule governing clearances on top copper:

1. Invoke the Design Rule Manager command on the Technology menu.
2. Press the New Button at the top of the dialogue form and enter a name for the rule (e.g. TOPCOPPER).
3. Change the Layer Selector on the dialogue form to Top Copper.
4. Change the Clearances as required and hit OK to confirm and exit.

To create a rule governing clearances on the SIGNAL net class:

1. Invoke the Design Rule Manager command on the Technology menu.
2. Press the New Button at the top of the dialogue form and enter a name for the rule

(e.g. SIGNAL).

3. Change the Net Class Selector on the dialogue form to Signal.
4. Change the Clearances as required and hit OK to confirm and exit.

To create a rule governing clearances on the SIGNAL net class with respect to itself on top copper:

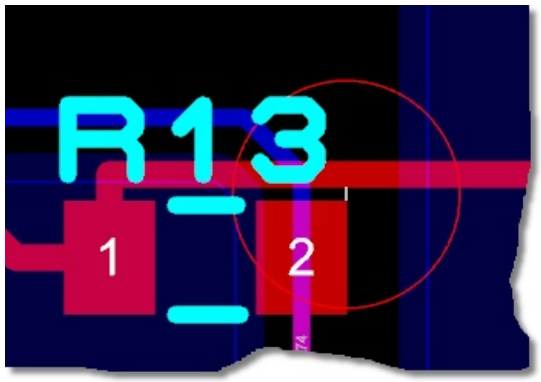
1. Invoke the Design Rule Manager command on the Technology menu.
2. Press the New Button at the top of the dialogue form and enter a name for the rule (e.g. SIGNALSIGNALTOP).
3. Change the Layer Selector on the dialogue form to Top Copper.
4. Change the Net Class Selector on the dialogue form to Signal.
5. Select the Same Net Class checkbox in the 'With Respect To' section of the dialogue.
6. Change the Clearances as required and hit OK to confirm and exit.

Analysis

We have now created three rules and have one default rule that exists for all designs. All four rules in play 'overlap' on the top copper layer so the behaviour of the design rule system for this board is worth further examination.

- Pads and traces on the Signal net class that are on the top copper layer will flag a design rule error should their clearance to other pads and traces on the Signal net class be less than those specified in the SIGNALSIGNALTOP rule.
- Pads and traces on the Signal net class that are on any layer apart from top copper will flag a design rule error should their clearance to any other object be less than those specified in the SIGNAL rule. On top copper a SIGNAL rule violation will occur should a pad or trace on the Signal net class be closer to a pad or trace on a different net class than the clearances specified in the SIGNAL rule.
- Objects on the top copper layer that are not assigned to the Signal net class will flag a DRC error if their clearance is less than that specified in the TOPCOPPER rule.
- Objects on any layer apart from top copper that are not assigned to the Signal net class will flag a DRC error if their clearance to another object not assigned to the Signal net class is less than that specified in the Default Rule.

In reality, the system is very intuitive; after all, you only create additional rules when you need to override the default behaviour for specific areas or electrical considerations. Additionally, when viewing the DRC errors both the rule that has triggered the error and the layer on which the error occurs is displayed for transparency.



DRC errors are also displayed on the layout

A good guideline when thinking about creating a rule set for a layout is to work down the design rule hierarchy, creating more specific rules as required after any less stringent rules have been created.

PRINTER OUTPUT

Overview

Output through standard Windows device drivers is performed using the Print command on the File menu whilst the device to print to may be selected using the Printer Setup command. Context sensitive help is available on all fields within these dialogue forms.

Specifying a Printer Device

You can specify the printer device in one of two ways :

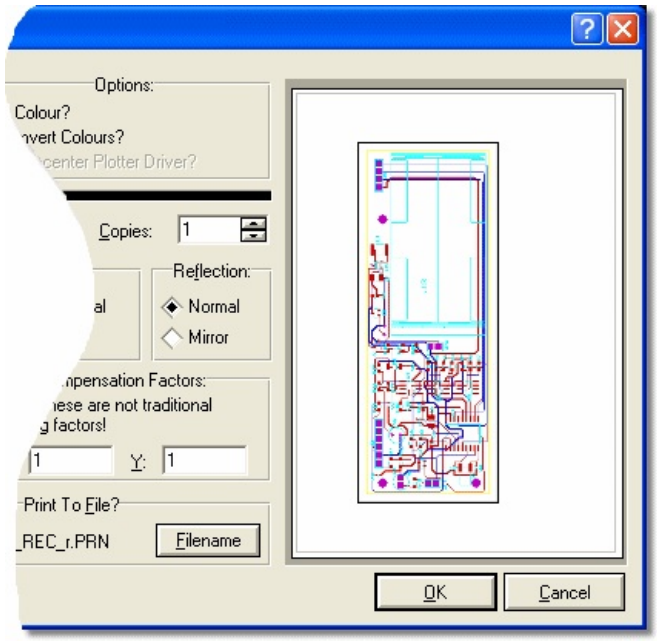
- Use the Printer Setup command from the Output Menu. All printer settings including the printer device specified will be remembered across ARES sessions and the printer device selected will be the default for future ARES sessions.
- Use the Printer button from the Print command on the Output Menu. Under these circumstances no printer settings will be retained. This is useful if you want to print to a different printer for a specific design.

Choosing a Colour Profile for Printing

The combo box in the general options allows you to choose a colour profile for the print job. Three colour profiles are provided as standard corresponding to Monochrome (black and white printing), Black Paper and White Paper. You can easily create your own colour profiles through the Displayed Layers dialogue form on the View Menu and these will then be available both for screen usage and in this dialogue form for printing.

Print Preview

The Print Preview appears on the far right of the Print dialogue form and displays the position and size of the schematic relative to the specified paper size. It will 'live update' as you configure the dialogue (scaling,colour,etc..) to your requirements.

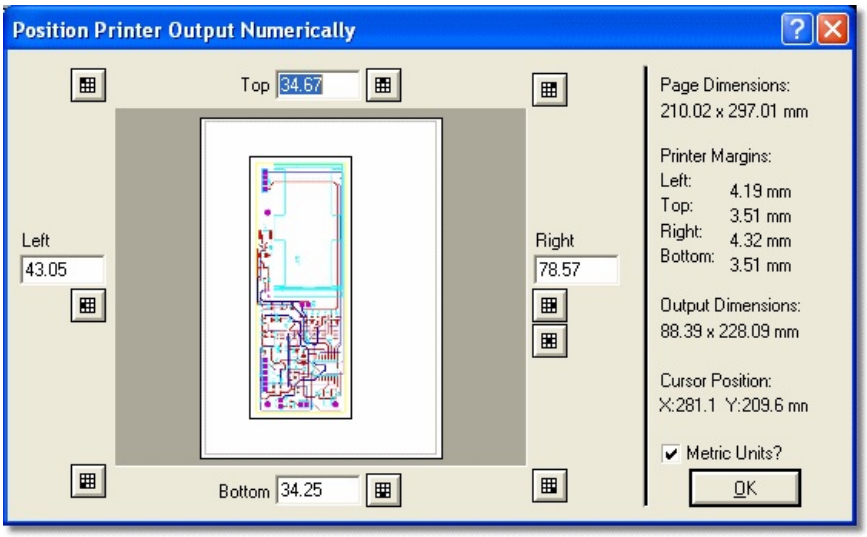


Print Preview in the Printer output dialogue for

You can adjust the position of the printed area by left depressing the mouse on the print preview and dragging to the desired position. Alternatively, you can choose a preconfigured position by right clicking the mouse on the preview pane and selecting the desired position from the context menu.

