ViewDraw file format

Description of ViewDraw (DxDesigner) file format for schematic symbols and schematic drawings.

Introduction

ViewDraw stores information in a line-oriented fashion. The beginning of each line declares the item it will describe, and the remainder of the line carries the attributes of the item. In the beginning of each line there is a letter indicating the object type.

Format structure

```
|-V
|
|-K
|
|-F
|
|-R
|
|-Y
|
|-D
|
|-Z
|
|-i
|
|-P
| |-Q
| |
| |-L
| | |-Q
| |
| |-A
|   |-Q
|
|-U
| |-Q
|
|-T
| |-Q
|
|-a
| |-Q
|
|-b
| |-Q
|
|-c
| |-Q
|
|-l
```

```
| |-Q
|
|-E
```

Line description

| (pipe symbol) - comment line.

Format: **|** ***comment text***

Example:

```
| my comment
```

This line may not appear before **V** and **K** lines. Comments are **not** saved when file is re-written by DxDesigner.

+ (plus symbol) - continuation of previous line.

Format: **+** ***<line continuation>***

Example:

```
+ 100 80 15 5
```

Lines that may span more than one line:

- l - line
- A - attribute
- I - instance (if a long abbreviated REFDES exists)
- L - label
- T - text

V - version

Format: **V** ***InternalVersionNumber***

Example:

```
V 50
```

Must be the first line of the symbol file.

note: ViewDraw 8.0.0 Aug 13 2001 denotes newly created files as **V 51**. New "high-precision" (metric) symbols (since EE2007.8) have **V 54**.

K - license

Format: **K *MagicNumber OriginalName***

Example:

```
K 199811641700 resistor
```

Must be the second line of the symbol file.

MagicNumber is created from the ViewLogic license and the file name. Calculation algorithm is unknown. If you change this line, you can break your file. This is also linked to the ***InternalVersionNumber***. The thing is that a long long time ago when ViewLogic reigned there was a licensed access to symbol libraries and ***MagicNumber*** was a way to control it.

OriginalName is a name of the symbol file when it was originally created with the GUI tool itself. If you copy stuff from other schematics or old symbols you'll get those old names in the new symbol. It's sane to match ***OriginalName*** to the current symbol file name.

F - case preservation

Format: **F Case**

Example:

```
F Case
```

Show if Case Preservation option was enabled for this symbol. Also affects the V line - version will be incremented to at least 5.3 (i.e. **V 53**)

R - timestamp

Format: **|R *Timestamp of Component***

Example:

```
|R 14:42_9-26-01
```

Timestamp is embedded in the schematic and symbol since EPD 2.0. It includes pipe character (comments) so older versions previous to EPD 1.0 ignore it. You might remove the timestamp from the symbol so that the schematic always appears current.

Timestamp of Component has the following format: *hour*:*minute*_*month*-*day*-*year*

Y - item type

Format: **Y *ItemType***

Example:

```
Y 1
```

ItemType

D - symbol block size

Format: **D *Xmin Ymin Xmax Ymax***

Example:

```
D 0 0 60 40
```

Determines the size of the symbol block (bounding box). The min and max values are the numeric values giving the symbol block size available on the "properties" pop-up menu.

Z - sheet size

Format: ***Z SheetSize***

Example:

```
Z 10
```

SheetSize

i - object counter

Format: **i *MaxObj***

Example:

```
i 6
```

MaxObj maximum used used object number (N #, I #, C #, X #, ...). Always incremented, deleted net's and other object's numbers are abandoned. There should not be any object with number greater than this one.

U - unattached attribute

Format: **U *XPos YPos FontSize RotMir Orientation Visibility Name(=Value)***

Example: U 0 -10 14 0 3 0 REFDES

Unattached attribute available under the "properties" pop-up menu. Unattached attribute is not associated with any net, bus, component, pin or component pin. It belongs just to the symbol or schematic in which it is present.

RotMir

Orientation

Visibility

P - symbol pin

Format: **P *InternalPinNumber Xend Yend Xbeg Ybeg Rot Side Inversion***

Example:

```
P 1 0 20 12 20 0 2 0
```

This line is specific to symbols. ***InternalPinNumber*** is a unique numerical ID of pin. Numbering starts at 1 and it should not be greater than ***MaxObj***. ***Xend***, ***Yend***, ***Xbeg***, ***Ybeg*** - end and begin of line designating pin. Right and left side reverse this so when Invert is selected, the bubble goes on the inside edge of the symbol. Endpoint (***Xend***, ***Yend***) which is furthest from the device must be listed first (actually this point must be on the border of symbol block).

