GDSII - ASCII Utilities Reference Guide v. 1.2.

Table of Contents

- GDSINFO Syntax 2
- GDS2ASC Syntax 3
- ASC2GDS Syntax 4
- ASCII GDS Format 5

GDSINFO Syntax

GDSINFO provides basic information found in GDSII file. This information is written to the output file if it is specified, or directed to the screen otherwise.

EXAMPLE

> gdsinfo -q test.gds

GDS version:	3	
Library name:	test	
Units: 0.001	1e-009	
Total number of	structures:	3
Total number of	layers:	2

List of unreferenced structures:

Cell

List of all structures:

Name	Layers	Inst	Rects Fla	tRects	Polys Fla	tPolys
Cell	0	2	0	2	0	5
Unit1	2	0	1	1	3	3
Unit2	2	0	1	1	2	2

End of report

Report contains number of distinct layers in each cell specified in Boundary records, number of instances, number of rectangles defined in the cell, number of rectangles in the flattened cell, number of polygons in the cell, and number of polygons in the flattened cell.

GDS2ASC Syntax

Usage: gds2asc [OPTIONS] <gds file> [<output file>]

OPTIONS (professional versions only):

-h	show usage
-d	suppress logo and information messages
-cell = cellname	convert only cellname cell (default all)

GDS2ASC reads the GDSII file, converts it to ASCII format, and writes ASCII file to the output file, if provided, or to the screen otherwise. If you are interested in a single cell in the GDSII file, you can use -cell option to convert only that named cell. This can significantly speed up the conversion process. If -cell option is not specified, the whole GDSII file will be converted. During the conversion polygons that represent rectangles are converted to RECT records.

ASC2GDS Syntax

Usage: asc2gds [OPTIONS] <ascii file> <output file>

OPTIONS (professional versions only):

-h	show usage
-q	suppress logo and information messages
-cell=cellname	convert only cellname cell (default all)

ASC2GDS reads ASCII representation of GDSII file and converts it to binary GDSII representation. You can use -cell option to convert a single named cell. During the conversion rectangles specified in RECT records are converted to GDSII polygon records.

ASCII GDSII format

ASCII representation closely follows GDSII binary format. Refer to the GDSII binary format reference for explanation of the fields and valid ranges of the values. ASCII format doesn't restrict the values to the valid GDSII ranges, so the values outside of the valid ranges will still be converted to the binary representation.

BNF description of ASCII GDSII format:

StreamFormat ::=
 Header
 Bgnlib
 [Libdirsize]
 [Srfname]
 [Libsecur]
 Libname
 [Reflibs]
 [Fonts]
 [AttrTable]
 [Generations]
 [FormatType]
 Units
 {Structure}*
 EndLib

The following characters are used as separators: ' '(space), '\t', '\r', '\n'. Multiple separators are equivalent to a single separator. ASCII format is case sensitive.

Header

Header ::= "VERSION" number

A header record contains VERSION keyword followed by the version number.

EXAMPLE

VERSION 3

Bgnlib and Libname

Bgnlib ::= "LIBRARY" '[' date time ',' date time ']'
Libname ::= string

Bgnlib record starts library description. It consists of LIBRARY keyword followed by the date and time of the last modification and time of the last access.

Libname record is a string which specifies the library name.

EXAMPLE

LIBRARY [2006/1/1 10:00:00, 2006/1/1 10:30:00] TestLibrary

Units

Units ::= "UNITS" double double

Units record contains two double size real numbers, which are the size of database units in user units and the size of a database unit in meters.