Configuring Margins

When not printing all sheets you can manually configure the position of the output within the printeable area. Right click on the Print Preview and select Position Output Numerically and enter the margins that you want between the edge of the printeable area and the edge of the output area. This is, in effect, an absolute method of positioning the output area within the printable area. In general, however, we would anticipate most users to position the output area via the mouse drag method detailed above. The graphical illustration below depicts the use of margins.



Setting the margins manually

If you turn off the option to Use Printer Margins on the context menu the margins you specify will be from the edge of the paper itself. In most cases this is not recommended as, depending on your printer driver, you are likely to lose part of your layout if you position it absolutely at the edge of the paper (outside of the printable area).

Printer Information

The Printer Information option on the Output Menu will dump a log of your printer details and capabilities. This is primarily for support purposes. Whilst our experience of printing support issues is that they are almost invariably the result of a defect in the printer driver we can make a better determination of the problem with the information detailed via this command.

PLOTTER OUTPUT

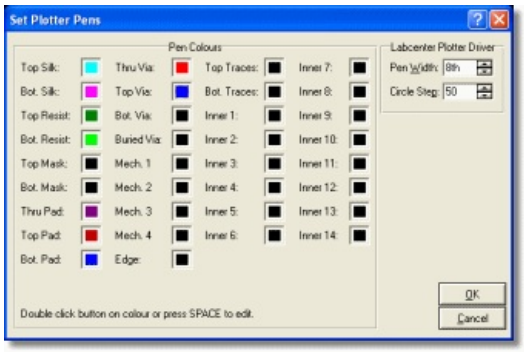
Overview

Windows support for Pen plotters is, unfortunately, very poor. Although drivers are supplied for HPGL and other plotters, the implementation of these drivers is very sketchy. Better drivers may be available for particular plotters, but we have not relied on this in designing the plotter support within ARES for Windows. Instead, we rely on the Windows plotter driver only to draw straight lines and then ARES itself does the rest. To do this, it uses the plotter driver module from the DOS version. This module allows control of the following options which are adjusted using the Set Plotter command on the System menu.

Plotter Pen Colours

The plotter drivers are able to generate a multi-coloured plot when ALL layers are selected on the Print dialogue form.

The Set Plotter command on the System menu is used to determine the colours for each layer. Where through hole pads are encountered, the pen number is determined by ORing the colours for the outer layers of the pad in the same way as is done for screen colours.



Note that this only applies to an ALL layers plot - single layer plots always use the black pen.

Unfortunately, the Windows plotter drivers do not (as far as we have been able to find out) support the direct selection of plotter pens by number. Therefore it is up to you to find out how your plotter driver maps colours onto pens and to select appropriate colours on the Set Plotter Pens dialogue.

Plotter Tips

On the Set Plotter command form are fields for Pen Width and Circle Step.

Pen width

This is a value in thou which should correspond (at least initially) to the width of pen you are using.

Circle step

This value is the size of step (in thou) that ARES will use when plotting the outer radius of a 1 inch diameter pad. A smaller number results in faster but more jagged circular pads whilst a value of 0 will instruct ARES to use the plotters internal circle drawing command. Whilst this is generally fast and smooth, it does mean that the pen is picked up and put down for each pass round the pad.

Getting high quality plots will require experiment with pens, film, ink, and the settings described above. You may also find it necessary to tweak the sizes of some of the pad, via and trace styles the problem is that there are too many variables outside our control to guarantee that

the dimensions of plotted objects will accurately tie up with their nominal size. As a starting point, we recommend a 0.25mm or 0.30mm tungsten carbide tipped pen, professional drafting film (obtainable from a good stationers) and Marsplot 747 ink.

Finally, the Line Width command on the Text Style command form is ignored when plotting a single pass of the pen is used for all silk screen graphics for reasons of speed. If you need a different line thickness, use a thicker pen for your silk screen plots!

POSTSCRIPT OUTPUT

Overview

Postscript, invented by Adobe Inc, is most commonly used in professional DTP and typesetting. However, it is an extremely flexible language well suited to accurately specifying a PCB artwork. More importantly, the £10K photo typesetters which are used by professional publishers are capable of producing >1200dpi images on film. Access to these machines is facilitated by the existence of bureaux and our experience is that they charge less than Gerber Photoplotting bureaux for an equivalent plotted area. Try calling a few local printers from the Yellow Pages to track down a local bureau.

Those of you with a Postscript laser printer will be able to proof your boards on it before generating a file for photo typesetting.

With ARES for Windows, you just select and install the appropriate Postscript printer driver, and use the Print command in the normal way. If you use a local phototypesetting bureau, they should be able to supply you with the correct driver for their phototypesetter.

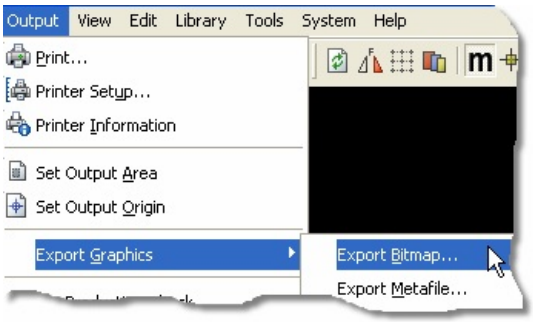
GRAPHICS FILE GENERATION

Overview

As well as printing directly to Windows print devices, ARES for Windows can generate output for use by other graphics applications. You have the choice of generating this output as either a Bitmap or a Windows Metafile, and you can transfer the output to the other applications either through the clipboard, or by saving it to a disk file. Context Sensitive Help is provided for the fields on all dialogue forms.

Bitmap Generation

The Export Bitmap command on the Output menu will create a bitmap of the board and place it either on the clipboard or in a disk file.

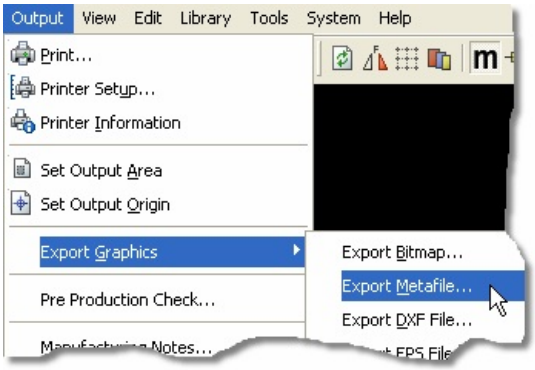


Exporting a Bitmap

Metafile Generation

The Windows Metafile format has the advantage of being truly scalable where a bitmap is not. However, not all Windows applications (e.g. Paintbrush) can read a metafile.

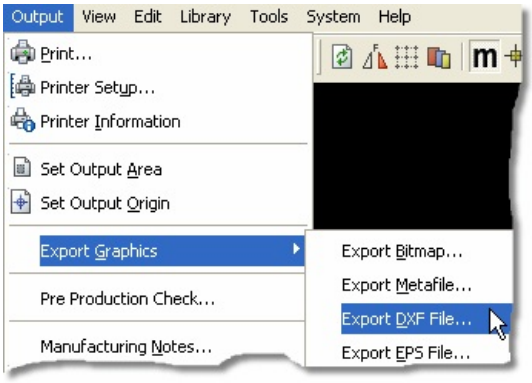
The Export Metafile command on the Output menu will create a metafile of the board and place it either on the clipboard or in a disk file.



DXF File Generation

The DXF format can be used to transfer output to DOS based mechanical CAD applications (it is better to use a clipboard metafile to transfer to Windows based CAD programs). The file is generated by a Labcenter output formatter, rather than by Windows.

