100 Elwood Davis Road ♦ North Syracuse, NY 13212 ♦ USA

Sonnet Project Format
Release 12

Specialists in High-Frequency Electromagnetic Software

(315) 453-3096   Fax: (315) 451-1694 http://www.sonnetsoftware.com

Rev 12.56.1

Table of Contents

Sonnet Project Format: Release 12

Introduction

This document details the Sonnet project file format. A detailed example of both a geometry project and a netlist project is shown followed by a detailed syntax for all possible entries in the file.

A Sonnet project file specifies a circuit geometry in the case of a geometry project and a circuit netlist in the case of a netlist project. This project file contains the specification of the circuit geometry or netlist, the analysis controls, and the analysis output data. What types of analysis data are contained in the project file depends on the types of analyses run on the project.

The structure of the project file consists of the main ".son" file which appears in the highest level directory. Residing in this directory is a folder "sondata" which retains all the response data for all the ".son" files in the parent directory. All of this data is now stored as part of the project. In the directory "sondata" is a directory for each ".son" file, with the basename of the project.

For example, you have a working directory `c:/myfilters`. You use the project editor to create three projects in this directory: steps, filter1 and filter2. These projects would produce the directory structure pictured below:

```
c:\myfilters
     |
     |------- steps.son
     |------- filter1.son
     |------- filter2.son
     |------- \sondata (Directory)
                  |
                  |------- \steps (Directory containing Sonnet data files)
                  |------- \filter1 (Directory containing Sonnet data files)
                  |------- \filter2 (Directory containing Sonnet data files)
```

A Sonnet geometry project specifies the circuit geometry to be analyzed by the electromagnetic analysis engine, ***em***. The geometry of the metalization is represented in terms of polygons. Any part of a polygon outside the box is ignored by ***em***. The coordinates of the polygon vertices are called out in terms of actual dimensions using floating point values. The polygons are automatically subsectioned by ***em*** for analysis. Polygons with dimensions smaller than the selected resolution can provide unreliable results.

A Sonnet netlist project specifies a circuit netlist composed of circuit elements defined in the project editor. The netlist is represented in terms of modeled elements, response data file elements, subproject elements and networks.

Both types of projects can be created using Sonnet's project editor or by direct editing in a text editor. The project must have a file name ending with ".son." Other information, including the box dimensions and substrate parameters, are also specified. The experienced user may wish to make minor modifications to a specific geometry or netlist by editing it directly. With this in mind, the file format has been set up to be as forgiving as possible; however, keep in mind that it is still possible to corrupt the file in a manner which will not be detected by the analysis software. For this reason, only experienced users should attempt any modification.

New Keywords and Changes to Existing Syntax

New

Variables: There is a new feature which allows you to define variables which can be assigned to dimension parameters and various material properties. You may also define a variable in terms of an equation. For details, see "VALVAR" on page 66.

Scaling for dimension parameters: There is a new field in the GEOVAR statement which defines how dimension parameters are scaled when their nominal value is changed. See "GEOVAR" on page 68 for details.

FOLDER: There is a keyword in the FILEOUT block which allows you to specify the folder in which your output files are created. See "FOLDER" on page 99.

Roger's Surface Roughness: There is a new beta feature which allows you to model the effects of surface roughness on Rogers copper foil. This is done by using the new Rogers metal type.See "MET" on page 62 for details.

Anisotropic Metal: There is a new beta feature which allows you to model anisotropic metal in which the current flows only in one direction, either entirely in the X direction or entirely in the Y direction. This is done by applying either the AnisotropicX fill or AnistropicY fill to a metal polygon. See "NUM" on page 78 for details; specifically the <filltype> field.

