

ELECTRA v8.05

Routing engine

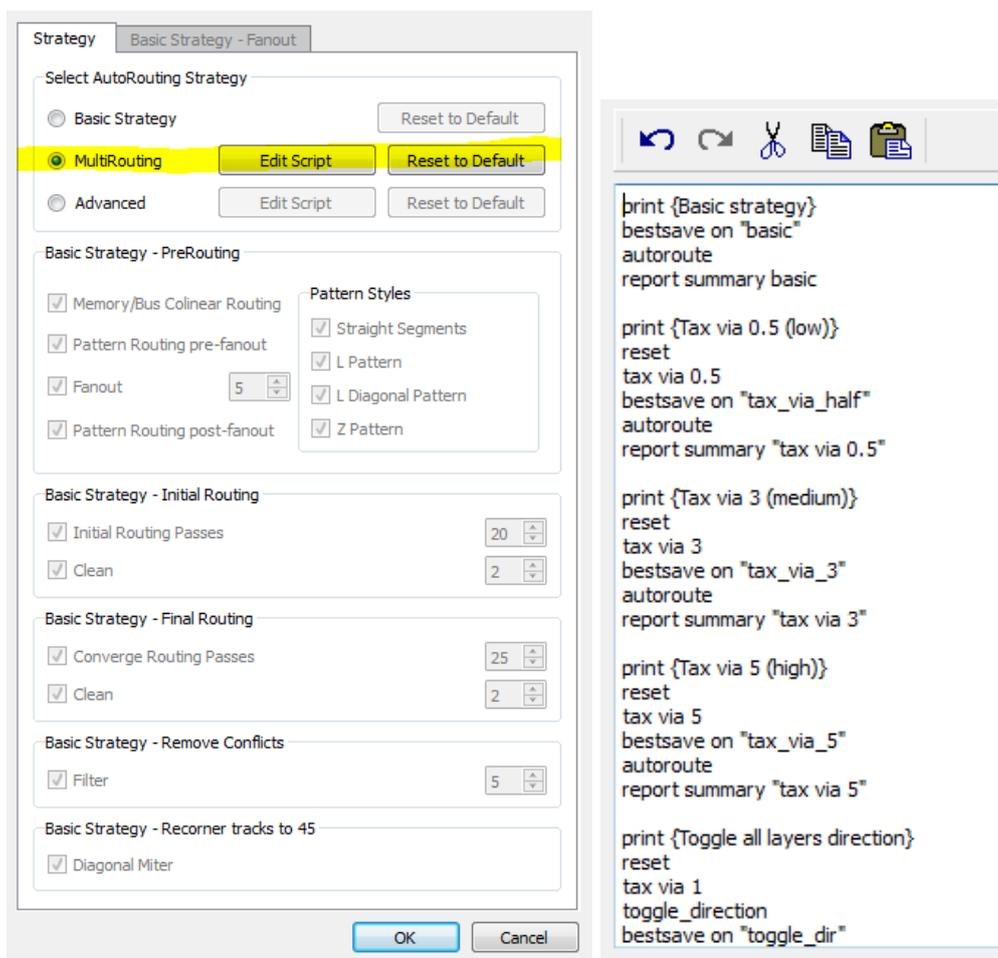
- A new push and shove routing pass is now applied automatically on highly congested boards to resolve spacing violations more quickly than by rerouting nets. This improves performance and completion rate.

Multi-Routing strategy

Provides an easy method to define and run multiple routing strategies within the same session.

The default multi-routing script runs the basic strategy, then re-runs multiple times the same strategy with low / medium / high via cost, and finally re-runs again after toggling the layer orientation.

A built-in text editor can be used for further customization.



The image shows the 'Strategy' dialog box in the software, with the 'Basic Strategy - Fanout' tab selected. The 'MultiRouting' option is highlighted in yellow. Below the dialog box, a text editor displays a script for the MultiRouting strategy.

Strategy Dialog - Basic Strategy - Fanout

- Select AutoRouting Strategy:
 - Basic Strategy (Reset to Default)
 - MultiRouting** (Edit Script, Reset to Default)
 - Advanced (Edit Script, Reset to Default)
- Basic Strategy - PreRouting:
 - Memory/Bus Colinear Routing
 - Pattern Routing pre-fanout
 - Fanout (5)
 - Pattern Routing post-fanout
 - Pattern Styles:
 - Straight Segments
 - L Pattern
 - L Diagonal Pattern
 - Z Pattern
- Basic Strategy - Initial Routing:
 - Initial Routing Passes (20)
 - Clean (2)
- Basic Strategy - Final Routing:
 - Converge Routing Passes (25)
 - Clean (2)
- Basic Strategy - Remove Conflicts:
 - Filter (5)
- Basic Strategy - Recorner tracks to 45:
 - Diagonal Miter

Text Editor Content:

```
print {Basic strategy}
bestsave on "basic"
autoroute
report summary basic

print {Tax via 0.5 (low)}
reset
tax via 0.5
bestsave on "tax_via_half"
autoroute
report summary "tax via 0.5"

print {Tax via 3 (medium)}
reset
tax via 3
bestsave on "tax_via_3"
autoroute
report summary "tax via 3"

print {Tax via 5 (high)}
reset
tax via 5
bestsave on "tax_via_5"
autoroute
report summary "tax via 5"

print {Toggle all layers direction}
reset
tax via 1
toggle_direction
bestsave on "toggle_dir"
```

New commands are introduced to support multi-routing strategy.