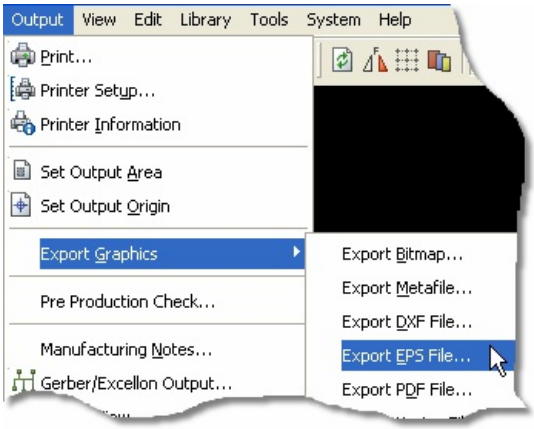
Our current experience is of considerable incompatibility and disagreement between applications on what constitutes a valid DXF file. To put this another way, given 6 applications supporting DXF, only about 30% of file exchange pairings seem to work! For Windows work, the Clipboard provides a much more reliable transfer medium.



Exporting DXF format

EPS File Generation

An EPS file is a form of Postscript file that can be embedded in another document. Although popular in the world of DTP, for Windows based DTP work you are much better off transferring graphics using a clipboard metafile.

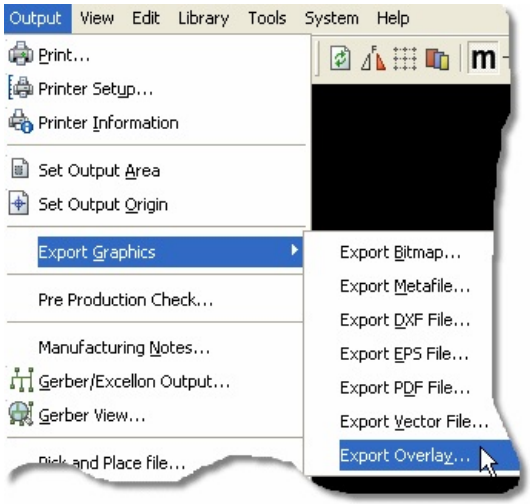


Exporting EPD Format

Overlay Bitmap Generation

The Export Overlay command on the Output menu will create an overlay diagram and place it as a bitmap either on the clipboard or in a disk file.

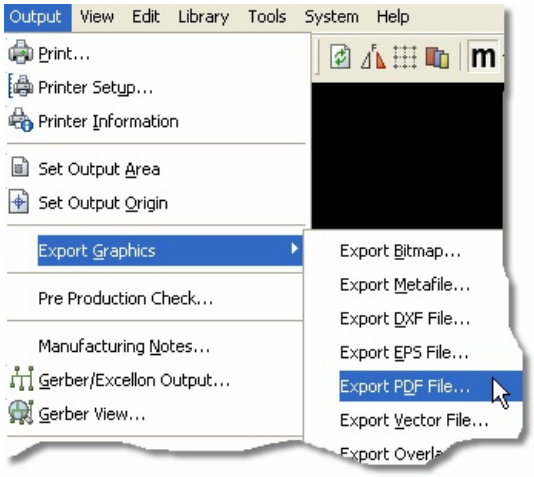
The overlay image is created by first rendering the copper layers in a ‘tint’ and then superposing one or more silk screen layers on top of this. The effect is to show the positions of the components with the tracking faintly visible underneath.



Exporting Overlay Format

PDF Generation

PDF files are widely recognised and adopted as a standard format for file sharing. ARES can output a PDF of the layout via the Output Menu where you can select layers, specify overlay or seperate pages per layer and either colour or monochrome output.



Exporting PDF format

PRE-PRODUCTION CHECKLIST

Overview

The pre-production checklist is an automatic test of your layout for common errors which is normally launched prior to sending the board for manufacture (it can also be launched manually from the Output Menu). We strongly recommend that you resolve any errors that are reported in this check before production of any hardware.

⚠ The pre-production check is an aid to and not a replacement for manual inspection of the layout and designer approval. Quality assurance responsibilities lie with the board designer – in particular Labcenter always recommend the creation and testing of a physical prototype prior to mass production.

The following is a list of the checks performed and suggested course of action when errors are reported:

Connectivity

This test automatically runs the CRC, checks and identifies that all connections specified in the netlist have been made.

If this fails, you should run the Connectivity Checker from the Tools Menu and double click on entries to locate and then track the missing connections on the layout.

Object Validity Check

This check tests the integrity of the layout objects and should always pass.

If this fails you should contact Labcenter support.

DRC Check

This tests re-runs the Design Rule Checker on the layout.

If the Design Rule Checker is disabled you should enable it from the Design Rule Manager on the Technology Menu.

If you have Design Rule Errors they can be viewed by clicking on the DRC Error report on the status bar at the bottom of the application. Errors can be located by double clicking the mouse on the resulting dialogue form.

Design rules are configured via the Design Rule Manager on the Technology Menu.

You should resolve all DRC errors before moving to manufacture. This may involve setting appropriate design rule for your layout (if you have not already done so), adjusting the layout to conform with the specified design rules or –very rarely – acknowledging the error as an intentional design decision and suppressing it by right clicking and selecting ‘Ignore this Error’ from the resulting context menu.

Zone Overlap Check

This check tests all the power planes on the layout for overlap. Importantly it uses a completely separate algorithm (code path) from the zone placement code in general use and therefore forms a secondary verification on zone integrity.

This test should never fail. If a fail is reported please contact Labcenter support without changing the layout

Unplaced Components Check

This simply checks that all the components specified in the netlist have been placed on the board. If it fails you need to place the components or change the schematic and re-netlist if they are superfluous.

Board Edge Check

This checks that the board edge is present and complete (no gaps). If this fails you should either place a board edge or rework the existing board edge so that it is complete.

Complex board edges in particular should **always** be drawn with the 2D Graphics Path tool as this guarantees a complete board edge. Curved segments can be drawn by holding down the CTRL key while placing the segment and the outline can be adjusted after placement via the drag handles. Please see the section on the board edge in the tutorial for more information.

Components Outputs Board Edge Check

This checks that components are all inside the board edge and not placed outside and then forgotten about.

Fixing this error involved re-working the layout such that all components are inside the board edge

Via Validation Check

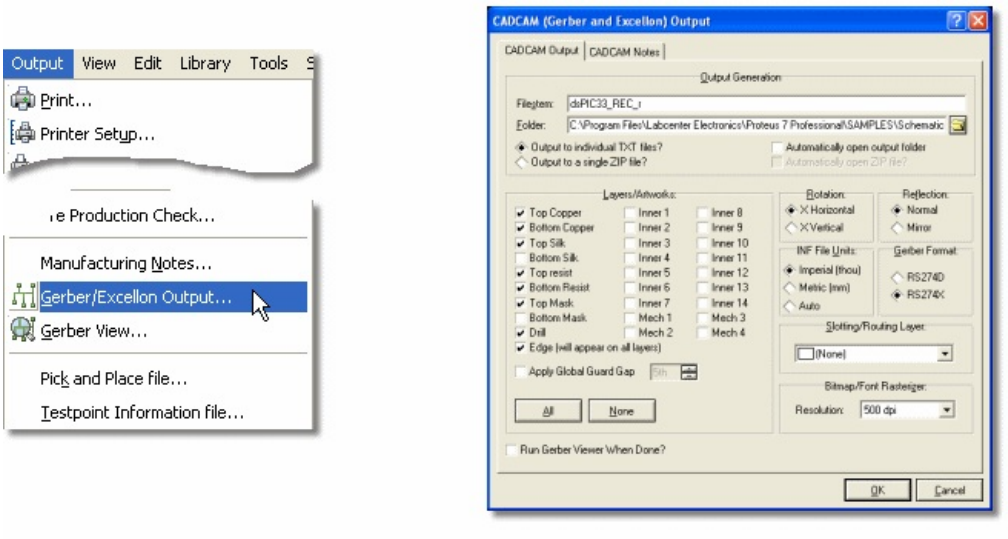
This checks that valid and contiguous via ranges have been specified for the vias on the layout.