Changes

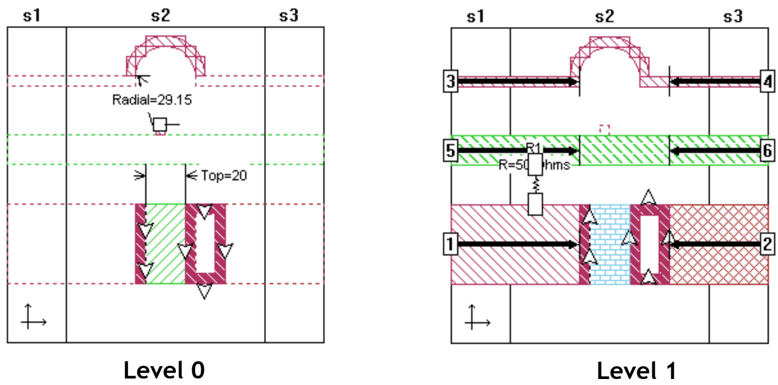
Sweep Mode: There is a new sweep mode field added to Parameter Sweep frequency controls which has changed the syntax of the VARSWP command. See "VARSWP" on page 84.

Granularity: A new granularity field has been added when defining a variable value range for an optimization. See "VARS" on page 89.

Anisotropic Dielectrics: You may now define anisotropic dielectrics for both dielectric layers and brick materials. For dielectric layers, see "BOX" on page 72. For brick materials, see "BRA" on page 66.

Geometry Project Example

The file format.son is pictured below as it appears in the project editor, ***xgeom***, followed by the file listing as it would appear in a text editor. Each line of the file is annotated as to its meaning. Note that the geometry is not a realistic circuit but designed to demonstrate various elements available in the project file.



The line numbering has been added to the text file to aid this discussion.

In general, any input on a line after “!” is ignored unless the “!” is immediately followed by a “<“. “&” is used as a continuation character. The next line is treated as a continuation of the previous line.

Line 1

`FTYP SONPROJ 4! Sonnet Project File`

This statement is mandatory at the beginning of a project. The “4” is required and is reserved for future use by Sonnet.

If the project file contains a netlist, then this entry would be “`FTYP SONNETPROJ 4`”.

Line 2

VER 12.56

This indicates the version of the project editor, which in this case is 12.56. Not required.

Line 3

`HEADER`

This command begins the HEADER block of the project file. This block contains licensing and date creation information on the project file.

Line 4

`LIC test.101`

License ID for the license used to create the project file. Not required.

Line 5

`DAT 04/22/2009 11:22:43`

The date the file was last saved. Format is MM/DD/YYYY.

Line 6

`BUILT_BY_CREATED xgeom 12.56 04/22/2009 10:37:38`

Provides information on the origin of the project. This project was created using the version 12.56 Sonnet project editor, ***xgeom***, at the date and time listed. If another program has created the original project, its name would replace “xgeom.”

Line 7

`BUILT_BY_SAVED xgeom 12.56`

The program and version which performed the most recent save on the project file; in this example, it was the version 12.56 Sonnet project editor, ***xgeom***. If another program has performed the most recent save, its name would replace “xgeom.”

Line 8

`MDATE 04/22/2009 11:22:43`

Date that file was last saved with “Medium Importance” changes.

Line 9 `HDATE 04/22/2009 11:22:43`
Date that file was last saved with “High Importance” changes. This is a change which necessitated cleaning out the project data.

Line 10 `END HEADER`
This indicates the end of the HEADER block. Required.

Line 11 `DIM`
This indicates the beginning of the dimensions block. This provides the units used for various circuit elements.

Line 12 `FREQ GHZ`
The frequencies are in gigaHertz.

Line 13 `IND NH`
The inductor values are in nanoHenries.

Line 14 `LNG mils`
Lengths in the geometry are in mils.

Line 15 `ANG DEG`
Angles are in degrees.

Line 16 `CON /OH`
The conductivity values are in siemens/meter.

Line 17 `CAP PF`
The capacitance values are in picofarads.

Line 18 `RES OH`
The resistance values are in Ohms.

Line 19 `END DIM`
This indicates the end of the dimensions block. Required if there is a DIM statement in the file.

