

A Comprehensive CIF Test Set

by

Stephen Trimberger

Technical Report #4022

January 17, 1980

Computer Science Department  
California Institute of Technology  
Pasadena, California 91125

Silicon Structures Project

sponsored by

Burroughs Corporation, Digital Equipment Corporation,  
Hewlett-Packard Company, Honeywell Incorporated,  
International Business Machines Corporation,  
Intel Corporation, Xerox Corporation,  
and the National Science Foundation

The material in this report is the property of Caltech, and is subject to patent and license agreements between Caltech and its sponsors.

Copyright. California Institute of Technology, 1978

### A Comprehensive CIF Test Set

The test set consists of several CIF2.0 files which are used to test CIF processing software. There are six CIF files in the CIF test set. They are: BLTTST.CIF, EXTEST.CIF, EXTBAD.CIF, NASTY.CIF, PLA.CIF and TEST.CIF. All of the files can be found on <SSPDOC>.

This document describes each test briefly, its purpose and some specifics on how it tries to test. This document describes some of the tests in detail and explains in some cases where to look for. The CIF test files and this document will be evolving over the next few months, so stay tuned to this printer.

#### BLTTST.CIF

BLTTST is a tester for "bloating", the sizing of graphic features in PAT. BLTTST contains a few graphic primitives and allows the user to try different bloating values on them, observing the error messages or lack of them, and checking the resulting features to ensure their correctness.

BLTTST has one symbol with five graphic features inside it: a box on layer ND and a circle on NM inscribed inside it (before any bloating). There is a triangle on NP which rests inside the square. There is a NI wire which runs along the inside of the box and a wire on NG which runs around the outside of the box.

When running the test, apply bloating to all layers except NG, which should be used as calibration. Alternative you may leave all layer unchanged except ND and watch the

relationship of the box to the other graphics.

### EXTEST.CIF and EXTBAD.CIF

EXTEST and EXTBAD are used to test the Caltech user extensions in CIF. EXTEST checks good use of the user extensions, EXTBAD tests bad use of the user extensions. The user extensions in use are:

- 1 "Hello there"; Parse-time comment text with comment string.
- 2 "Vdd line" T 40, 40; Text on the plot with text string and x,y position compatible with transformation matrix.
- 5 40 40 "VDD Connection"; Connection point with position and optional connector name.
- 6 3 4 4 100 300; (call sym#3 in an array 4x4 spaced 100x300); Two-dimensional array call with cell to be replicated, repetition counts and repetition spacing.
- 9 ShiftReg; Cell name text.

The BNF for these user extensions is as follows:

```
remark      = "1" [ sep ] "" { any character except "" } ";",
plottext    = "2" [ sep ] "" { any character except "" } "T" point ";",
connector    = "5" point "" [ { any character except "" } ] ";",
arraycall   = "6" [ sep ] integer point point ";",
cellname    = "9" [ sep ] { any character except ";" } ";",
```

Point, integer and sep are as defined in CIF.

### NASTY.CIF

NASTY is a vicious test for a CIF2.0 parser. It is constructed of several bad and questionable constructs.

You can expect it to evoke every error message in your parser, and/or some in your run time system. If your CIF handler can take NASTY, it is very robust. Many hackers spent lots of time making NASTY truly nasty. To give you an example of their abilities, NASTY blasts CIF20P into space dust. NASTY is not for the faint-hearted. A listing of the error messages produced on one sample run of NASTY is appended to this document. This should give you an idea of what to look for. There are a lot of comments in NASTY describing what feature each test is checking. You should use NASTY only after your program has passed TEST (below). Be armed with a source listing of your program and a listing of NASTY. Use it on somebody else's software (it hurts too much to see your own in such bad shape).

There are several tests in NASTY. In order of listing in the file, they are: bad command test, user extension test, no-layer-specified test, layer test, symbol test, symbol call test, graphic elements test, and termination test.

#### Bad Command Test

The bad command test checks commands that is not a legal CIF including legal, but strange commands, like the null command and illegal commands, like CIF command numbers and badly formed definition commands, a comment without a terminating semicolon and a comment with too many closing parentheses.