If you get an error here you need to edit your via styles and check the layer ranges.

CADCAM OUTPUT COMMAND

Overview

All CADCAM output - Gerber files, NC Drill and Tool Information is generated from a single command.



Launch the CADCAM form from the Output menu

The dialogue form allows you to select:

- A common filestem and location for all files produced. This defaults to the layout filestem but a neat trick is to change it to something like C:\GERBER\MYBOARD such that all files produced will go directly into a location of your choice. You can, should you desire, set a default directory for CADCAM Output files via the System-Set Paths command.
- Whether you want all the outputted files to be collated in a single zip archive or left as a collection of individual files in the folder of your choice. Depending on your choice you can then opt to automatically launch windows explorer or the zip archive on completion, allowing you to browse the CADCAM files.
- Which layers you wish to output for the current design. This will be defaulted automatically based on an analysis of what is contained in the layout.
- Whether the screen X co-ordinate is directed to the CADCAM x co-ordinate (X-Horizontal) or to the CADCAM y co-ordinate (X-Vertical).
- Whether the newer RS274X or the older RS274D Gerber format is used. RS274X has the major advantage that aperture information is embedded within the files and does not have to be manually entered by the photoplotting bureau.
- Which layer is used for mechanical routing data.
- What resolution is used for rastering polygons, truetype fonts, and bitmaps. A finer resolution will result in smoother edges but can result in the generation of very large files.
- A Zip File comment. Zip archives allow a comment to be appended to them and this comment appears by default when you (or your board manufacturer) opens the zip archive. This is a useful way to 'sign' your archives or to provide special instructions etc. for your board manufacturer.
- A Tool File comment. This comment, if specified, allows you to add contact information etc. to the readme file generated with the CADCAM output. This readme file also lists all

files generated together with the tool settings for the photoplotter and the drilling machine.

- A comment for manufacturing notes. This comment, if specified, allows you to add design specific manufacturing notes to the readme file generated with the CADCAM output.
- Whether or not to launch the Gerber Viewer automatically on production of the CADCAM files. This option will load the CADCAM files into the Gerber Viewer on completion, allowing you to view the resulting output prior to sending the files to your PCB manufacturer for prototyping.

Any files generated from a previous run but now corresponding to a disabled layer are deleted - the new tool information table may be wrong for them.

Finally, context sensitive help is available for all options on the dialogue form, providing detailed information on the functionality of each field..

GERBER OUTPUT

Overview

The Gerber format, named after Gerber Scientific Instruments Inc., is almost universal in the PCB industry as regards specifying PCB artwork.

A photoplotter is essentially the same as a pen plotter except that it writes with a light beam on photographic film rather than with an ink pen on paper. The aperture through which the beam shines can be varied, allowing it to expose a pad in a single flash and a track with just one movement of the plotting head.

In order for a photoplot to be produced (usually by a bureau, as photoplotters are not cheap) it is first necessary to set the machine up with a set of apertures corresponding to the various pad shapes and track widths used on the board. Each different aperture is referenced by a so called D Code in the Gerber file and a table listing D Codes against aperture shape and size must thus be compiled. This table is compiled automatically and then written out as part of the tool information file.

ARES is clever enough to tell if two pad styles result in the same aperture, and it will only use one D code in this situation. Also, styles which are present in the selectors but not used on the board do not waste table space.

Other points to bear in mind:

- An aperture cannot render the image of a pad with a centre hole the piece of metal for the hole would fall out! Photoplots thus come out with completely solid pads and boards produced from them are very difficult to drill manually.
- Rectangular pads that are rotated to non-orthogonal angles have to be rendered by hatching them with a fine line.
- Solid polygonal ground planes have to stroked by the driver and the resulting output files can be massive. However, the boundaries will be drawn exactly as computed - using an aperture corresponding to the boundary thickness for the zone - so there is no risk of design rule violations.
- The origin for the Gerber co-ordinate system is defined by the Set Output Origin command on the Output menu.

NC DRILL OUTPUT

Overview

Many PCB manufacturers now have Numerically Controlled drilling machines which provided with the positions and drill tools to use for each pad, can automatically drill a circuit board. Nearly all these machines use the Excellon format, invented by Excellon Industries.

The format is very simple, consisting of the (x,y) locations of the holes and T Codes specifying which tools to use. As with Gerber Format, a table is created within the Tool Information File that lists the hole diameter corresponding with each T Code. ARES handles this in the same way as it handles each Gerber D Code it assigns a new T Code for each new hole size it encounters, and this information is also put in the tool information file.

ARES also embeds the basic tool size information within the Excellon Drill file; this will be sufficient provided that you are not creating a board with complex drilling requirements such as blind/buried vias or a mix of PTH and non PTH holes.

ARES uses the same co-ordinate systems for Gerber and NC Drill output - based on the position of the Output Origin - so your manufacturer should have no problem in aligning the drilling machine with the artwork.

⚠ The drill sizes used are taken from the Drill Hole attributes of the pad styles. Please make sure that the default sizes we have allocated are suitable for your application before having large numbers of boards manufactured.

MECHANICAL ROUTING & SLOTS

Overview

There are two sets of circumstances in which it may be necessary to form make a non-circular hole within a board:

- Some components have solder lugs rather than round pins, and these are best mounted on a pad which has a slotted rather than a drilled hole.

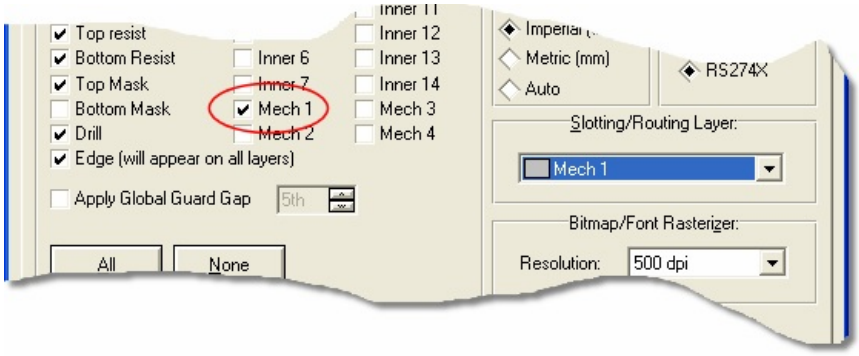
This can be achieved by defining a slotted pad-stack. See pad stacks for more information.

- Sometimes there is a need to make a large cut-out in a board, usually to accommodate some aspect of the mechanical design of the product.

This can be achieved by drawing 2D graphics on the appropriate MECH layer - see below.

Unfortunately, there is no widely accepted mechanism within the industry for specifying the location of slots and cut-outs within the CADCAM data. Therefore, the best we can do is to output the coordinates of each routing stroke in Gerber format, and then leave it to individual board manufacturers to convert the data for their own particular routing machines.

To do this, we make use of one of the MECH layers within ARES. The CADCAM Output dialogue form allows you to specify which MECH layer is to be used, and any slotted pads which then generate appropriate routing strokes on that layer. 2D graphics placed on that layer will then also be interpreted as defining routing strokes.