Line 20 `FREQ`

This indicates the beginning of the frequency sweep block. This section details all the frequency sweeps specified for the project. If there is more than one type of sweep present in this block, you must refer to the Control block to see which type of sweep is presently being used in the project file. For a detailed syntax description of this block and any possible entries, see "FREQ," page 81.

Line 21 `SIMPLE 5.0 10.0 1.0`

This line specifies a Linear Frequency sweep starting at 5.0 GHz, ending at 10.0 GHz with a step value of 1.0 GHz.

Line 22 `ABS 5.0 10.0`

This line specifies an analysis band synthesis (ABS) for the frequency band 5.0 GHz to 10.0 GHz.

Line 23 `SWEEP 5.0 10.0 1.0`

This line specifies a linear sweep entered in the frequency combinations. The sweep starts at 5.0 GHz, ends at 10.0 GHz with a step of 1.0 GHz.

Line 24 `ABS_ENTRY 5.0 10.0`

This line specifies an ABS sweep entered in the frequency combinations. The analysis band is 5.0 GHz to 10.0 GHz.

Line 25 `STEP 45.0`

This line specifies a single frequency entered in the frequency combinations. The circuit is analyzed at 45.0 GHz.

Line 26 `ESWEEP 5.0 10.0 5`

This line specifies an exponential sweep from 5.0 GHz to 10.0 GHz using five analysis frequencies.

Line 27 `ABS_FMIN NET= S13 5.0 10.0`

This line specifies a Find Minimum sweep entered in the Frequency Combinations. NET= is used for all ABS_ FMIN statements. The response whose minimum is to be found is S13. The circuit will be analyzed using ABS over a frequency band of 5.0 GHz to 10.0 GHz.

Line 28 `ABS_FMAX NET= S21 5.0 10.0`
This line specifies a Find Maximum sweep entered in the Frequency Combinations. NET= is used for all FINDMIN statements. The response whose minimum is to be found is S13. The circuit will be analyzed using ABS over a frequency band of 5.0 GHz to 10.0 GHz.

Line 29 `END FREQ`
This indicates the end of the frequency sweep block. Required if there is a FREQ statement in the file.

Line 30 `CONTROL`
This indicates the beginning of the Control block which is a mandatory block. This section details the frequency sweep presently being used as well as run options and Speed/Accuracy control. All possible entries are not present in this example file; for a detailed syntax description of this block and any possible entries, see "CONTROL," page 83.

Line 31 `EXTFILE`
This indicates that the presently defined sweep uses an external frequency file. The file is identified below using the FILENAME keyword. This means that the analysis control statements in the external Frequency file are used and the statements in the FREQ block are ignored.

Line 32 `OPTIONS  -bd`
This entry details the run options selected in the analysis setup. The "b" indicates the Box Resonance Info option and the "d" indicates the de-embedding option.

Line 33 `FILENAME sondata\format\format.eff`
This entry identifies the pathname of the external frequency file being used to control the analysis frequencies.

Line 34 `SPEED 0`
This entry indicates the position of the Speed/Memory Slider and is only valid for Geometry projects.The Speed/Memory Slider is accessed by clicking on the Speed/Memory button in the Analysis Setup dialog box in the project editor. 0 is the Fine/Edge Meshing (far left) position of the slider. This is the default value for SPEED.

Line 35 `RES_ABS N 200.0`

Line 36 `CACHE_ABS 1`

This indicates the ABS caching level set in the Advanced Options dialog box. A value of 1 indicates that the ABS caching level is Stop/Restart which is the default value. For a detailed syntax description, see "CACHE_ABS," page 87

Line 37 `TARG_ABS 300`

This indicates that the target for the number of frequencies for the ABS sweep is 300. This statement is for the Automatic setting for ABS Frequency Resolution found in the Advanced Options dialog box. Since the RES_ABS statement does not appear in this project, Automatic, which is the default, is selected. For a detailed syntax descriptions of both entries, see "TARG_ABS," page 88 and "RES_ABS," page 87.

