

Attributes to include the CAD netlist and components in Gerber.

Final, 2016.10

Please send your comments to gerber@ucamco.com

This specification was developed by Karel Tavernier

1 Preface

The goal of the Gerber CAD netlist is to facilitate upfront communication between the different parties involved in design, assembly and automation. The X2 attributes proposed include CAD netlists in Gerber fabrication data and allow to:

- Attach the component reference designator, pin number and net name to the component pads in the outer copper layers. This information is essential for a complete board display and for a complete board display. More importantly, the netlist provides a powerful checksum to guarantee PCB fabrication data integrity.
- Attach the netlist name to any conducting object on any copper layer. Lightweight viewers can then display netlists without the need for an algorithm to compute connectivity
- Attach the component reference to any object, e.g. to identify all the legend objects belonging to a given component, for example.

We have taken great care to make this new capability complete, yet simple and straightforward to implement and adopt – true to the Gerber hallmark of simplicity there are no needless bells and whistles.

The new capability is fully compatible with legacy applications and files as the new attributes do not affect the image. If you do not need the netlist information when processing a Gerber file, you simply skip the corresponding attributes. Furthermore, the CAD netlist attributes are not mandatory. While these attributes provide very useful fabrication data, if they do not benefit your application you need not expend any development effort on them to remain compatible.

This proposed attributes overlap to some degree with the venerable IPC-D-356A format. However, the Gerber CAD netlist attributes and 356 are not equivalent. Their design goals are completely different: 356 is designed as *standalone* input file for bare board electrical testers, the Gerber netlist attributes *add* net and component information to an existing job to facilitate communication between design, fabrication and assembly, and this in the simplest possible way. Accordingly, Gerber netlist attributes do not contain adjacency, test voltage or other electrical test specific information. and so on. Gerber files, even with netlist attributes, cannot be used as input for an electrical tester. On the other hand, Gerber netlist attributes contain information not found in 356, e.g. to which component legend objects belong, or the location of a net over all layers.

In the United States a 356 file is often included in CAD to CAM PCB fabrication data, an excellent practice providing a netlist to check the integrity of the copper image. Outside the US such files are rare. This is a usage for which 356 was not designed. The 356 file is not used directly on an electrical tester but read in CAM software that extracts the netlist, where it is a powerful check on the integrity of copper data. As 356 is designed as standalone file it contains approximate pad shapes, via locations and so on which in the context of complete fabrication data is superfluous, as is the electrical test specific context. As a result the 356 is much more complex to implement than needed for this quite restricted usage. This usage of 356 overlaps with the intended use of Gerber netlist attributes; for this usage the advantage of the Gerber netlist attributes is its simplicity, which results from its focused goal: it is an order of magnitude easier to understand and implement netlist attributes than 356; the Gerber netlist specification is just a few pages – even if it contains information which is not present in 356.

Our sincere thanks go to Jean-Pierre Charras for his insightful comments which contributed so much to this draft.

Please send your comments and criticism to gerber@ucamco.com .

Thank you

Karel Tavernier,
Managing Director,
Ucamco

2 Object Attributes

There are three types of attributes by the *domain* they attach to:

- ❑ File attributes attaching metadata to the file as a whole.
- ❑ Aperture attributes attaching metadata to an aperture or a region.
- ❑ Object attributes attaching metadata to graphics objects

During the processing of a Gerber file an *attribute dictionary* is maintained. Dictionary entries consist of the attribute name, its domain and its value. The attribute name is the key for the entry; the name must therefore be unique.

The current aperture dictionary is defined after each command in the file according to the following rules:

- ❑ Initially the attribute dictionary is empty
- ❑ File attributes are added or updated with the TF command
- ❑ Aperture attributes are added or updated with the TA command
- ❑ Object attributes are added or updated with the TO command
- ❑ Aperture attributes are deleted from it with the TD command

When an aperture or a graphics object is created all attributes with the corresponding domain in the dictionary at that moment are attached to the it. The attachment is static. Attributes remain attached and cannot be changed, whatever happens to the dictionary later.

An *object attribute* is attached to a graphics objects. When a D01, D03 or G36/G37 creates an object all object attributes in the attribute dictionary at that moment are attached to it. They remain fixed and cannot be changed.

