gEDA/gaf Master Attribute Document

Ales V. Hvezda, ahvezda@geda.seul.org

July 6th, 2004

Contents

1 Overview

This document describes all the attributes used in in gEDA/gaf (GPL'd Electronic Design Automation / Gschem And Friends). This document is broken down into several section: this overview, symbol only attributes, schematic only attributes, attributes which can appear in both symbols and schematics, and attributes which are obsolete or deprecated.

In this document, attribute names are in **bold** and examples are in the `typewriter` font.

2 What are Attributes?

Attributes in the gEDA/gaf system are nothing more than text items which take on the form: **name**=value. Name can be anything just as long as it doesn't contain a equals sign. Value can also be anything just as long as it is something (vs nothing). **name**= (without a value part) is not a valid attribute. Also, there cannot be any spaces immediately before or after the equals sign.

Attributes can be attached to some part of the symbol. If the attribute conveys information specific to an object, then the attribute should be attached directly to the object, otherwise the attribute should be free standing or floating. Free standing attributes just exist in the symbol file as text items which take on the form **name**=value.

3 Symbol only Attributes

3.1 device

device= is the device name of the symbol and is required by gnetlist.

device= should be placed somewhere in the symbol and made invisible. This is a free standing or floating attribute. If the object is a graphic then **device**= should be set to none (**device**=none) and attach a **graphical**= (3.2) attribute. Do not confuse this attribute with just having a text label which the device name. Do not put spaces into the device name; there are some programs which dislike spaces in the device specifier. Generally the device name is in all caps.

Examples: `device=7400` `device=CONNECTOR_10` `device=NPN_TRANSISTOR`

3.2 graphical

Symbols which have no electrical or circuit significance need a **graphical**=1 attribute. Symbols like titleboxes are purely graphical symbols. Any symbol which has **graphical**=1 is ignored by gnetlist.

graphical=1 should exist somewhere in the symbol and made invisible. This is a free standing or floating attribute. Don't forget to set **device**=none (3.1).

Example: `graphical=1`

3.3 description

The **description** attribute provides a simple one line description of what the symbol is supposed to represent.

Example: `description=4 NAND gates with 2 inputs`

3.4 author

The **author** attribute identifies the name of the author of this symbol and their e-mail address. This attribute is optional, but it is nice to know who created which symbols. It also serves the purpose of known who to contact if there are questions about the intent(s) of the symbol. This attribute is free form and it can also include people's names who modified the symbol as well as multiple e-mail addresses. It is probably also a good idea to obfuscate the e-mail address so it is not harvested for spam purposes.

Example: `author=Ales Hvezda ahvezdaATgeda.seul.org`

3.5 comment

The **comment** attribute can contain anything. This attribute can convey any additional information which might not fit into any other attribute. There can be multiple instances of this attribute.

Example: `comment=This is a comment inside a symbol`

3.6 pinseq

This attribute is used to give each pin an unique number or sequence. All pins must have a **pinseq**=# attribute attached to the pin object. This attribute should be hidden. This attribute is used extensively by gschem and gnetlist.

gnetlist will output pins in the order of increasing pin sequence. The sequence numbers start at 1 and should increase without skipping any numbers. This attribute is not the pin number (i.e. device pin numbers, like GND is 7 on TTL). For pin numbers see the **pinnumber** (3.7) attribute.

Examples: `pinseq=1 pinseq=2 pinseq=3`

This attribute replaces the obsolete **pin#**=# attribute.

3.7 pinnumber

This attribute is the pin number (i.e. like GND is 7 on 74 TTL). All pins must have a **pinnumber**=# attribute attached to the pin object.

You can have numbers or letters for the value. This attribute should be visible with the value only visible. You also need a **pinseq** (3.6) attribute.

Examples: `pinnumber=1 pinnumber=13 pinnumber=A0`

This attribute replaces the obsolete **pin#**=# attribute.

3.8 pintype

Each pin must have a **pintype**=value attribute attached to it and should be make hidden. Table 1 shows valid values for this attribute.

This attribute is not used extensively in the symbol library, but it will be used for DRC and netlisting.