Setting Mech1 to be the Slotting / Routing layer

⚠ It is important to ensure that your board manufacturer understands that the specified mech layer contains routing co-ordinates in Gerber format. You will need to tell them this explicitly.

We will continue to monitor the industry to see if a more standardized method for specifying mechanical routing operations will emerge.

GERBER VIEWING

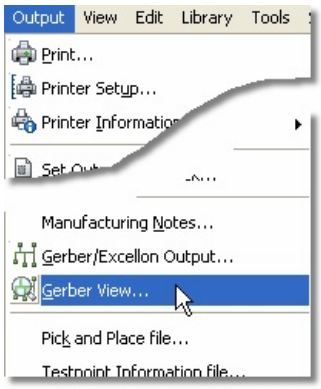
Overview

Since photoplotting charges can be quite substantial, it is useful to be able to view Gerber files to ensure that all is well prior to sending them to the bureau.

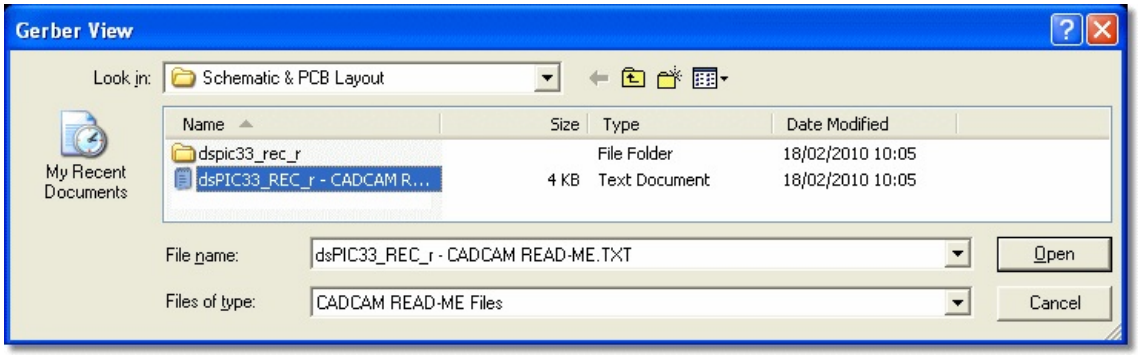
To facilitate this, ARES provides a Gerber View command which will load and display selected files produced by the CADCAM command.

To view Gerber output files

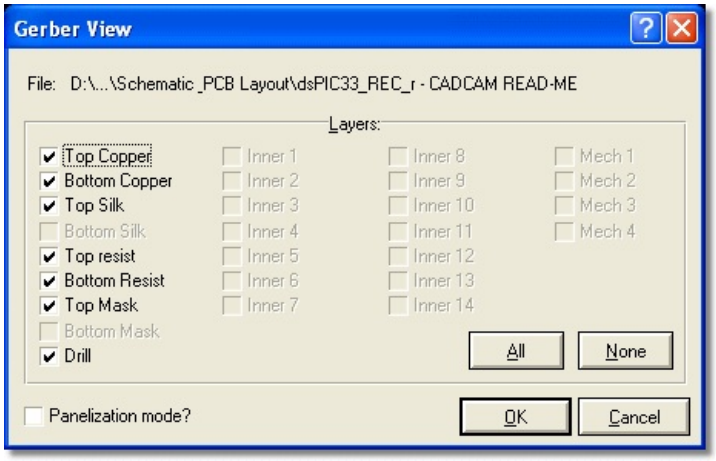
1. Invoke the Gerber View command from the Output menu. If you have modified but not saved the current design you will be prompted to do so, as the current design data is lost when the Gerber files are read.



2. Choose a Readme.txt file from the file selector. The Gerber View can only read Gerber files produced by ARES for which a Readme.txt file exists.



3. ARES will then parse the file and display a dialogue form for selecting which layers to view. By default, all available layers are loaded.



Although it is possible to perform editing operations on the Gerber View data, we do not recommend or support this mode of operation, aside for panelization which is discussed in the next section.

⚠ Although integrated with ARES The Gerber Viewer is **not** the place to make edits to your layout. If you spot something that needs altering or removing then you should do so inside the ARES Editor and then regenerate the Gerber/Excellon files.

PICK AND PLACE FILE

Overview

ARES incorporates the ability to produce a file for use as a starting point in setting up automatic insertion machines. Essentially this file lists the component layers, positions and rotations in standard quote/comma delimited format.

An example file is shown below:

```
LABCENTER PROTEUS PICK AND PLACE FILE
=====================================
Component positions for K:\Prodev\Ares\ppsu.LYT
Fields: Part ID, Value, Package, Layer, Rotation, X, Y
Units:  Rotation - degrees, X, Y  - thou
Notes:  The X, Y value is the centre of package as drawn in ARES.
        The origin for these values is the Output Origin.
        The values are a guide only and must be checked manually when
        setting up automatic insertion equipment.
"U1","","DIL08",TOP,0,6000,5000
"Q1","","TO220",TOP,180,6050,5375
"D1","","DIODE30",TOP,180,6050,5250
"R1","","RES40",TOP,270,6300,5050
"R2","","RES40",TOP,270,6400,5050
"Q2","","TO92",TOP,90,5650,5050
"R3","","RES40",TOP,180,5800,4750
"C1","","CAP10",TOP,180,6200,4750
"C2","","CAP10",TOP,0,5650,5350
```

There are a number of points that must be appreciated about this file:

- The origin for the co-ordinates is the Output Origin - this is the same origin as is used for the Gerber and Excellon outputs.
- The (x,y) coordinates are quantified in units of 1 thou and represent the centre of the component's package. This position may or may not correspond to the origin of the component for auto-insertion, but it will be in approximately in the correct place. Our understanding is that this is helpful in that it provides a starting point for manual alignment of the placement head.
- The rotations are in anti-clockwise values in degrees relative to the orientation of the package when it was defined. Since there is no standard for default orientations of packages these values may be of limited use unless they can be combined with a translation table that is specific to ARES packages including ones you have defined yourself. This is a matter between yourself and whoever is providing the auto-insertion facility.

As far as we are concerned, this is an evolving and experimental area of the software; a number of users have requested an output file of this nature but there is as yet no industry standard for us to base it on. Any feedback on this issue will be greatly appreciated.

TESTPOINT INFORMATION FILE

Overview

A common method of testing both bare and assembled boards is to use a 'bed of nails' test unit. This device makes connections to a large number of the exposed pads on the underside of the board, and then - in conjunction with a netlist and other information - checks the connectivity between the test-points. If there are any missing components, short circuits or dry joints, these will likely be detected and the board can be pulled out of the production line for repair or rejection.

In order to set up the testing machine, information is required as to the location of component pads on the underside of the board and the electrical nets that they are one. ARES provides a special output file containing this information.