The TO command adds an object attribute into the attributes dictionary. It has the same syntax as the TF or TA command:

<TO command> = %TO<AttributeName>[,<AttributeValue>]*%

NEW

<AttributeValue> = <Field>{,<Field>}

The attribute name must follow the naming syntax in section 3.6.5. This name cannot be used for any other attribute. The fields composing the attribute value must follow the string syntax in section 3.6.7 with the additional restriction: a field must not contain commas.

The value of an object attribute can be modified by a new TO command with the same attribute name.



Example:

```
%TO.N,Clk13*%
```

This command adds the .N attribute with value Clk13 to the dictionary.

The TD command deletes an attribute from the attributes dictionary. Note that TD does not affect attached to apertures and objects created earlier.

<TD command> = %TD[<AttributeName>]*%

The <AttributeName> is the name of the attribute to delete. If omitted, all attributes are deleted.

3 Component and Netlist Attributes

3.1 The Net Attribute (.N) NEW

The .N object attribute attaches a CAD netlist name to any conducting object. The attribute can be attached to objects on any copper layer. It indicates the object is part of the given net. The .N attribute is intended to allow quick visualization of nets and, more importantly, to define the CAD netlist.

Normally an object is fully connected and consequently belongs to a single net. However, if an object consists of different disconnected parts or is split in several disconnected parts by clear (LPC) objects it may belong to different nets. Then the .N attribute value must include all net names involved. It is recommended to avoid creating disconnected objects: one object, one net.

The syntax is:

<.N Attribute> = .N,<netname>{,<netname>}

| .N field | Usage |
|-----------|-------------------------------------------------------------------------|
| <netname> | The CAD net name. It can take any value conforming to the field syntax. |



Example:

```
%TA.aperFunction,Conductor*%  
%ADD21C,1*%           Create aperture 21, for conductive tracks  
...  
D21*                   Select aperture 21  
%TO.N,Clk3*%          Select net Clk3  
X5600000Y1200000D02*  Move to the start of a track  
X5600000Y1202500D01*  Draw the tracks with Clk3 is attached  
X5605000Y1205000D01*  
X5605000Y1220000D01*  
...
```

There are two reserved net names:

- 1) The empty string, defined by %TO.N,*% identifies objects not connected to a net, such as tooling holes, text, logos, pads for component leads not connected to the component circuitry.
- 2) The name N/C, defined by %TO.N,N/C*%, identifies a single pad net, as an alternative to giving each such net a unique name.

Except the reserved names all net names must be unique.

It is recommended to attach a .N attribute to all copper objects, also those without net. The absence of the .N attribute does not mean there is no net; the absence is therefore ambiguous.

Normally an object is fully connected and consequently belongs to a single net. However, if an object consists of different disconnected parts or is split in several disconnected parts by clear

(LPC) objects it may belong to different nets. Then the .N attribute value must include all net names involved. It is recommended to avoid creating disconnected objects: one object, one net.

3.2 The Component Attribute (.C)



The .C object attribute attaches the reference descriptor of a component to an object. It indicates that the object belongs to the given component. The attribute can be attached to objects on any layer. It is intended to identify e.g. which objects on a legend belong to which components.

The syntax is:

<.C Attribute> = .C,<component>

| .C field | Usage |
|-------------|-------------------------------------------------------------------------------------------|
| <component> | The component reference descriptor. It can take any value conforming to the field syntax. |



Example in a legend layer:

| | |
|----------------------|------------------------------------------------|
| D21* | Select aperture 13, used for component symbols |
| %T.O.C,R2*% | Select reference descriptor R2 |
| X5600000Y1200000D02* | Move to the start of the symbol |
| X5600000Y1202500D01* | Draw the symbol |
| X5605000Y1205000D01* | |
| X5605000Y1220000D01* | |

...

The attribute .C,R2 is attached to all tracks drawing the resistor symbol.

3.3 The Pin Attribute (.P) NEW

The .P object attribute attaches the reference descriptor and pin name of a component pin to an object. It indicates that the object belongs to the given layer. It is intended to identify e.g. which objects on a legend belong to which components.