Line 38 `Q_ACC N`

This statement determines if the Q-Factor Accuracy option in the Advanced Options dialog box is being used. "N" indicates that the value is not being used.

Line 39 `END CONTROL`

This indicates the end of the Control block. Required.

Line 40 `GEO`

This indicates the beginning of the Geometry block which is present only for a geometry project. This section details the geometry entered in the project editor. The GEO block must appear in the file before any blocks which reference GEOVARS which is part of the GEO block. All possible entries are not present in this example file; for a detailed syntax description of this block and any possible entries, see "GEO," page 59.

Line 41 `DRP1 LEFT FIX 65`

The DRP1 entry defines a reference plane extending from the left box wall for a fixed length of 65 mils. If there are no reference planes defined in a circuit, then there are no DRP1 entries.

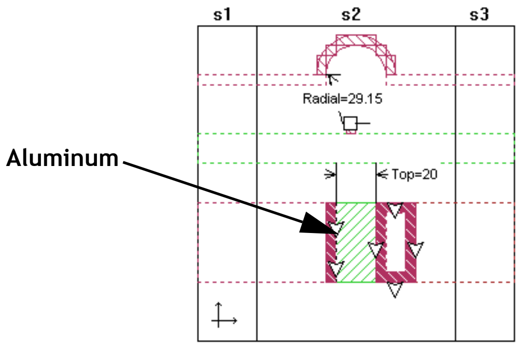
Line 42
`DRP1 RIGHT FIX 50`
The DRP1 entry defines a reference plane extending from the right box wall for a fixed length of 50 mils. If there are no reference planes defined in a circuit, then there are no DRP1 entries.

Line 43
`TMET "Free Space" 0 FREESPACE 376.7303136 0 0 0`
Definition of the Box Top Metal. The name of the metal is “Free Space” which appears in quotes since there is a space in the name. This is one of the default metals available in the project editor. The index number for the fill-pattern is zero. The metal type is FREESPACE. DC resistance is 376.7303136 ohms/sq, RF resistance is 0, DC reactance is 0, Kinetic inductance is 0. This is the definition used for free space, i.e., no box cover. For a detailed syntax description, see “TMET,” page 60.

Line 44
`BMET "Lossless" 0 SUP 0 0 0 0`
Definition of the Box Bottom Metal. The name of the metal is “Lossless.” This is one of the default metals available in the project editor. The index number for the fill-pattern is zero. The metal type is General indicated by the keyword SUP. This metal type requires the following four values to define the loss: DC resistance is 0 ohms/sq, RF resistance is 0, DC reactance is 0, Kinetic inductance is 0. This is the definition used for lossless metal. For a detailed syntax description, see “BMET,” page 62.

Line 45
`MET “Aluminum” 1 NAT 0.0011 3.3e-007`
This entry defines a metal. The metal name is Aluminum. The index number for the fill-pattern is 1. The metal type is Native indicated by the NAT keyword. This metal type requires two values; the DC resistance is 0.0011 ohms/sq and the RF resistance is 3.3e-007. For a detailed syntax description, see “MET,” page 62.

This metal type was user-defined in the Metal Types dialog box. Its use in the example file is shown below.



Line 46

`MET "Brass" 2 NAT 0.0025 5e-007`

This entry defines a metal. The metal name is Brass. The index number for the fill-pattern is 2. The metal type is Native indicated by the NAT keyword. This metal type requires two values; the DC resistance is 0.0025 ohms/sq and the RF resistance is 5e-007. For a detailed syntax description, see "MET," page 62.

Line 47

`MET "Thick Silver" 52 TMM 61700000 0 3 2`

This entry defines a metal. The metal name is Thick Silver. The name is in quotes since it contains a space. The index number for the fill-pattern is 52. The metal type uses the Thick Metal Model indicated by the TMM keyword. This metal type requires four values; the Conductivity is 61700000 S/m. The current ratio is not used and is set to zero. The next field is the thickness of the metal which in this case is set to 3 mils. For this definition the thickness represents a physical thickness. The next field is 2 indicating that the metal is modeled using two sheets. For a detailed syntax description, see "MET," page 62.