The report command has been extended to support the generation of short status report between the various runs and a final summary table report for the comparison of the routing results.

report summary <name> -Make summary routing status and assign name

```
report summary toggle_direction
```

Summary	Completion	Shorts	DRC Errs	Failures	Unroutes	Vias	Length	Total time
toggle_direct	100.00%	0	0	0	0	876	1072357.4 mil	0:00:49

report summary -all -Report all summaries from previous named run

```
report summary -all
```

Summary	Completion	Shorts	DRC Errs	Failures	Unroutes	Vias	Length	Total time
basic	100.00%	0	0	0	0	861	1077354.5 mil	0:00:47
tax via 0.5	100.00%	0	0	0	0	893	1057655.2 mil	0:00:56
tax via 3	100.00%	0	0	0	0	812	1091568.8 mil	0:00:46
tax via 5	100.00%	0	0	0	0	802	1096323.9 mil	0:00:52
toggle_direct	100.00%	0	0	0	0	876	1072357.4 mil	0:00:49

report summary -clear - Clear all routing summaries

reset

Removes all of the routing result, including routing history.
Allows restarting with a new routing strategy.

Advanced Routing strategy

Electra includes a built-in TCL interpreter with DO file extensions. You can develop a routing strategy with high level programming language features such as for-loop, test conditions, file IO, etc... The default advanced routing strategy illustrates such strategy as a code program.

The image shows a software interface for configuring a routing strategy. On the left is a dialog box titled "Strategy - Basic Strategy - Fanout". It has several sections:

- Select AutoRouting Strategy:** Three radio buttons: "Basic Strategy", "MultiRouting", and "Advanced" (which is selected and highlighted in yellow). Each has a "Reset to Default" button. "Advanced" also has an "Edit Script" button.
- Basic Strategy - PreRouting:** Checkboxes for "Memory/Bus Colinear Routing", "Pattern Routing pre-fanout", "Fanout" (with a value of 5), and "Pattern Routing post-fanout". A "Pattern Styles" sub-section has checkboxes for "Straight Segments", "L Pattern", "L Diagonal Pattern", and "Z Pattern".
- Basic Strategy - Initial Routing:** Checkboxes for "Initial Routing Passes" (value 20) and "Clean" (value 2).
- Basic Strategy - Final Routing:** Checkboxes for "Converge Routing Passes" (value 25) and "Clean" (value 2).
- Basic Strategy - Remove Conflicts:** Checkboxes for "Filter" (value 5).
- Basic Strategy - Recorner tracks to 45:** Checkboxes for "Diagonal Miter".

At the bottom of the dialog are "OK" and "Cancel" buttons. To the right of the dialog is a text editor window showing the TCL script for the "Advanced" strategy:

```
|
print {PreRoute phase =====}
bestsave on
bus diagonal
proute
fanout 5
proute
print {Initial Route phase =====}
route 10
if {[completion] < 100} {
  clean 2
}
print {Route phase 1 =====}
set count 5
while {$count > 0 && [completion] < 100} {
  set starting_completion [completion]
  if {$count == 5} {
    route 10 11
    clean 2
  } else {
    route 10 16
    clean 2
  }
}
if {[completion] < 100 && [completion] > $starting_completion} {
  incr count -1
} else {
  set count 0
}
}
print {Retrieve best result and complete=====}
```

Advanced Routing and access to routing data

get_unrouted_nets

Outputs the list of net names that are completely or partially unrouted.

get_unrouted_wires <netname>

Outputs the list of unrouted wires within a given net.

select wire <fromComp-fromPin> <toComp-toPin>

Selects the wire connection between the From and To pin. The connection must be defined as part of the ratsnest. To redefine a connection, you can use the *select fromto* <fromCmp-pin> <toComp-pin> command as overwriting command.

get_val <arg>

supported arguments:

<i>completion</i>	- routing completion rate
<i>connections</i>	- total number of connections
<i>instances</i>	- number of component instances
<i>nets</i>	- number of nets
<i>pairs</i>	- number of pairs defined
<i>routed_length</i>	- total routing length
<i>selected_nets</i>	- number of selected nets
<i>selected_wires</i>	- number of selected wires
<i>shorts</i>	- number of conflicts of type short
<i>spacings</i>	- number of DRC clearance violations
<i>total_vias</i>	- number of vias in total
<i>unroutes</i>	- number of unroutes

setr undocumented command

setr is mainly used to store GUI settings that are related to the routing strategy dialog box. The GUI settings values are persistent on disk, the user does not need to intervene to reload them.

setr is also used to control some routing style.

Here are the controls that are available:

droute_style <45/90> (default = 45)
Differential pair must be autorouted with 45 degrees (default) or 90.

tune_corner_style <0/1> (default = 1, chamfered)
Serpentine pattern must have 90 degrees or chamfered corners.

pad_wire_necking <0/1> (default = 0)
When set to 1 his will neck the last wire segment entry/exit to the pad if the pad size is smaller than the track width.