#### User Extension Test

The user extension test has a user extension with userText inside it. UserText may include "(" and ")" so this user

extension test has them. A watchful parse should check and flag unpaired parentheses inside a user extension because inclusion of that command inside a comment would result in a misaligned comment

#### No-Layer-Specified Test

This test tries to draw a box without a layer specification both inside and outside a cell, and again after the cell definition and after a bad layer specification. The first three attempts should generate errors, the fourth depends on the error handling of the bad layer command.

#### Layer Test

The layer test checks various ways to screw up a layer statement. They are: garbage after the layer specification, no layer name, garbage inside a name leading to an unrecognized name, a very long shortName and blanks inside a layer specification.

#### Symbol Definition Test

This checks all the myriad things that can go wrong in a symbol definition, including a=0, b=0, re-definition of symbols, deletion of a symbol inside another symbol, negative symbol numbers on a definition, definition of a symbol inside another one, a spurious DF command, and a very large symbol number (just to see if you use a symbol table).

### Symbol Call Test

This test includes calling a negative symbol number, recursive calls, both directly and indirectly, a call to an undefined symbol, bad rotation vector, badly formed transformation entries: mirroring rotation and translation, and calling a symbol after it has been deleted.

### Graphic Elements Test

The box test includes bad rotation vector, negative size, spurious minus sign in the position field, a box with huge position coordinates, an incomplete rotation vector, an incomplete box and extraneous characters following the box specification.

The wire test includes a no-point and a one-point wire. It has wires with duplicated points, a zero-width wire and a negative-width wire, and a wire with an unpaired coordinate.

The polygon test includes a zero-point, one-point and two-point polygon, a polygon with one point duplicated five times, a polygon with retraced segments, self-intersecting and a polygon with an unpaired coordinate.

The round flash test checks negative diameter, zero diameter, too few numbers and a comment shoved into the middle of the graphic element.

### Termination Test

The termination test puts the CIF end statement inside an open symbol definition. In addition, there is more data after the end. Both should be caught.

### PLA.CIF

PLA is a real chip designed by Alastair Thompson at Caltech. There is nothing very special about it. In fact, there are a lot of known bugs in it as well as some systematic errors caused by bugs in the design system which created the CIF. But it is an example of a real chip. PLA was made with no thought to testing CIF software, its indifference to CIF processing programs is the main part of its desirability. PLA is big enough to be a real test and short enough to be reasonably quick test. We have found enough bugs in our programs with PLA to warrant its inclusion unchanged in the CIF test set.

### TEST.CIF

TEST is an example of forthright CIF code to test a number of CIF parser and plotter properties. Test is designed to be a powerful test of many CIF parsing and plotting features. It contains only legal syntax, and strains it to the limit. Your parser may identify some questionable CIF constructs, but it should not identify any errors. The plot from Test is shown below and a detailed description of each test is included. Read this section entirely before running Test. If your parser refuses to accept Test, the reason may well be buried in the description of the particular section.

The plot from TEST looks like this:

P I C T U R E		L E G E N D	
		!	l mirror, rotation stack test
l	l	!	Q quadrant test
	+	!	+ DD bug test
l	l	!	o Def #0 test
	+ f f	!	F transformation test
l	l	!	f rotation test
	::: , , . ;}	!	: polygon test
l	l	!	, box test
Q		!	. round flash test
		!	; wire test

#### Mirror, Rotation Stack Test

This test checks all mirror and rotation combinations two calls deep. In order to get to those calls, though, you have to go through four other calls. Therefore, the maximum calling depth is six. Your CIF processing software should take this in stride. The figure output is an "L", transformed in all the various ways in sets of eight. The result should look like this:



p v d m	/ m	!	
b ? d d	w p	b = !	/ = +---
	v v	+ -	!
	v d		
		? = -+	w = !
/ m	/ m	!	----+
w p	w p	!	
v v	v v		
v d	v d	d = !	v = !
		!	+---
p p b ?		-+	
v p m d	/ m		
	? v	p = +-	m = ----+
	w w	!	!
	b w	!	
p m	/ ?		
m m	/ /		
d /	m b		
v w	v w		

If the result does not look like this, you have a problem in either your transformation routines for mirroring and rotating, or your matrix operations for calling cells. To determine which one, check the results of the transformation tests.

#### Quadrant Test

CIF allows coordinates to be specified in any quadrant. This test draws a red (LNP) square (as a polygon) around the origin. The polygon has one vertex in each quadrant. Your software should have no trouble with this polygon.

