

net= attribute mini-HOWTO

Ales Hvezda

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

October 2nd, 2003

Contents

1	What is the net= attribute used for?	3
2	What is the format of the net= attribute?	3
3	How do you actually use the net= attribute	4
4	Caveats / Bugs	4
5	Example	5
6	Final notes	7

The information in this document is current as of 19991011.

1 What is the net= attribute used for?

The **net=** attribute is used to specify power, ground, and/or arbitrary nets in the gEDA system.

The **net=** attribute is used instead some of the other systems of specifying power/ground (such as having power/ground pins on symbols or power boxes). Some devices have lots of power/ground pins and having all of these pins on the symbol would increase its size and make it unmanageable. The **net=** attribute is the power/ground specification of choice in the gEDA system because of its simplicity and versatility.

Now having said all this, you can have power/ground pins on a symbol, but gnetlist will probably not recognize these nets connected to these pins as separate power/ground nets. Please keep this in mind as you draw symbols.

2 What is the format of the net= attribute?

Attributes in gEDA are simple text items which are in the form **name=value**. All proper attributes follow this form. Attribute names are always lower case, but the value can be upper or lower case. gnetlist and friends are case sensitive. Typically net/signal names by default are upper case.

Attribute can be attached to an object or in certain cases (like the **net=** attribute) can be free floating (not attached to anything). The free floating attributes are also called toplevel attributes.

The **net=** attribute is a text item which takes on the following form:

```
net=signalname:pinname,pinname,pinname,... where  
net=          The attribute name (always the same, lowercase)  
signalname   The signal or net being defines (like +5V, GND, etc...)  
pinname      The pin name (or number) which is assigned to this signal/net (or  
               pin names/numbers)
```

The **signalname** cannot contain the **:** character (since it's a delimiter). The **pinname** is the pin name (A1, P2, D1, etc...) or pin number (1, 2, 5, 13, etc...). The **pinname** cannot contain the **,** character (since it's also a delimiter). **pinnames** are typically the same sort of numbers/names like the **pin#=#** attribute (if your familiar with that attribute).

You can only have ONE **signalname** per **net=** attribute, but you can have as many **pinnames/numbers** as you want.

3 How do you actually use the net= attribute

You can place the `net=` attribute in several places. Here's the list so far:

- Inside a symbol either as an attached attribute or an unattached attribute (toplevel attribute). Example which creates power/gnd nets: `net=GND:7` or `net=+5V:14`
- Outside a symbol (which is instantiated on a schematic) attached as an attribute to override an existing `net=` created net/signal. Suppose a symbol has a `net=GND:7` inside it already; attaching this to the symbol: `net=AGND:7` overrides the GND net (on pin 7) calling it AGND and connects/associates it to pin 7.
- Outside or inside a symbol to connect a net to a visible pin automatically. This is still untested and still might have some undesirable (negative) side effects. Use with caution.
- Attached to one of those special power/gnd symbol (like `vcc/gnd/vdd`) and you can change what that symbol represents. You could change the ground symbol to create a net called `DIGITAL_GND` without editing the symbol (`net=DIGITAL_GND:1`).

In the current symbol (19991011) library there are symbols named `vdd-1.sym`, `vcc-1.sym`, `vee-1.sym`, etc... which do not have a `net=` attribute inside, so you must attach the `net=` attribute yourself (in the schematic).

There also symbols named `5V-minus-1.sym`, `12V-plus-1.sym`, `9V-plus-1.sym` etc... which have the appropriate `net=` attribute in them already (can be overridden though). You can use these symbol as examples of how to use the `net=` attribute.

You can have as many `net=` attributes as you want. Just remember that `net=` attributes attached to the outside of a symbol override any equivalent internal (inside the symbol) `net=` attributes. If you run into a case where this doesn't work, please let ahvezdaATgeda.seul.org. In fact, send any bug reports to that individual.

4 Caveats / Bugs

The `net=` attribute/mechanism is fairly new, so there are bound to be bugs (many bugs). Here are some of the identified issues:

- As of 19991011 almost all of the symbols in the standard library do not have the `net=` attribute or any other power/ground specifiers. Hopefully this will be updated sometime (any volunteers?).

- Attach a special power symbol (vcc/gnd) to a already named net will alias (rename) that net to the signalname specified in the **net=** attribute (in/attached to the vcc/gnd symbol). You can override this (so the reverse is true) by playing with the "net-naming-priority". Be careful with this. There might be other "aliasing" issues which have not been identified yet.
- Creating a **net=** attribute which associates a signal name with a pin which is already visible on the symbol, is probably a bad idea. This does work, but all the ramifications have not been explored yet.
- It is probably a bad idea to have the same **net=** attribute attached several times. Ales has not formalized what happens in this case. Just remember that the **net=** attribute on the outside of a symbol should override the internal one.

5 Example

Here's a schematic which uses standard symbols (note: the 7400 does not have the **net=** attribute inside yet). This schematic consists of a 7400 with the **net=** attributes attached for power and ground, One of the input pins grounded using a gnd symbol and the other at a logic one using the vcc symbol (with an attached **net=** attribute). One of the input net is named, but as you will see, the netname is replaced by the **net=** signal name (see above for more info on this). The output is pulled up with a pull up resistor which has power specified using the +5V symbol.

[Maybe insert picture of schematic as well]

```
v 19991011
C 38700 58100 1 0 0 7400-1.sym
{
T 39000 59000 5 10 1 1 0
uref=U100
T 38900 59500 5 10 1 1 0
net=GND:7
T 38900 59300 5 10 1 1 0
net=+5V:14
}
N 38700 58800 37400 58800 4
{
T 37600 58900 5 10 1 1 0
netname=NETLABEL
}
N 37400 58800 37400 59200 4
N 38700 58400 37400 58400 4
N 37400 58000 37400 58400 4
```

```

C 37300 57700 1 0 0 gnd-1.sym
C 37200 59200 1 0 0 vcc-1.sym
{
T 36800 59200 5 10 1 1 0
net=+5V:1
}
N 40000 58600 41600 58600 4
{
T 41200 58700 5 10 1 1 0
netname=OUTPUT
}
C 40700 58800 1 90 0 resistor-1.sym
{
T 40800 59200 5 10 1 1 0
uref=R1
}
N 40600 58800 40600 58600 4
N 40600 59900 40600 59700 4
C 40400 59900 1 0 0 5V-plus-1.sym

```

gnetlist (using the geda netlist format) run using this sample schematic outputs this:

START header

gEDA's netlist format
Created specifically for testing of gnetlist

END header

START components

R1 device=RESISTOR
U100 device=7400

END components

START renamed-nets

NETLABEL -> +5V

END renamed-nets

START nets

+5V : R1 2, U100 14, U100 1

```
GND : U100 7, U100 2  
OUTPUT : R1 1, U100 3
```

```
END nets
```

Notice how NETLABEL was renamed (aliased to the +5V net).

6 Final notes

Send all bugs to ahvezda@geda.seul.org or geda-dev@geda.seul.org (mailing list, please subscribe first)

[I'm sure there's more to say here]