Line 48

`BOX 2 160 160 64 64 20 0`

The BOX entry defines the substrate size (the bottom of the six-sided box enclosure) and the dielectric layers used in the circuit. There are 2 metal levels (in addition to ground which is the box bottom). The substrate is 160 mils by a 160 mils with 32 cells in the X dimension and 32 cells in the Y dimension. The number of cells is given as 2N where N is the number of cells. The next field is no longer used and the estimated relative epsilon effective is 0. For a detailed syntax description, see “BOX,” page 72.

The next three entries in the geometry file define the three dielectric layers in the circuit. These three lines are part of the BOX section.

Line 49

```
100 1 4 2 5 3 1 "Air"
```

The top dielectric is defined as 100 mils thick with the following parameters: Erel, the relative dielectric constant, is 1.0, the relative magnetic permeability is 4, the dielectric loss tangent is 2, the magnetic loss tangent is 5, the dielectric conductivity is 3, and the Z-Partitioning is 1. These values define the dielectric “Air” between the top metal level and the box top. Note that these are not the actual parameters for air but were used for purposes of the example.

Line 50

```
25 12.9 1 0.006 0 0 1 "Gallium Arsenide"
```

This next line defines the next dielectric layer. It is 25 mils thick with a relative dielectric constant of 12.9. All other values are the same as those cited above for air. The name of the dielectric is Gallium Arsenide. Quotation marks must be used when there is a space in the name.

Line 51

```
25 2.94 1 0.0025 0 0 1 "Arlon CLTE" A 3 1 0.003 0 0
```

This next line defines the third dielectric layer. This is an anisotropic dielectric indicating the dielectric properties in the z direction are different from those in the x and y direction. The first set of constants are the properties in the x and y direction. It is 25 mils thick with a relative dielectric constant of 2.94. All other values are the same as those cited above for air. The name of the dielectric is Arlon CLTE. Quotation marks must be used when there is a space in the name. The “A” following the dielectric name indicates that this is an anisotropic dielectric and is followed by the properties of the dielectric in the z direction. The relative dielectric constant is 3, the relative magnetic permeability is 1, the dielectric loss tangent is 0.003, the magnetic loss tangent is 0, and the dielectric conductivity is 3.

Line 52

```
VALVAR Top LNG 20 "Dim. Param. Top"
```

The VALVAR statement defines a variable in a project. There is a VALVAR statement for each variable that is defined. This statement defines a variable with the name “Top”. “Top” is a variable whose value is 20 mils. The units are “LNG” which indicates the present length units of the project which in this case are mils. "Dim. Param. Top" is the description of the variable entered when it was defined. This description is displayed in the variable list.

Line 53 `GEOVAR Top SYM XDIR 1 SCUNI`

The GEOVAR statement defines a dimension parameter in the geometry circuit. There is a GEOVAR statement for each parameter that is defined. This statement defines a dimension parameter with the variable "Top" assigned to it. This is a symmetrical parameter in the x direction (horizontal plane) and is scaled in the x-direction. The GEOVAR statement is followed by statements which define the label position, anchor position, reference points and point sets.

Line 54 `POS 4.999998091 -14.46808381`

The POS entry defines the location of the parameter label in the circuit in relation to the first reference position.

Line 55 `NOM 20`

This entry is the present nominal value of the parameter.

Line 56 `REF1 POLY 13 1`

This entry defines the first reference point for the parameter. The reference point is attached to polygon 13 and there is one point. The next statement identifies which point of the polygon is used which in this case is point 0 on polygon 13. The number of the polygon is the file ID, not the position of the polygon in the project file.

Line 57 `0`

As stated above, this identifies point 0 as the point of the polygon which is used as the first reference point.

Line 58 `REF2 POLY 13 1`

This entry defines the second reference point for the parameter. The reference point is attached to polygon 13 and there is one point. The next statement identifies which point of the polygon is used which in this case is point 1 on polygon 13. The number of the polygon is the file ID, not the position of the polygon in the project file.