### DD Bug Test

If the figure in the upper right is a big green (diffusion layer) plus sign "+", then your software passed this test. If it is a big red (polysilicon) "-", then you have the Definition Delete bug. You should read the discussion in SSP File #2686, "The Caltech Intermediate Form for LSI Layout Description" carefully about the Definition Delete and call. Remember: Symbol references are resolved in the context in which they are called, not the context they are written. Therefore, you can throw out the data defining a cell after you encounter a DD statement which deletes that cell. A CIF file is meant to be plotted in one scan, all data outside symbol definitions is to be plotted immediately as it is found in the file. See also SSP File #2685 "Recent Clarifications of CIF2.0".

Both bars of the "+" are the same length. This can be used to check the aspect ratio of the output device.

### Def #0 Test

Definition #0 is a valid number. Your design system should allow it. If it does, there will be a little red box here. If not, you probably got an error in parsing.

### Transformation Test

There should be one (count it, one!) blue (metal) "F" next to the Def #0 Test. If there are nearby "F"s mirrored or rotated, then your symbol call transformation scanning is not working. If there is only one "F" misplaced, it may be

a problem with the matrix multiplication algorithm. If the rotation and mirroring test failed also, the latter is probably the case. If the rotation test next to this test failed, then there may be no problem with your transformations besides the rotation failure. If there are many "F"s displaced slightly from one another, the precision of your transformations is not great enough.

Above the "F", there should be on green horizontal bar. If the bar is blue, then your system incorrectly does not preserve the layer over a symbol call. This should be fixed.

#### Rotation Test

This test consists of four "F"s rotated in each quadrant. Each "F" should have the single bar closer to the origin than the two horizontal pieces of the "F". The "F" should face outward, away from all other "F"s in this test. None of the "F"s should appear mirrored.

#### Polygon Test

The polygon test consists of nine polygons, arranged in a three-by-three block as follows:

Pentagon	A-shape	M-shape
Square	E-shape	Two-blocks
Triangle	Plus	O-shape

There are some concave and some convex, acute, right and obtuse angles and one even touches itself. Your system should be able to plot them all without batting an eye.

### Box Test

The box test consists of one box with many rotations that you can give to such things. The boxes are listed below by their amount of rotation (in degrees):

```
135      atan(157/37)
90       315
45       270
0        225          atan(200/(-1))
0        180          0
```

If the lower right one is not there, your parser may have ignored it because it mistakenly thought the syntax was bad. Check the syntax in SSP File #2686, there need not be separators between keyletters and numbers. The box above that one can tell you if your raster-conversion algorithm suffers from jaggies.

### Round Flash Test

This test consists of five round flash commands as listed below by radius:

```
0
10
20
40
20
```

The top one, radius zero may or may not appear, as you wish, but your system must accept it as a legal input. If the bottom one is not there, your parser may have ignored

it because it mistakenly thought the syntax was bad. Check the syntax in SSP File #2686, there are quite a few valid separators.

#### Wire Test

The wire test consists of a large number of wires, testing the wire plotting code. The wires are:

1-point	
2 coincident points	
wire body touches self	
circular path	
intersecting path	extremely acute
path doubles back on itself exactly	very acute
path retraces self within width distance	acute angle
high slope wire	high angle
moderate slope wire	moderate angle
low slope wire	low angle
no slope 2-point wire	colinear