An example file is shown below:

```
LABCENTER PROTEUS TESTPOINT INFORMATION FILE
============================------------------------------
Testpoint positions for k:\prodev\ares\test.LYT
Fields: ID, Type, X, Y, Net
Units:  X, Y  - thou
Notes:  This file lists pads which are accessible from the bottom
        The X, Y value is the centre of drill hole or pad origin.
"Q1:B","THRU",6150,5400,"#00000"
"Q1:C","THRU",6050,5400,"UNREG"
"Q1:E","THRU",5950,5400,"VOUT"
"D1:A","THRU",6200,5250,"VOUT"
"D1:K","THRU",5900,5250,"#00004"
"R1:1","SURF",6350,5250,"VOUT"
"R1:2","SURF",6350,4850,"#00001"
"R2:1","SURF",6450,5250,"#00001"
"R2:2","SURF",6450,4850,"GND=POWER"
"Q2:E","THRU",5650,4950,"GND=POWER"
"Q2:C","THRU",5650,5050,"#00004"
"Q2:B","THRU",5650,5150,"#00005"
"R3:1","THRU",6000,4750,"DIGITAL"
"R3:2","THRU",5600,4750,"#00005"
"C1:1","SURF",6250,4750,"ANALOG"
"C1:2","SURF",6150,4750,"GND=POWER"
"C2:1","SURF",5600,5350,"#00003"
"C2:2","SURF",5700,5350,"#00004"
"VIA","THRU",4650,4550,"#00003"
"VIA","THRU",4550,5700,"#00004"
```

There are a number of points that must be appreciated about this file:

- Pads are identified as either <PARTREF>:<PINNO> or "VIA".
- Pad types are either "THRU" where there is a drill hole, or "SURF" if not.
- Only pads and which are accessible from the underside of the board, and which have

exposed copper are listed. Pads which are entirely covered by the solder resist are not listed.
- The (x,y) coordinates are quantified in units of 1 thou and represent the connection point of the pads. This is the point that you track to in ARES in order to achieve electrical connectivity.
- The origin for the co-ordinates is the Output Origin - this is the same origin as is used for the Gerber and Excellon outputs.

As far as we are concerned, this is an evolving and experimental area of the software; a number of users have requested an output report of this nature but it isn't entirely clear to us exactly what information is needed. Any feedback on this issue will be greatly appreciated.

ODB++ Output

Overview

The ODB++ format was developed by Valor Systems and combines the information contained in the Gerber, Excellon, Pick & Place and Testpoint Information outputs into one fileset. The ODB++ database also contains a high level representation of the board in terms of the components, packages and netlist.

There are various advantages in using ODB++ rather than the individual output formats:

- ODB++ is a single, standard format, used throughout the PCB manufacturing industry.
- All the information required to manufacture the bare board including the layer stackup, tool sizes and hole types is included in the database; there is no need for a human readable Tool Information File.
- The package types, component placement and netlist information are also included so that pick and place and bare board test equipment can be driven from the ODB++ data.
- Power planes are represented as polygons rather than being rastered; this results in smaller file sizes especially for boards of large physical size.
- A free ODB++ viewer is available from Valor enabling you to validate the output using an independent software package before sending your boards for manufacture.

ODB++ Output is part of the PCB Design Advanced Features Set and therefore requires a licence for PCB Design Level 2, PCB Design Level 2+ or PCB Design Level 3.

Using the ODB++ Output Command

The ODB++ command is similar to the CADCAM output command although there are slightly fewer options.

- A common filestem and location for all files produced. This defaults to the layout filestem. Note that the ODB++ format is contained in a directory hierarchy, so the filestem determines the root folder of the ODB++ database.
- Whether you want all the outputted files to be collated in a single zip archive or left as a collection of individual files in the folder of your choice. Depending on your choice you can then opt to automatically launch windows explorer or the zip archive on completion, allowing you to browse the CADCAM files. ZIP file generation is not implemented for ODB++ output at the time of writing (ARES 7.3 SP0). It will likely be released as part of a service pack. You can, of course, create a ZIP file of the database folder from within Windows Explorer.
- Which layers you wish to output for the current design. This will be defaulted automatically based on an analysis of what is contained in the layout.
- Which layer is used for mechanical routing data.
- What resolution is used for rastering truetype fonts and bitmaps. Note that unlike CADCAM output, power planes are represented as polygons and are not subject to rastering.
- Comment fields for manufacturing notes and contact details. This comment, if specified, allows you to add design specific manufacturing notes to the README.TXT generated within the ODB++ database folder.
- A Zip File comment. Zip archives allow a comment to be appended to them and this comment appears by default when you or your board manufacturer) opens the zip archive. This is a useful way to ‘sign’ your archives or to provide special instructions etc. for your board manufacturer.
- Whether or not to launch the Valor Viewer automatically on production of the CADCAM

files. This option will load the ODB++ database into the Valor Universal Viewer on completion, allowing you to view the resulting output prior to sending the files to your PCB manufacturer for prototyping.

You will need to download and install the free Valor Universal Viewer before this option will work.

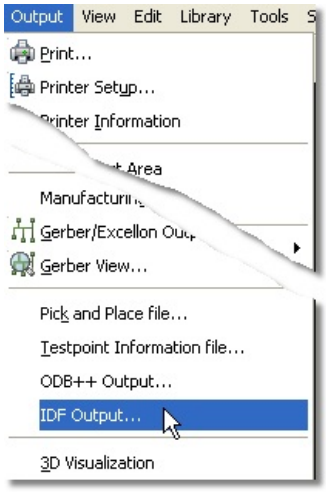
IDF OUTPUT

Overview

The Intermediate Data Format (IDF) is a specification designed to exchange printed circuit assembly (PCA) data between mechanical design and PCB layout. Initially developed in 1992, the IDF format has since become a de facto industry standard.

The format includes all information that is commonly shared among mechanical design, circuit board layout and physical analysis and is directly supported by most MCAD products including Solidworks.

Export in this format from the IDF output command on the Output Menu.



Exporting to in IDF format

To generate an IDF fileset for the layout

1. Select the IDF Output command from the Output Menu in ARES
2. Specify the job name, directory and optional archive options.
3. Select the layers to be included in the output - this will default to the layers used in the current project.
4. Identify the mechanical layer used for slots if present on the layout.
5. If you want a common solder resist expansion (guard gap) across the entire layout then check the box and set the clearance.
6. Enter any notes in the log field; useful both to log design specific information and also to identify the slotting layer.
7. Generate the IDF fileset.

Note that IDF output is fundamentally a 2D format that contains some 3D information. This information is taken from the 3D Visualisation parameters such as height specified for the footprint (right click on package, select make package and switch to the 3D tab). If no 3D information is supplied a default is passed.

The board and feature thickness is taken from the Board Properties command on the Technology menu.

PANELIZATION

Overview

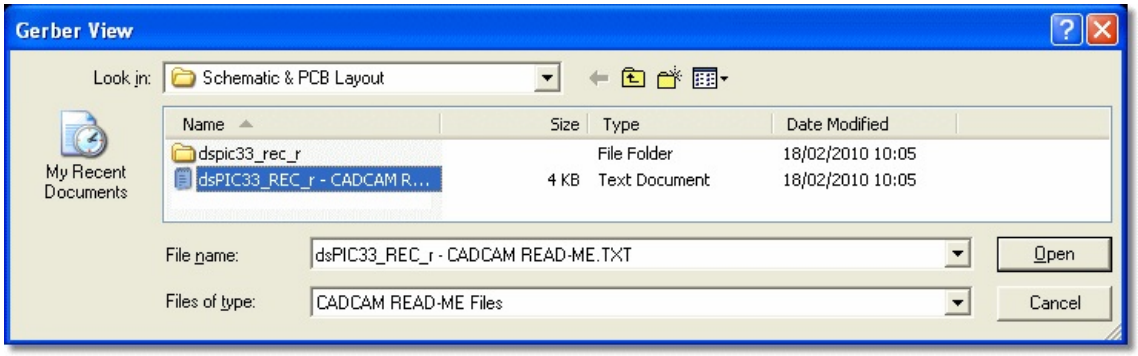
Panelization refers to the processes of combining several board artworks onto one panel in order to reduce manufacturing costs. It is common practice to panelize both multiple copies of the same board and/or several different boards.

ARES provides facilities to achieve this through the Gerber Viewer.