Examples: `pintype=clk pintype=in pintype=pas`

in	Input
out	Output
io	Input/Output
oc	Open collector
oe	Open emitter
pas	Passive
tp	Totem pole
tri	Tristate (high impedance)
clk	Clock
pwr	Power/Ground

Table 1: **pintype** values

3.9 pinlabel

This attribute labels a pin object. This attribute is primarily used by gnetlist to support hierarchical designs.

This attribute must be attached to the pin and be left visible. Please make this attribute green (instead of the default attribute yellow).

Examples: `pinlabel=A0` `pinlabel=DATA1` `pinlabel=CLK`

3.10 numslots

If a component has multiple slots in a physical package (such as a 7400 (NAND) which has 4 NANDs per package) then you need a **numslots**=# attribute. The # is the number of slots that are in a physical device. **numslots**=# should exist somewhere in the symbol and be made invisible. This is a free standing or floating attribute. If the symbol does not need slotting, then put **numslots**=0 into the symbol file.

Example: `numslots=4`

3.11 slotdef

If a component has multiple slots in a physical package then you must attach a **slotdef**=slotnumber:#,#,#... for every device inside the physical package.

The slotnumber corresponds to the slot number. The colon after the slot number is required. For example, if a device has 4 slots then there would be **slotdef**=1:..., **slotdef**=2:..., **slotdef**=3:..., and **slotdef**=4:... attributes somewhere in the symbol and be made invisible. This is a free standing or floating attribute.

The #'s have a one-to-one correspondence to the **pinseq** attributes and specify which **pinnumber**=# is used during display (gschem) or netlisting (gnetlist).

It is recommended that all symbols which have slots have a **slot**=1 (5.2) attribute attached in the same fashion as the **device**= (3.1) attribute.

See 7400-1.sym as a concrete example.

Examples: `slotdef=1:1,2,3 slotdef=2:4,5,6 slotdef=3:7,8,9`

This attribute replaces the obsolete **slot#**=# attribute.

3.12 footprint

footprint=package_name should exist somewhere in the symbol and be made invisible. This attribute is used by gnetlist and primarily for the PCB package.

Attach this attribute just like the **device**= (3.1) attribute. This is a free standing or floating attribute.

package_name is the pcb footprint or package type like DIP14 or DIP40. Although this attribute in principle is pcb package dependent, gEDA/gaf conventions exist to make this attribute as portable as possible, allowing for easy collaboration and sharing between users. See the *Footprint naming conventions* in the *gEDA/gaf Symbol Creation Document*.

If the symbol does not have a footprint, then the value of **footprint**= should be set to none. If the footprint must be overridden in a schematic, then the value of **footprint**= should be set to none. If the footprint is not known, then the value of **footprint**= should be set to unknown.

3.13 documentation

documentation=documentation_locator may exist somewhere in the symbol and be made invisible. This attribute is used by gschemdoc to find relevant documentation for the symbol, or rather, the device or component associated with the symbol.

Attach this attribute just like the **device**= (3.1) attribute. This is a free standing or floating attribute.

documentation_locator is either the base filename of the documentation, or it is the complete Internet URL (Uniform Resource Locator). If it is the filename, an attempt will be made to search for it in the local gEDA share directory named `documentation`.

Filename example: `documentation=sn74ls00.pdf`

URL example: `documentation=http://www-s.ti.com/sc/ds/sn74ls00.pdf`

4 Schematic only Attributes

4.1 netname

This attribute should be attached to a net object to give it a name. Multiple net names for connected net segments is discouraged. All nets which have the same value are considered electrically connected. This attribute is not valid inside symbols (as you cannot have nets inside of symbols).

Examples: `netname=DATA0_H` `netname=CLK_L`

4.2 source

The **source**= attribute is used to specify that a symbol has underlying schematics. This attribute is attached directly to a component.

This attribute should only be attached to instantiated components in schematics. Attach the attribute to a component and specify the filename (not the path) of the underlying schematic (like block.sch) for the value. The specified schematic must be in a source-library path. This attribute can be attached multiple times with different values which basically means that there are multiple underlying schematics.