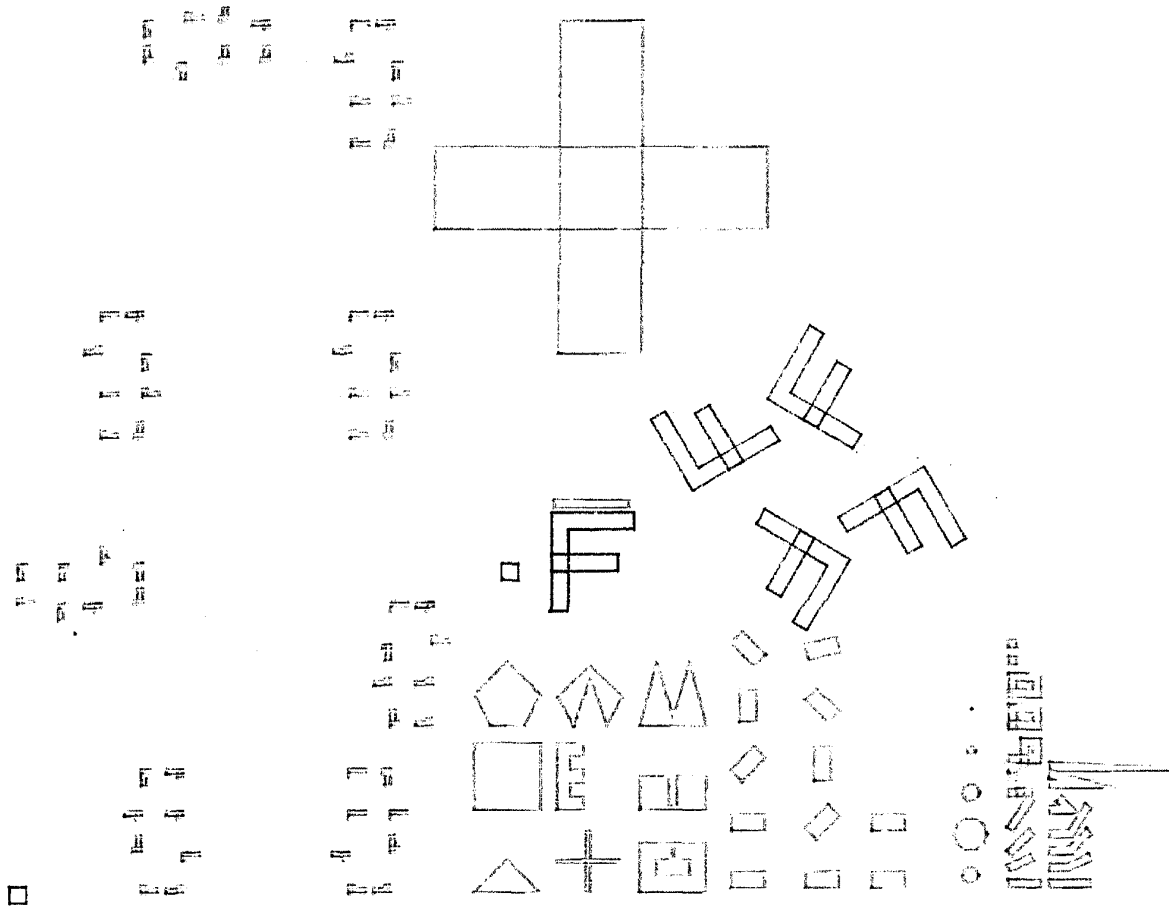
Much CIF software cannot handle the first wire. It should be caught as a special case and treated as a round flash (diameter=width). See SSP File #2686 for details. The second wire should be the same. The self-intersecting and self-touching wires should not be any problem for your software. Some raster conversion software leaves annoying holes where the wire overlaps itself. This should be avoided. Unless doubling-back is caught as a special case, it might show as an X-shape. Nearby retracing should be no problem. The lower left slope tests can give you an estimate of the accuracy of your output device.

The top of the second column provides much difficulty for most CIF software. If your program computes the polygon from the path by re-intersecting lines parallel to the path, this will appear as a very wide diamond shape, instead of a very-acute angle. Your program should detect the special case where a point is within WIDTH of the previous segment, and position points properly. Note that this is a separate check from the check for co-linear points. The second wire is a very acute wire. Again, if your merely re-intersect the wires, this will have a long point to the right. This could be disastrous in practice. The other wires in the second column give one feedback on the quality of the wire-plotting algorithm for many different angles.

#### Concluding Remarks

We believe that this test set is a good one. It tests many features of CIF that might not be adequately checked otherwise. This test set is the standard test of our software. We have not had a user-discovered bug in any of our CIF parsers since we first ran these tests.