Side

Inversion

L - label

Format: **L *Xpos Ypos FontSize RotMir Orientation Locality Visibility Inversion Name***

Example:

```
L -3 25 10 0 2 0 0 0 N1
```

Label for the previous item (pin, net, component etc.).

RotMir

Orientation

Locality

Visibility

Inversion

A - attached attribute

Format: **A *Xpos Ypos FontSize RotMir Orientation Visibility Name(=Value)***

Example:

```
A -3 10 10 0 7 0 PINTYPE=BI
```

Directly follows the item (component, pin, net or bus) it belongs to.

RotMir

Orientation

Visibility

T - text

Format: **T *Xpos Ypos FontSize RotMir Orientation Text***

Example:

```
T 0 0 14 0 3 mytext
```

RotMir

Orientation

b - box

Format: **b *Xmin Ymin Xmax Ymax***

Example:

```
b 14 16 50 28
```

Determines the size of a box drawn on the screen (inside the symbol block).

l - line

Format: **l *SegNum X1 Y1 X2 Y2 ... Xn Yn***

Example:

```
l 2 6 4 18 32
```

SegNum tells how many endpoints the line has. So it always will be that ***SegNum*** = n and n >= 2. May span multiple lines in symbol file.

a - arc

Format: **a Xend Yend Xon Yon Xbegin Ybegin**

Example:

```
a 18 32 0 19 22 20
```

- The first XY pair is the end of the arc
- The middle XY pair is any other point on the arc
- The last XY pair is the beginning of the arc

c - circle

Format: **c *Xpos Ypos* r**

Example:

```
c 8 34 4
```

Xpos and **Ypos** define center and **r** defines radius

Q - font, color, style

Format: **Q *Color FillStyle LineStyle***

Example:

```
Q 2 4 2
```

Defines font, line type and color definition. Affects the previous object.

Color

Fillstyle

Linestyle

I - component

Format: **I *ComponentID [LibAlias:]SymbolName SheetNumber Xpos Ypos Orientation Scale RefDes***

Example:

```
I 3 analog_ic:LM7805 1 550 480 0 1 '
```

Refers to an instance of the named symbol on the schematic. Must never appear in symbol file.

ComponentID - unique internal id of this component instance.

[LibAlias:]SymbolName - symbol name with optional library name

SheetNumber - sheet number of the symbol

Xpos Ypos - component instance origin

RefDes - abbreviated REFDES attribute value of the component instance. The single quote character (**'**) is used if no abbreviated REFDES exists for the component instance.

Orientation

Scale - scale factor of the component instance.

C - connected pin

Format: **C net_id joint_id pin_id pin_num**

Example:

```
C 131 4 2 0
```

Identifies a component pin that is attached to a net or bus. Component pins immediately follow components (**I** line) in ASCII file. net_id is the internal net id attached to this pin. joint_id refers to the position whithin the list of joints for the net. The joint in this position is the specific joint of the net attached to this pin. pin_id is the internal pin id of the pin (of the underlying symbol for this component) associated with this component pin.pin_num is the value of the abbreviated attribute associated with the component pin.

X - not connected pin

Format: **X *PinID PinNum***

Example:

```
X 23 0
```

Not connected pin of the above component. Precedes "A" attribute line related to the respective pin number.

Defines a component pin that is not attached to any net or bus. Component pins immediately follow component (**I** key) in ASCII file.

PinID - internal pin id of the pin (on the underlying symbol for this component) associated with this component pin.

PinNum - the value of the abbreviated attribute associated with the component pin.

N - net

Format: **N *NetNumber***

Example:

```
N 8
```

Defines beginning of a net or bus.

J - joint

Format: **J *Xpos Ypos JointType***

Example:

```
J 530 540 3
```

Defines joint that belongs eз a net or bus. Joints are referenced by one or more net or bus segments. Joint identifies a vertex of the net. The order in which joints appear after a net or bus is used by **S** and **B** keys.

JointType

S - segment

Format: **S *LowJointID HighJointID***

Example:

```
S 2 1
```

Defines net segment. Net segment belongs to a net and connects 2 joints which are part of that net.

LowJointID refers to the position in the preceding list of joints of one of the joints belonging to this segment.

HighJointID refers to the position in the preceding list of joints of the other of the joints belonging to the segment.

If the segment is horizontal the referenced high joints **X** will always exceed the referenced low joints **X**. If the segment is vertical the referenced high joints **Y** will always exceed the referenced low joints **Y**.

An attribute or label associated with a net actually belongs to a net segment and will follow the **S** key in symbol file.