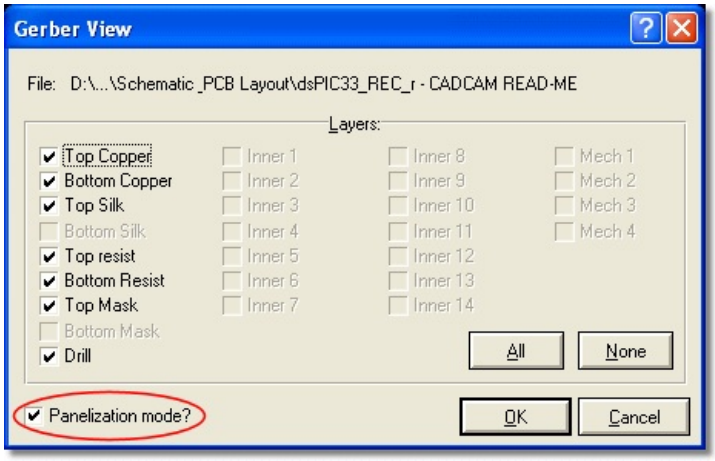
To create a panelized artwork

1. Use the CADCAM Output command to produce Gerber and Excellon files for each board to be included on the panel.

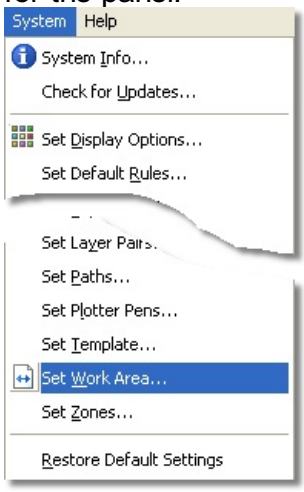
2. Invoke the Gerber View command and choose the Readme.txt file of the first board to go on the panel.



3. Accept the default settings for the layers (these should correspond with the layers you chose to generate at step 1) and check the Panelization Mode checkbox.



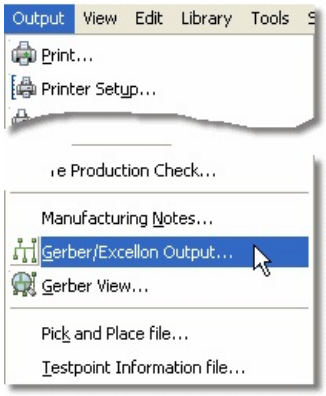
4. Click OK. ARES will import the CADCAM data for the board and display it as a tagged set of objects.
5. Use the Board Properties command on the Technology menu to specify the dimensions for the panel.

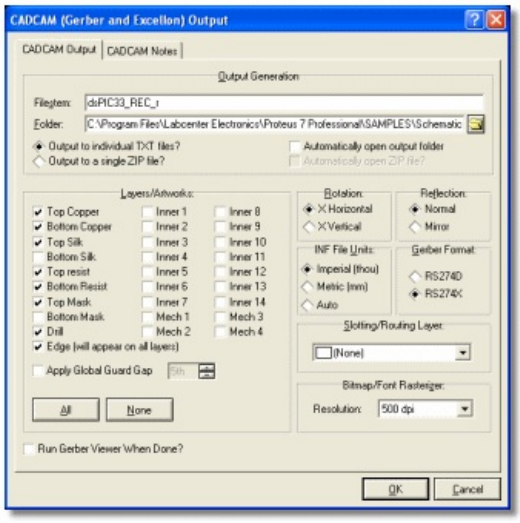


6. Use the Block Move command to re-locate the board image to where you want it on the panel.

7. If the panel is to contain multiple images of a single board only, you can now use the Block Copy command to lay them out.

8. If the panel is to contain images of additional boards, you can import them into the panel by returning to step 2 above.
9. When the panel is complete, use the CADCAM Output command to generate final artwork files for the panel. It is these files that you need to send to your board manufacturer.





There are a number of points to note about the panelization process:

- You can add additional text and graphics to the panel as required to designate board IDs and other manufacturing information.
- When importing multiple board images, ARES will assign new pad and track style codes (D-Codes) for each image. However, when the CADCAM files are re-generated, ARES will re-combine styles that are actually the same, so that the total usage of aperture and tool codes will not be excessive.
- The resulting set of files may be quite large - especially if the board contains ground planes or non-orthogonal components. However, that is an inevitable consequence of producing a file that contains multiple board images. Fortunately, the files are ASCII, and will compress quite well with WINZIP or similar, should you need to transmit them by email.
- The process is not suitable for panelizing boards containing buried or blind vias, because the layer ranges of the drill holes are not preserved. If you need to do this, we recommended you make use of a 3rd party Gerber viewer/editor package.

DXF IMPORT

Overview

The DXFCVT converter is fully integrated into ARES and is driven through the Import DXF tool on the File menu. It facilitates the import of mechanical data in DXF format, and we envisage it being used in situations where a board needs to be fitted into an existing mechanical design. It could also be used to import other graphics such as company logos.

The conversion process is as follows:

- Generate the DXF file from your mechanical CAD application. This should, if possible, be a 2D drawing. ARES cannot handle 3D data and the converter thus ignores all z co-ordinates in the DXF file so 'flattening' 3D drawings to two dimensions.
- Determine the mechanical layers in the DXF file that are to be combined and placed on particular ARES layers. The DXF Import dialogue form allows you to specify one or more layer assignments with each assignment specifying which of the mechanical layers in the DXF file are to be combined and placed on the specified ARES layer.

The DXF importer will automatically determine the correct units, provided that they are specified within the input file.

You should appreciate that Autocad (the application around which the DXF format was based) supports much more powerful mechanical drafting than that offered by ARES. Thus some entities in a DXF file are ignored by DXFCVT, and others are approximated. However, for the purpose of displaying simple cabinet internals and the like inside ARES, DXFCVT is quite adequate to the task.

See the Limitations section for specific information on the limitations of the DXFCVT converter.

PERFORMING A DXF IMPORT

Generating the DXF File

The DXF file to be imported should be generated by the mechanical CAD application you use. Consult the documentation or on-line help associated with the CAD application to determine how best to do this.

As far our import converter is concerned, the only requirements are that:

- The file exported by the application be in plain ASCII and not in the alternative binary (DXB) format.
- AutoCAD 2000 is the preferred output version, although other versions should work as well (versions prior to R14 may not contain the dimension units).

Layer Assignments

Having generated the DXF file you must now decide on what layer assignments you require. Each layer assignment tells the converter which layers in the DXF file are to be combined and placed on a single ARES layer. The controlling dialogue form is laid out in a fashion that should make this process fairly transparent and Dialogue Level Help is provided for all fields on the form should you encounter any problems.

Note that, for a given conversion, an ARES layer may only be assigned once. An attempt to assign a layer twice will result in an error.

Basic Conversions

To perform a DXF file conversion you need to load the converter with the name of the DXF file to be converted. This process is implicit with ARES in that immediately on selecting the Import DXF tool you will be asked to open the appropriate DXF file. A successful loading of the file will result in the appearance of the DXF Import dialogue form. All available options on the dialogue form are covered via Dialogue Level Help on the appropriate field.

Conversion Errors

With regard to DXF file parsing errors, the converter has been written to be as fault-tolerant as is consistent with reading a DXF file. Parsing errors are only generated when:

1. The DXF file doesn't contain an ENTITIES section. A DXF file without entities is deemed 'empty', which makes conversion impossible.
2. The DXF file contains a section (e.g. HEADER, ENTITIES) or an primitive (e.g. LINE, CIRCLE, etc.) that doesn't contain a single record. All sections should at least contain an end-of-section record and all primitives should contain at least one record. This implies the DXF file is corrupted.
3. The end of a BLOCK entity in a non-BLOCKS section or the end of the BLOCKS section before the end of a BLOCK entity. This implies the DXF file is corrupted.
4. The failure to extract (parse) data of a type consistent with the group record code. For example a record with a group code of 10 implies an integer value and a record with a group code of 41 implies a floating point value. An error is generated if (regardless of formatting) a value cannot be parsed.