incl. 14022.app



TEST CIF

```

A COMMAND THAT PROBABLY IS NOT UNDERSTOOD;
Error on Line #    9 at char pos  2
Error -- Unrecognized Command.  Skipping to semicolon.

DU; (do you know what this is?);
Error on Line #   10 at char pos  2
Error -- Unrecognized Def command.  Command ignored.

DU; (do you know what this is?);
Error on Line #   10 at char pos  2
Warning -- Semicolon missing.  Who cares?

DU; (do you know what this is?);
Error on Line #   10 at char pos  3
Error -- Unrecognized Command.  Skipping to semicolon.

D (mumble) D 300; (you should flag this);
Error on Line #   11 at char pos  3
Error -- Unrecognized Def command.  Command ignored.

D (mumble) D 300; (you should flag this);
Error on Line #   11 at char pos  3
Warning -- Semicolon missing.  Who cares?

D (mumble) D 300; (you should flag this);
Error on Line #   11 at char pos 12
Warning -- Semicolon missing.  Who cares?

D (mumble) D 300; (you should flag this);
Error on Line #   11 at char pos 14
Error -- Unrecognized Def command.  Command ignored.

D (mumble) D 300; (you should flag this);
Error on Line #   11 at char pos 14
Warning -- Semicolon missing.  Who cares?

D (mumble) D 300; (you should flag this);
Error on Line #   11 at char pos 17
Warning -- Unrecognized user command.  Command Ignored

(Well! can we do null commands? #2686 says no);;
Error on Line #   14 at char pos 48
Warning -- Null statement not allowed.  Statement Skipped.

(meanwhile, back at NASTY -- no semicolon)    (did you see that?);
Error on Line #   15 at char pos 47
Warning -- Semicolon missing.  Who cares?

(extra close paren, and one after the semicolon)); );
Error on Line #   16 at char pos 49
Warning -- Semicolon missing.  Who cares?

(extra close paren, and one after the semicolon)); );
Error on Line #   16 at char pos 50
Error -- Unrecognized Command.  Skipping to semicolon.

(extra close paren, and one after the semicolon)); );
Error on Line #   16 at char pos 54
Error -- Unrecognized Command.  Skipping to semicolon.

```



(you can have unbalanced parens in here, you know, causing disaster  
 Error on Line # 25 at char pos 46  
 Warning -- UserExtention text overran array. Text Truncated. Implementation  
 Limitation.

)))when you try to comment out a bunch of CIF.;  
 Error on Line # 26 at char pos 47  
 Warning -- Unrecognized user command. Command Ignored

B 10 20 30 40; (a box before a layer);  
 Error on Line # 32 at char pos 14  
 Error -- No layer specified before graphic command. Command Skipped.

B 10 20 30 40; (should be no defined layer);  
 Error on Line # 37 at char pos 18  
 Error -- No layer specified before graphic command. Command Skipped.

Layer mumble; (bad layer);  
 Error on Line # 38 at char pos 17  
 Error -- Short name must be digits and letters. Proceeding.

Layer mumble; (bad layer);  
 Error on Line # 38 at char pos 17  
 Error -- Layer specification not recognized. Layer unchanged.

B 10 20 30 40; (now can we do a box?);  
 Error on Line # 39 at char pos 18  
 Error -- No layer specified before graphic command. Command Skipped.

B 10 20 30 40; (this is still bad after setting a layer in a sym);  
 Error on Line # 44 at char pos 14  
 Error -- No layer specified before graphic command. Command Skipped.

Layer mumble; (bad layer);  
 Error on Line # 46 at char pos 13  
 Error -- Short name must be digits and letters. Proceeding.

Layer mumble; (bad layer);  
 Error on Line # 46 at char pos 13  
 Error -- Layer specification not recognized. Layer unchanged.

B 10 20 30 40; (now can we do a box? Your choice.);  
 Error on Line # 47 at char pos 14  
 Error -- No layer specified before graphic command. Command Skipped.

Layer NP 0 0 100 50; (junk after a layer selection);  
 Error on Line # 54 at char pos 10  
 Warning -- Semicolon missing. Who cares?

Layer NP 0 0 100 50; (junk after a layer selection);  
 Error on Line # 54 at char pos 20  
 Warning -- Unrecognized user command. Command Ignored

L mumble in lower case ; (no layer after all);  
 Error on Line # 55 at char pos 24  
 Error -- Short name must be digits and letters. Proceeding.