The syntax is:

<.P Attribute> = .P,<component>,<pin>

| .P fields | Usage |
|-------------|------------------------------------------------------------------------------------------------|
| <component> | The component reference descriptor. It can take any value conforming to the field syntax. |
| <pin> | The pin name. It can take any value conforming to the field syntax. It is very often a number. |



Example :

```
D13*                               Select aperture for the pads for R5
%TO.P,R5,1*%
X5600000Y1200000D03*             The pad for R5, pin 1
%TO.P,R5,2*%
X5600000Y1202500D03*             The pad for R5, pin 2
```

The .P attribute can be attached to any pad belonging to a component and only those. The pin name must be a non-empty field, with one exception: if the pad is part of the component footprint but not connected to the component circuitry the name may be an empty field, e.g. defined by `%TO.P,U3,*%`; pads with an empty name are normally not part of a net and therefore also have an empty net name attached defined by `%TO.N,*%`.

Consequently, it can only be used in copper layers that carry the components. Typically, these are the outer layers and then the .P attribute can only be used in the outer layers. Embedded or etched components can be on inner layers and then these inner layers can also carry the .P attribute.

The .P attribute can be attached to *flashed pads only*. It can therefore not be used on painted pads.

A single component pad can consist of multiple flashes. Each flash then carries the same reference descriptor and pin number.

It is technically possible to create a single object representing multiple pads. For example, a single macro aperture can describe a complete component footprint. This is possible but *not allowed*. (It would be a neat way to make life miserable for CAM engineers though.)

3.4 The CAD Netlist

The CAD netlist describes which component pins are connected together. Pins are identified by their component reference descriptor and pin number or name. Nets are identified by the net name. Here is an example of a CAD netlist; The first line lists all pins of net Clk3.

```
Clk3: U1-4,U2-3,U5-9,U6-9,U7-9
Sig11: U1-3,U5-12
Data8: U2-4,U5-10,U6-10,U7-8,U8-1
GND: U1-1,U2-1,U3-8,U4-16,U5-16,U6-1,U7-8,U8-8
...
```

The CAD netlist links component pads to nets. It specifies the function of the PCB and is the basis for the layout.

In Gerber the CAD netlist is defined by attaching both the pin and the net attribute to each component pad. This defines a pad - net entry in the CAD netlist, and at the same time associates a pad location and shape to it. For instance, the first entry in the CAD netlist above is Gerber is given by the following sequence:

```
%TO.P,U1,4*%
%TO.N,Clk3*%
X...Y...D03*      The flash that creates the pad
...
```

If the .P attribute is present then the *complete* CAD netlist must be present, including all edge connectors test points and etched components. In other words, all the end points of the nets must be included, all the pads, *not only* the pads of the physical components that are part of the BOM. Although vias are part of a net, they are not component pads and cannot have a .P attached. Washer pads or any pads that are not part of an component cannot have a .P attached.

3.4.1 Etched Components

Etched components are embedded inductors, transformers and capacitors which are etched into the PCB copper. The following illustration shows two etched inductors.



They are identified by the .AperFunction attribute value 'EtchedComponent' attached to the aperture used to create them. See <Cross-reference>.

For the CAD netlist these are components like others: the net names are different on both sides. (However, for bare-board electrical test they may be conducting copper and connect the net on both sides.)

Etched components do not need and normally do not have pads. There is no .P associated with them. The net on each side is different however.

3.4.2 Benefits of Including the CAD Netlist.

For the assembly process the location and orientation of each component must be known. This can easily be extracted from the Gerber file.

The netlist and component names facilitate the communication between the parties involved in design and fabrication. Viewers show more complete PCB information.