EXAMPLE

UNITS 0.001 1e-009

Structure

```
Structure ::=
        "STRUCT" '[' date time ',' date time ']'
        StrName
        [StrClass]
        {Element}*
        "ENDSTR"
Element ::= ( Boundary | Path | Sref | Aref | Text | Node | Box ) {Property}* ';'
Property ::= "PROP" number string
EXAMPLE
STRUCT [2006/1/1 10:00:00, 2006/1/1 10:30:00] TestStructure
   BOUNDARY 0 0
        RECT (-10000 -1000, 10000 1000)
   ;
ENDSTR
        Xy
Xy ::=
        "RECT" '(' number number ',' number number ')'
         "XY" number '(' {number number, ','}+ ')'
        | '(' number number ')'
Xy records are used to describe a rectangle, an arbitrary number of points, or a
single point.
Rectangles have their coordinates listed in this order: left, bottom, right, top.
XY record contains the number of points followed by X and Y coordinates of each
point separated by comma.
A single point contains X and Y coordinates.
A rectangle can be represented either by using a RECT record or a polygon XY record.
EXAMPLE
Rectangle:
RECT (-1000 -100, 1000 100)
The same rectangle as a polygon:
XY 5 ( -1000 -100, 1000 -100, 1000 100, -1000 100, -1000 -100)
```

A single point: (1000 0)

Boundary

Path

Path ::= "PATH" [Elflags] [Plex] Layer Datatype [Pathtype] [Width] [Bgnextn]
[Endextn] Xy
Pathtype ::= "PT" '=' number
Width ::= 'W' '=' number
Bgnextn ::= "BGNEXTN" '=' number
Endextn ::= "ENDEXTN" '=' number

Path record describes a path which type is specified in Pathtype record.

EXAMPLE

```
PATH 1 0 PT=4 W=200 BGNEXTN=100 ENDEXTN=100
XY 4 (-1000 -1000, 1000 -1000, 1000 1000, -1000 1000);
```

```
Sref
```

Sref ::= "SREF" [Elflags] [Plex] Sname [Strans] Xy
Strans :: = "STRANS" '=' number [Mag] [Angle]
Mag ::= 'M' '=' double
Angle ::= 'A' '=' double

Sref record describes a reference to another structure (an instance.) Sname is the name of the structure that this record refers to. Strans record describes transformation, where Mag is used to specify magnification and Angle is used to specify rotation. Xy record is always a single point.

EXAMPLE

```
SREF TestStructure (100000 0)
;
```

Aref

Aref ::= "AREF" [Elflags] [Plex] Sname [Strans] Colrow Xy
Colrow ::= "COLROW" '=' number ':' number

Aref record specifies array reference. Transformations have the same meaning as in Sref record. Additionally Aref record contains Colrow record in the form <columns> : <rows>.

EXAMPLE

```
AREF TestStructure COLROW= 5:8 XY 3 (0 0, 1000 0, 0 1000);
```

Text

```
Text ::= "TEXT" [Elflags] [Plex] Layer Textbody
Textbody ::= Texttype [Presentation] [Pathtype] [Width] [Strans] Xy '"' string '"'
Texttype ::= "TYPE" '=' number
```

Text record specifies a text string. ASCII format allows using any character string in quotation marks ('"'). GDSII binary format has additional limitations. ASC2GDS and GDS2ASC conversion tools do not impose those limitations.

EXAMPLE

```
TEXT 2 TYPE=0 (1000 0) "somestring"
;
```

Node

Node ::= "NODE" [Elflags] [Plex] Layer Nodetype Xy Nodetype ::= "NT" '=' number

Node record specifies a node.

EXAMPLE

NODE EF=0 PLEX=2 3 NT=5 (5 10);

Other records

```
Libdirsize ::= "LIBDIRSIZE" number
Srfname ::= "SRFNAME" string
Libsecur ::= "LIBSECUR" {number}+
Reflibs ::= "REFLIBS" {string}+
Fonts ::= "FONTS" {string}+
AttrTable ::= "ATTRTABLE" {string}+
Generations ::= "GENERATIONS" number
FormatType ::= Format [{Mask}+ EndMask]
Format ::= "FORMAT" number
Mask ::= "MASK" string
StrName ::= string
StrClass ::= "STRCLASS"
Box ::= "BOX" [Elflags] [Plex] Layer Boxtype Xy
EndLib ::= "ENDLIB"
```