L mumble in lower case ; (no layer after all);  
 Error on Line # 55 at char pos 24

```

Error -- Layer specification not recognized. Layer unchanged.

Layer NgarbageM; (you should complain about this);
Error on Line # 56 at char pos 8
Error -- Layer specification not recognized. Layer unchanged.

Layer NgarbageM; (you should complain about this);
Error on Line # 56 at char pos 15
Warning -- Semicolon missing. Who cares?

Layer NgarbageM; (you should complain about this);
Error on Line # 56 at char pos 16
Error -- Unrecognized Command. Skipping to semicolon.

Layer VERYLONGLAYERNAME; (and this);
Error on Line # 57 at char pos 11
Warning -- Shortname for layer more than 4 chars. ShortName Truncated.

Layer VERYLONGLAYERNAME; (and this);
Error on Line # 57 at char pos 24
Error -- Layer specification not recognized. Layer unchanged.

Layer N I; (this aint legal either);
Error on Line # 58 at char pos 8
Error -- Layer specification not recognized. Layer unchanged.

Layer N I; (this aint legal either);
Error on Line # 58 at char pos 9
Warning -- Semicolon missing. Who cares?

Layer N I; (this aint legal either);
Error on Line # 58 at char pos 10
Error -- Unrecognized Command. Skipping to semicolon.

L N?!#; (not our favorite layer); (it should use N the without error);
Error on Line # 59 at char pos 4
Error -- Layer specification not recognized. Layer unchanged.

DD -1; (not legal, twerp! gotta be non-negative);
Error on Line # 67 at char pos 4
Error -- Negative number not allowed here. Taking positive part.

DS 2 0 0; (0/0 as the sclae of the symbol. Your system should flag it);
Error on Line # 71 at char pos 9
Error -- A or B is zero in symbol def. Set to 1.

DS 2 0 100; (deja-vu. Alerady defined. Just a=0 this time);
Error on Line # 76 at char pos 11
Error -- A or B is zero in symbol def. Set to 1.

DS 2 0 100; (deja-vu. Alerady defined. Just a=0 this time);
Error on Line # 76 at char pos 11
Warning -- Symbol previously defined. Re-defining symbol.

    DD 2; (inside a def, delete self);
Error on Line # 79 at char pos 6
Warning -- Def Delete inside symbol. Command carried out.

DS -3; (negative sym nums are not allowed.);
Error on Line # 84 at char pos 4

```

Error -- Negative number not allowed here. Taking positive part.

DS 3 100 1;

Error on Line # 90 at char pos 11

Warning -- Symbol previously defined. Re-defining symbol.

DS 4 100 1; (plus a little confusion);

Error on Line # 93 at char pos 2

Error -- Start Def inside a Symbol Definition. Previous Def Terminated.

DF; (a spare finish);

Error on Line # 96 at char pos 2

Warning -- Def End command before Def Start. Command ignored.

DF; (another spare finish);

Error on Line # 97 at char pos 2

Warning -- Def End command before Def Start. Command ignored.

C -3; (can you call this? We defined it above (or tried to));

Error on Line # 110 at char pos 3

Error -- Negative number not allowed here. Taking positive part.

C 11 T 0,0; (the recursion express);

Error on Line # 121 at char pos 11

Error -- Recursive symbol call. Call was skipped.

From source line # 113

CIF # 11 My # 6

C 11 T 0,0; (the recursion express);

Error on Line # 121 at char pos 11

Error -- Recursive symbol call. Call was skipped.

From source line # 118

CIF # 11 My # 6

CIF # 15 My # 7

C 13 T 0,0; (Call an undefined symbol);

Error on Line # 122 at char pos 11

Error -- Symbol in call not defined. Call was skipped.

From source line # 118

C 3 R 0 0; (bad rotation);

Error on Line # 125 at char pos 10

Warning -- Rotation direction must not be (0,0). Rotation deleted.

C 3 MN; (illegal mirror);

Error on Line # 126 at char pos 6

Warning -- Axis must be X or Y. Mirroring deleted.

C 3 MXY; (another bad mirror);

Error on Line # 127 at char pos 7

Warning -- Semicolon missing. Who cares?

C 3 MXY; (another bad mirror);

Error on Line # 127 at char pos 8

Error -- Unrecognized Command. Skipping to semicolon.