B - bus

Format: **B *low_joint_id high_joint_id***

Example:

```
B 2 4
```

Defines bus segment. Bus segment belongs to a bus and connects 2 joints which are part of that bus. **low_joint_id** refers to the position in the preceding list of joints of one of the joints belonging to this segment. **high_joint_id** refers to the position in the preceding list of joints of the other of the joints belonging to the segment. If the segment is horizontal the referenced high joints **X** will always exceed the referenced low joints **X**. If the segment is vertical the referenced high joints **Y** will always exceed the referenced low joints **Y**. An attribute or label associated with a bus actually belongs to a bus segment and will follow the **S** key in symbol file.

E - end of file

Format: **E**

Example:

```
E
```

Structure semantics

Net description

```
N-record
J-record1
J-record2
[...J-record(n)]  - min 2 J-records
[:
S-record - at least one occurrence (one less than # of J-records)
[A-record(s)] - optional Attribute(s) and
[L-record(s)] - optional Label(s) associated with above Segment
:]
```

Terminated by another structure (starting with N-record, I-record, L-record, T-record, ...)

Component Instantiation

```
I-record
[A-record(s)]
[:
C-record | X-record - at least one of these is "a must" depending on whether
connected or disconnected from net
[A-record] - optional attribute related to above connection point
:]
```

Terminated by another structure (starting with N-record, I-record, L-record, T-record, ...)

Values of some fields

ItemType

- 0 = Composite
- 1 = Module
- 3 = Annotate
- 4 = Pin

SheetSize

- 0 = A
- 1 = B
- 2 = C
- 3 = D
- 4 = E
- 5 = A4
- 6 = A3
- 7 = A2
- 8 = A1
- 9 = A0

- 10 = Z (defined by user)

Visibility

- 0 = Invisible
- 1 = Visible
- 2 = Name
- 3 = Value

RotMir

- 0 = Not rotated
- 1 = Rotated 90 degrees
- 2 = Rotated 180 degrees
- 3 = Rotated 270 degrees
- 4 = Mirror horizontal
- 5 = Mirror horizontal rotated 90 degrees
- 6 = Mirror vertical (Mirror horizontal rotated 180 degrees)
- 7 = Mirror horizontal rotated 270 degrees

Orientation

- 1 = Upper left
- 2 = Middle left
- 3 = Lower left
- 4 = Upper center
- 5 = Middle center
- 6 = Lower center
- 7 = Upper right
- 8 = Middle right
- 9 = Lower right

Note: imagine numeric keypad rotated by 90 deg

```
1 4 7
2 5 8
3 6 9
```

Side

- 0 = Top
- 1 = Bottom
- 2 = Left
- 3 = Right

Locality

- 0 = Local

- 1 = Global

JointType

- 0 - Joint all alone.
- 1 - Dangling joint with one attached segment (dangling net, square is shown).
- 2 - Joint attached to a pin with one attached segment.
- 3 - Joint on a corner with 2 segments.
- 4 - Straight joint with 2 segments.
- 5 - Solder joint with 3 or more segments (dot is shown).
- 6 - Bus joint all alone.
- 7 - Dangling bus joint with one attached segment
- 8 - Bus joint attached to a pin with one attached segment.
- 9 - Bus joint on a corner with 2 segments.
- 10 - Straight bus joint with 2 segments.
- 11 - Solder bus joint with 3 or more segments (dot is shown).

Inversion

- 0 = Not inverted
- 1 = Inverted

Color

- 0 = Black
- 1 = Blue
- 2 = Green
- 3 = Cyan
- 4 = Red
- 5 = Magenta
- 6 = Brown
- 7 = Lt. Gray
- 8 = Gray
- 9 = Lt. Blue
- 10 = Lt. Green
- 11 = Lt. Cyan
- 12 = Lt. Red
- 13 = Lt. Magenta
- 14 = Yellow
- 15 = White

Fillstyle

- 0 = Hollow
- 1 = Solid
- 2 = Grey92

- 4 = Grey50
- 6 = Grey08
- 7 = Grey04
- 8 = Diagdn2
- 11 = Diagdn1
- 13 = Diagup2
- 16 = Diagup1
- 19 = Horiz
- 21 = Vert
- 22 = Grid2
- 23 = Grid1
- 24 = X2
- 25 = X1

Linestyle

- 0 = Solid
- 1 = Dash
- 2 = Center
- 3 = Phantom
- 4 = Big dash
- 5 = Dot
- 6 = Dash-Dot
- 7 = Medium dash

Tips

- Use DxD>Project>Block>Detail to "decode" DxD symbol text format
- Line continuation is effected by placing a '+' symbol in the next line, inserting one space, and then continuing with the information from the previous line. (**add example**)