More importantly, the netlist information dramatically increases the security of the design to fabrication data transfer. If the image CAM reads from a Gerber file significantly differs from the image intended by CAD, due to bugs, operator errors or transmission errors, the inevitable result is scrap. Such a difference in image results in a difference in board netlist, which can be detected by comparing the calculated board netlist with the supplied CAD netlist. The CAD netlist therefore provides a very powerful redundancy check against image errors. To be precise, the following assert must be valid:

- Interpret all N/C's as unique names
- Flashes with the same reference descriptor and pad are deemed connected
- Etched components are removed before connection is calculated
- Assert 1: pads with the same net name must be connected
- Assert 2: pads with different net names must be isolated.
-

Lastly, a bare board PCB fabricator is expected to perform an electrical test on the bare PCB and guarantee the PCB conforms to the CAD netlist. It then is logical to provide him with the netlist to test against. Without providing the netlist the fabricator is expected to reverse engineer the netlist and must test against a reverse engineered net – hardly a secure procedure.

3.4.3 IP Considerations

Net names such as Clk13 provide information about the design. This may be a concern. A solution is to replace the meaningful names with obfuscated names such as a sequential number. This still allows to compare the design netlist with the image netlist as a redundancy check – meaningful names are not needed for that. The obfuscated names are a little less convenient when communicating between creator and receiver of the Gerber file, but both can still identify the same net as long as the creator can identify the net corresponding to the obfuscated name he created. Obfuscated names are sometimes a sensible balance between IP protection and data transfer security.

It is sometimes alleged that even a net list with obfuscated names pose an IP security risk as it still shows the connections between the pads. This is an obvious fallacy as the connections between the pads can be worked out from the image. In fact, if this were not true, a fabricator would be unable to perform a bare board electrical test without netlist information.

4 Revisions

Final 2016.10

This public revision process is now closed and the specification is now final and can be used. The text will be merged with the main specification in an upcoming revision.

No material change since the previous revision.

Rev 2016.09

Clarified positioning in the preface. No changes in the specification itself which is considered finalized and will be incorporated in a future revision of the main Gerber specification.

Rev 2016.08.21

Improved example of the netlist on an etched component.

Rev 2016.08.19

Clarifications about unused and non-connected pads triggered by remarks from Jean-Pierre Charras.

Rev 2016.08.10

Replaced the combined .CN attribute by the .CN, as suggested by Wim De Greve; more general and orthogonal. Define objects without net with the empty net name, %TO.N,*%

Rev 2016.08

Corrections and clarification based on remarks by the initial distribution list.

Rev 2016.07

Initially created by Karel Tavernier on a long and boring flight. Jean-Pierre Charras provided very important insight and input. Remco Poelstra suggested to include the information carried by the .C attribute.

5 Copyright

© Copyright Ucamco NV, Gent, Belgium

All rights reserved. No part of this document or its content may be re-distributed, reproduced or published, modified or not, in any form or in any way, electronically, mechanically, by print or any other means without prior written permission from Ucamco.

The information contained herein is subject to change without prior notice. Revisions may be issued from time to time. This document supersedes all previous versions. Users of the Gerber Format[®], especially software developers, must consult www.ucamco.com to determine whether any changes have been made.

Ucamco developed the Gerber Format[®]. The Gerber Format[®], this document and all intellectual property contained in it are solely owned by Ucamco. Gerber Format[®] is a Ucamco registered trade mark. By publishing this document Ucamco does not grant a license to the intellectual property contained in it. Ucamco encourages users to apply for a license to develop Gerber Format[®] based software.

By using this document, developing software interfaces based on this format or using the name Gerber Format[®], users agree not to (i) rename the Gerber Format[®]; (ii) associate the Gerber Format[®] with data that does not conform to the Gerber file format specification; (iii) develop derivative versions, modifications or extensions without prior written approval by Ucamco; (iv) make alternative interpretations of the data; (v) communicate that the Gerber Format[®] is not owned by Ucamco or owned by anyone other than Ucamco. Developers of software interfaces based on this format specification commit to make all reasonable efforts to comply with the latest specification.

The material, information and instructions are provided AS IS without warranty of any kind. There are no warranties granted or extended by this document. Ucamco does not warrant, guarantee or make any representations regarding the use, or the results of the use of the information contained herein. Ucamco shall not be liable for any direct, indirect, consequential or incidental damages arising out of the use or inability to use the information contained herein. No representation or other affirmation of fact contained in this publication shall be deemed to be a warranty or give rise to any liability of Ucamco. All product names cited are trademarks or registered trademarks of their respective owners.