C 3 X 1500; (bad transform type);

Error on Line # 129 at char pos 5

Warning -- Semicolon missing. Who cares?

```
C 3 X 1500; (bad transform type);
Error on Line # 129 at char pos 6
Error -- Unrecognized Command. Skipping to semicolon.
```

```
C 3 T 100; (one-point translates are no good);
Error on Line # 130 at char pos 10
Error -- No number found. (using 0)
```

```
C 3 T 100 MX R 10 0; (the "MX R " is a legal blank!);
Error on Line # 131 at char pos 19
Warning -- Semicolon missing. Who cares?
```

```
C 3 T 100 MX R 10 0; (the "MX R " is a legal blank!);
Error on Line # 131 at char pos 55
Warning -- Unrecognized user command. Command Ignored
```

```
C3M()X; (we cant talk here!);
Error on Line # 132 at char pos 4
Warning -- Axis must be X or Y. Mirroring deleted.
```

```
C3M()X; (we cant talk here!);
Error on Line # 132 at char pos 5
Warning -- Semicolon missing. Who cares?
```

```
C3M()X; (we cant talk here!);
Error on Line # 132 at char pos 6
Error -- Unrecognized Command. Skipping to semicolon.
```

```
C3MZ F; (what do you mean we cant talk here?);
Error on Line # 133 at char pos 4
Warning -- Axis must be X or Y. Mirroring deleted.
```

```
C3MZ F; (what do you mean we cant talk here?);
Error on Line # 133 at char pos 10
Warning -- Semicolon missing. Who cares?
```

```
C3MZ F; (what do you mean we cant talk here?);
Error on Line # 133 at char pos 11
Error -- Unrecognized Command. Skipping to semicolon.
```

```
C 3; (...then use);
Error on Line # 137 at char pos 4
Error -- Symbol in call not defined. Call was skipped.
```

```
B -100 0 - 150 0; (negative size, cant have space bet. - and number);
Error on Line # 150 at char pos 7
Error -- Negative number not allowed here. Taking positive part.
```

```
B -100 0 - 150 0; (negative size, cant have space bet. - and number);
Error on Line # 150 at char pos 15
Error -- No number found. (using 0)
```

```
B -100 0 - 150 0; (negative size, cant have space bet. - and number);
Error on Line # 150 at char pos 21
Error -- No number found. (using 0)
```

```
B 100 300 - -500; (do you see a -0 there?); (do you think you should?);
Error on Line # 151 at char pos 16
Error -- No number found. (using 0)
```

```

      B -100 -200 10000000000000000000,-20000000000000000000;
Error on Line # 153 at char pos 7
Error -- Negative number not allowed here. Taking positive part.

      B -100 -200 10000000000000000000,-20000000000000000000;
Error on Line # 153 at char pos 12
Error -- Negative number not allowed here. Taking positive part.

      B -100 -200 10000000000000000000,-20000000000000000000;
Error on Line # 153 at char pos 28
Error -- Number too big for machine's wordsize. Max Allowable Substituted.

      B -100 -200 10000000000000000000,-20000000000000000000;
Error on Line # 153 at char pos 51
Error -- Number too big for machine's wordsize. Max Allowable Substituted.

      B 100 500 100 200 300; (half a direction vector);
Error on Line # 154 at char pos 26
Error -- No number found. (using 0)

      B 100 500 100 ; (not enough numbers to make this worth our while);
Error on Line # 155 at char pos 19
Error -- No number found. (using 0)

      B 100 100 0 0 1 1 100 200 300; (extra numbers);
Error on Line # 156 at char pos 23
Warning -- Semicolon missing. Who cares?

      B 100 100 0 0 1 1 100 200 300; (extra numbers);
Error on Line # 156 at char pos 34
Warning -- Unrecognized user command. Command Ignored

      B 100 100 0 0 1 1 M; (an extra letter);
Error on Line # 157 at char pos 23
Warning -- Semicolon missing. Who cares?

      B 100 100 0 0 1 1 M; (an extra letter);
Error on Line # 157 at char pos 24
Error -- Unrecognized Command. Skipping to semicolon.

      W 100 ; (points? who needs points?);
Error on Line # 167 at char pos 11
Error -- Zero-point path is illegal. Path remains zero-length.

      W 100 400 400 400 400; (dup. points reduces to 1-point wire);
Error on Line # 170 at char pos 26
Warning -- Duplicate points found. Extra one deleted.

      W 100 400 400 400 400 400 400 400 400;
Error on Line # 171 at char pos 42
Warning -- Duplicate points found. Extra one deleted.

      W 100 400 400 400 400 400 400 400 400;
Error on Line # 171 at char pos 42
Warning -- Duplicate points found. Extra one deleted.

      W 100 400 400 400 400 400 400 400 400;
Error on Line # 171 at char pos 42
Warning -- Duplicate points found. Extra one deleted.