Whenever an error occurs the converter reports the error in the form:

```
ERROR   (000413): Entity Group Expected - EOF found?
```

and the conversion is aborted.

For file parsing errors, the number of the last line read from the DXF file is shown in brackets after the ERROR keyword (note that the converter reads whole groups - two lines - from the DXF file at a time, and so any error may be on the line indicated or the preceding line). For non-parsing errors, the line number is displayed as dashes.

Conversion Warnings

Apart from errors (described above) all other reports of unusual or possibly erroneous conversion behaviour are treated as warnings. Warnings are not fatal (they do not stop the operation and completion of the converter) but are useful if the resulting region file is not what was expected. In order to be complete, the converter issues warnings whenever it encounters a situation that may or may not lead to unusual region files and this includes any record it reads and discards as being either of unknown purpose or unconvertible. Warnings are only generated if requested (through the Generate Warnings checkbox on the dialogue form).

Warnings are displayed in a similar format to errors (described above) except that the DXF file line number line and message text is preceded by the word WARNING. Generated warnings are available for viewing in the DXFCVT.LOG and DXFINFO.LOG files placed in the current working directory. The DXFINFO.LOG file is generated even if import is aborted before the dialogue form is shown.

DXF Limitations

DXF Limitations

The principle limitations of the DXFCVT converter are summarised as follows:

- DXF files must be in plain ASCII. The converter does not support the DXB binary file format.
- Three-dimensional DXF drawings are flattened to two-dimensions ('plan' view) by the converter as ARES only supports two dimensions.
- The entities (alone or as part of a BLOCK entity) supported are the BLOCK, LINE, SPLINE, POLYLINE, LWPOLYLINE, ELLIPSE, CIRCLE, ARC, TEXT, SOLID, TRACE, and INSERT entities. All other entities are ignored and do not affect file conversion. The SOLID and TRACE entities are converted as polygons. In particular, the DIMENSION entity is not supported.
- Only those attributes of an entity that can be honoured by ARES are converted. Thus text attributes such as font face, weight (boldness) and italic slope angle are discarded and do not affect the file conversion.

Other minor limitations include:

- Quadratic Bezier curves (1 per quadrant) are used to represent ARCS and ELLIPSES, resulting in a small inaccuracy.
- Line types (dots, dashes, thicknesses, etc.) are lost. All lines are converted to the standard graphics line width.
- Colours are lost - all converted entities are displayed in a colour appropriate to the ARES layer they are on.
- Text styles (from the DXF TABLES section) are ignored. The converter relies (for those attributes it converts, such as rotation, mirroring, etc.) on these attributes being specified as part of the TEXT entity - this is the norm.
- The conversion of DXF TEXT entities drawn in a proportional text style will convert to a fixed-pitch font in ARES where the width of the characters may result in the overall length of the string in ARES being different to the overall length of the string in the DXF drawing. This may then lead to alignment problems unless the user has made judicious use of the DXF text alignment flags (e.g. text that is specified as fitting between two points).As the ARES text primitive is limited to fixed-pitch characters which only allows for the specification of a character height and character width for the text string, the only way around this problem would be to adjust the ARES character width such that the number of characters in the string multiplied by the adjusted character width is the same as the overall length of the string in the DXF file. Unfortunately, except for certain justifications, the DXF file does not contain the overall string length and so this isn't possible. The actual behaviour of the converter is to calculate the character height as being equal to the height of the DXF TEXT entity and the character width as being equal to the TEXT entity height scaled by the TEXT entity's x-scaling factor (the latter defaults to 1.0).
- The following text attributes and properties are not converted:

1. Obliquing angle (for italics).
2. Styles - all text is in ARES's fixed-pitch vector font.
3. Control characters. These occur in the DXF file as a caret followed by an ASCII character - all such characters in the string are preserved.

- The following text attributes and properties are converted:

1. Rotation.

2. X and Y mirroring.
3. Justification, except attribute '5' ("Text is fit [sic] between two points (width varies)") as it is ambiguous and hasn't been tested.

- Only the following entities are converted as noted.:

1. LINE and 3DLINE entities.
2. CIRCLE entities.
3. ARC entities (by default, by using bezier quadrant arcs).
4. SOLID, TRACE and 3DFACE entities are converted to unfilled three/four sided polygons. The 'hidden' attributes of 3DFACEs is not honoured.
5. TEXT entities are converted as described above.
6. BLOCK entities.
7. INSERT entities are converted but any associated ATTRIB entities are discarded - this leads to text declared in the INSERT via a ATTDEF entity being lost.
8. LWPOLYLINE and POLYLINE with associated VERTEX entities. All polylines are assumed to be continuous (i.e. not a mesh) and all VERTEX entities are assumed to be single point or a bulge.
9. ELLIPSE using bezier quadrant arcs.
10. SPLINE entities are only supported if they are non-rational (weighting is 1) and or degree 3 or lower. Fit points are not supported.

- In addition to the above some other omissions are:

1. DIMENSION entities - these are usually also exported as LINE, SOLID and TEXT entities as well, so there should be no loss of visible information.
2. ATTRIB and ATTDEF entities, as outlined above.
3. SHAPE entities - these will (if a problem) be added in a future release.

- If the DXF file does not specify the units it is the user's responsibility to specify the scaling parameters (via the appropriate options on the dialogue form) scale the dimensions to 1th. Positive angles are assumed to be measured in degrees counter-clockwise from an implied zero axis corresponding to the positive x-axis. The necessary DXF file header variables are checked and a warning issued if this is not the case.

Layer names are honoured throughout out the conversion, with the layer name "0" (the single character zero) being taken to have the layer name of it's parent. For example, for a BLOCK object with entities on layers 0, BLUE and GREEN and an INSERT of the BLOCK on layer RED, the BLOCK entities on layer 0 are deemed to be on layer RED.How the converter determines which entities with a BLOCK are to be output is determined by which layers are enabled and the settings chosen under the Symbols group on the dialogue form.

ICON REFERENCE CHART

Command Icon Chart


New Layout


Load Layout


Save Layout


Import Region File


Export Region File


Print Layout


Print Area


CADCAM Output


Gerber Viewer


Refresh


Flip Layout


Grid On/Off


Display Layers

m
Metric / Imperial Toggle


False Origin


Z-Theta


Pan


Zoom In


Zoom Out


Zoom to Layout


Zoom to Area


Undo


Redo


Block Copy


Block Move


Block Rotate


Block Delete


Pick Device from Library


Make Package


Decompose Package


Trace Angle Lock Toggle


Auto Trace Style Toggle


Auto Track Necking Toggle


Search & Tag


Auto Component Annotator


Autorouter


Connectivity Rules Checker


Design Rule Checker

Mode Selection Icons


Selection Tool


Component Mode


Package Mode

Trace Mode

Via Mode

Zone Mode

Ratsnets Mode	*Connectivity Highlight*	*TH Circular Pad Mode*
TH Square Pad Mode	*DIL Pad Mode*	*SMT Edge Connector Mode*
SMT Circular Pad Mode	*SMT Polygonal Pad Mode*	*Pad Stack Mode*
2D Line Mode	*2D Square Mode*	*2D Circle Mode*
2D Arch Mode	*2D Path Icon*	*2D Text Mode*
2D Symbols	*2D Marker Mode*	*Dimension Tool*
Rotate Clockwise	*Rotate Anti-clockwise*	*Mirror Object*
Flip Object		