Examples: `source=underlying.sch` `source=memory.sch`

5 Symbol and Schematic Attributes

5.1 refdes

This attribute is used to specify the reference designator to a particular instantiated component. It must be on ALL components which have some sort of electrical significance. This attribute can also be on the inside of a symbol (it will be promoted, i.e. attached to the outside of the symbol, if it is visible) to provide a default refdes value (such as U?).

Examples: `refdes=U1` `refdes=R10` `refdes=CONN1`

5.2 slot

This attribute is used to specify a slot for a slotted component. It should be attached to an instantiated component. This attribute can also be on the inside of a symbol (it will be promoted, i.e. attached to the outside of the symbol, if it is visible) to provide a default slot.

5.3 net

The **net**= attribute is used to create power/ground and arbitrary nets. Please see the netattrib Mini-HOWTO for more info. When this attribute is inside a symbol, it is used to create nets. When this attribute is attached to an instantiated component (in a schematic), then the **net**= can also be used to create new nets and can used to override existing nets.

5.4 value

Used mainly in the spice backend netlister to specify the value of the various elements. No translation is done on this, and it is placed as is into the netlist.

Examples: `value=1K value=10V`

5.5 symversion

The **symversion**= attribute is used to version the contents of symbols. Normally this attribute is not present, but once a symbol has been accepted into the main gEDA symbol library and there are changes to it, this attribute must be placed into the symbol file and properly incremented. The value of this attribute takes the following form:

`major.minor`

where major and minor are just plain integers (separated by a period). The major number is incremented when a symbol has some sort of a change which will break or might break an existing schematic. The minor number is only incremented when a cosmetic or very minor change is made to the symbol. The major and minor numbers are not coupled in any way, however, when making major version changes, the minor version number can be reset to zero.

If this attribute is inside of a symbol and that symbol is placed onto a schematic, then the **symversion**= attribute will be automatically promoted and attached to the outside of the symbol. During the load of the symbol from disk, the value of the **symversion**= inside the symbol file (if any) and the **symversion**= attached to the symbol (if any) are compared. If the values are the same then the placed symbol matches the disk symbol file, however if the values do not match, then libgeda will output a warning/error message (based on whether it is a major or minor version change).

New symbols should not receive this attribute at all. Only when the symbol is change should this attribute be placed into the symbol file and maintained. Users should not attach this attribute manually to instantiated symbols. This attribute should normally be made invisible when placed inside of a symbol file. This attribute is always promoted when it is found inside of a symbol (during component placement).

Examples: `symversion=1.1`

6 Obsolete Attributes

6.1 uref

The **uref=** attribute is obsolete and cannot not be used. It was used to provide the same information as **refdes** (5.1).

6.2 name

The **name=** attribute should not be attached or appear in any symbol. It is considered ambiguous. **name=** was never used by gEDA/gaf.

6.3 label

The **label=** attribute is obsolete and cannot be used. It was used to give nets names/labels and to label pins. The replacement attributes for this are **netname** (4.1) and **pinlabel** (3.9) respectively.

6.4 pin#

The **pin#=#** attribute is obsolete and cannot be used. It was used to provide sequence and number information to pins. The replacement attributes for this are **pinseq** (3.6) and **pinnumber** (3.7).

6.5 slot#

The **slot#=#** attribute is obsolete and cannot be used. It was used to provide slotting information to components. The replacement attribute for this is **slotdef** (3.11).

6.6 type

The **type=** attribute is obsolete and cannot be used. It was used to provide type information on pins. The replacement attribute for this is **pintype** (3.8).

6.7 email

The information in this attribute has been merged with **author** (3.4).

7 Document Revision History

July 14th, 2002	Created attributes.tex from attributes.txt.
July 14th, 2002	Updated doc to be in sync with post-20020527.
August 25th, 2002	Added obsolete type= attribute.
September 14, 2002	Added description= attribute. Minor fixes
October 7, 2002	Added doc= attribute; Egil Kvaleberg.
February 11, 2003	Added reference to footprint conventions.
February 23, 2003	Added author=, email=, and comment= attributes.
July 6th, 2004	Added symversion= attributes.