```

200,800 500,700 800,800 800,800;  
 Error on Line # 175 at char pos 40  
 Warning -- Duplicate points found. Extra one deleted.

200,800 500,700 800,800 800,800;  
 Error on Line # 175 at char pos 40  
 Warning -- Duplicate points found. Extra one deleted.

200,800 500,700 800,800 800,800;  
 Error on Line # 175 at char pos 40  
 Warning -- Duplicate points found. Extra one deleted.

Wire -300 100 100 200 200; (neg. width);  
 Error on Line # 178 at char pos 10  
 Error -- Negative number not allowed here. Taking positive part.

Wire 300 100 100 200 200 300 ; (unpaired point!);  
 Error on Line # 180 at char pos 35  
 Error -- No number found. (using 0)

P; (trivial case polygon, (nullgon));  
 Error on Line # 190 at char pos 6  
 Error -- Zero-point path is illegal. Path remains zero-length.

P 100 100 100 100 ; (doubled points);  
 Error on Line # 194 at char pos 23  
 Warning -- Duplicate points found. Extra one deleted.

P 100 100 100 100 ; (doubled points);  
 Error on Line # 194 at char pos 23  
 Warning -- Duplicate points found. Extra one deleted.

P 100 100 100 100 100 100 100 100 100 100 ; (quintupled points);  
 Error on Line # 195 at char pos 47  
 Warning -- Duplicate points found. Extra one deleted.

P 100 100 100 100 100 100 100 100 100 100 ; (quintupled points);  
 Error on Line # 195 at char pos 47  
 Warning -- Duplicate points found. Extra one deleted.

P 100 100 100 100 100 100 100 100 100 100 ; (quintupled points);  
 Error on Line # 195 at char pos 47  
 Warning -- Duplicate points found. Extra one deleted.

P 100 100 100 100 100 100 100 100 100 100 ; (quintupled points);  
 Error on Line # 195 at char pos 47  
 Warning -- Duplicate points found. Extra one deleted.

P 100 100 100 100 100 100 100 100 100 100 ; (quintupled points);  
 Error on Line # 195 at char pos 47  
 Warning -- Duplicate points found. Extra one deleted.

P 100 100 400 400 400 100 100 100; (subtler -- first and last points  
 same);  
 Error on Line # 196 at char pos 38  
 Warning -- Duplicate points found. Extra one deleted.

-600,00 -600,200 -500,200 -600,00 00,00;  
 Error on Line # 202 at char pos 46

Warning -- Duplicate points found. Extra one deleted.

P 100 100 400 400 100 400 500; (missing a number);  
 Error on Line # 204 at char pos 34  
 Error -- No number found. (using 0)

R -1500 750 750; (negative diameter);  
 Error on Line # 214 at char pos 7  
 Error -- Negative number not allowed here. Taking positive part.

R 100 1500; (not enough numbers);  
 Error on Line # 216 at char pos 15  
 Error -- No number found. (using 0)

Round flash 50 (hi! expecting me?) 200 100 ; (you shouldnt be!!);  
 Error on Line # 217 at char pos 20  
 Error -- No number found. (using 0)

Round flash 50 (hi! expecting me?) 200 100 ; (you shouldnt be!!);  
 Error on Line # 217 at char pos 20  
 Error -- No number found. (using 0)

Round flash 50 (hi! expecting me?) 200 100 ; (you shouldnt be!!);  
 Error on Line # 217 at char pos 20  
 Warning -- Semicolon missing. Who cares?

Round flash 50 (hi! expecting me?) 200 100 ; (you shouldnt be!!);  
 Error on Line # 217 at char pos 40  
 Warning -- Semicolon missing. Who cares?

Round flash 50 (hi! expecting me?) 200 100 ; (you shouldnt be!!);  
 Error on Line # 217 at char pos 48  
 Warning -- Unrecognized user command. Command Ignored

E (open sym at end);  
 Error on Line # 227 at char pos 1  
 Error -- End inside symbol def. Symbol terminated.

E (open sym at end);  
 Error on Line # 227 at char pos 1  
 Warning -- Semicolon missing. Who cares?

227 lines.  
 62 errors detected. 53 warnings.