Line 59

`1`

As stated above, this identifies point 1 as the point of the polygon which is used as the first reference point.

Line 60

`PS1 3`

This entry begins the specification of the first point set for reference 1. There are three POLY statements in the point set. The next five lines continue the specification of the first point set.

Line 61

`POLY 13 1`

The first point in the first point set is one point in polygon 13. Which point on the polygon is specified by the next line.

Line 62

`3`

The first point in the first point set on polygon 13 is point 3. If additional points on this polygon were contained in the point set, each would be entered on its own line.

Line 63

`POLY 15 2`

The next two points in the first point set are in polygon 15. Which two points on the polygon are specified by the next two lines.

Line 64

`1`

The second point in the first point set on polygon 45 is point 1.

Line 65

`2`

The third point in the first point set on polygon 45 is point 2.

Line 66

`POLY 20 2`

The next two points in the first point set are in polygon 20. Which two points on the polygon are specified by the next two lines.

Line 67

`0`

The fourth point in the first point set on polygon 18 is point 0.

Line 68

`3`

The fifth point in the first point set on polygon 18 is point 3.

Line 69 `END`
This END statement indicates the end of the first point set, started with the PS1 statement. This is required each time a PS1 statement is used.

Line 70 `PS2 4`
This entry begins the specification of the second point set for reference 2. There four polygons which contain points in the point set. The next 13 lines continue the specification of the second point set.

Line 71 `POLY 13 1`
The first point in the second point set is one point in polygon 13. Which point on the polygon is specified by the next line.

Line 72 `2`
The first point in the second point set on polygon 13 is point 2. If there were additional points in the polygon which were contained in the point set, each would be entered on its own line.

Line 73 `POLY 21 4`
The next four points in the second point set are on polygon 21. Which points on the polygon are specified in the next four lines.

Line 74 `0`
The second point in the second point set is point 0 on polygon 21.

Line 75 `1`
The third point in the second point set is point 1 on polygon 21.

Line 76 `2`
The fourth point in the second point set is point 2 on polygon 21.

Line 77 `3`
The fifth point in the second point set is point 3 on polygon 21.

Line 78 `POLY 18 2`
The next two points in the second point set are on polygon 18. Which points on the polygon are specified in the next two lines.

Line 79 `0`
The sixth point in the second point set is point 0 on polygon 18.

Line 80 `3`
The seventh point in the second point set is point 3 on polygon 18.

Line 81 `POLY 20 2`
The next two points in the second point set are on polygon 20. Which points on the polygon are specified in the next two lines.

Line 82 `1`
The seventh point in the second point set is point 1 on polygon 18.

Line 83 `2`
The eighth point in the second point set is point 2 on polygon 18.

Line 84 `END`
This END statement indicates the end of the second point set, started with the PS2 statement. This is required each time a PS2 statement is used.

Line 85 `END`
This END statement indicates the end of the GEOVAR statement. This is required whenever a GEOVAR statement occurs.

Line 86 `VALVAR Radial LNG 29.15 ""`
The VALVAR statement defines a variable in a project. There is a VALVAR statement for each variable that is defined. This statement defines a variable with the name “Radial”. “Radial” is a variable whose value is 29.15 mils. The units are “LNG” which indicates the present length units of the project which in this case are mils. “” indicates that no description was entered when the variable was defined.

Line 87 `GEOVAR Radial RAD YDIR 1 NSCD`
The GEOVAR statement defines a dimension parameter in the geometry circuit. There is a GEOVAR statement for each parameter that is defined. This statement defines a dimension parameter with the variable “Radial” assigned to it. This is a

radial parameter and is not scaled (you may not apply scaling to radial dimension parameters). The GEOVAR statement is followed by statements which define the label position, anchor position, reference points and point sets.

Line 88

`POS -6.392375479 -12.49999864`

The POS entry defines the location of the parameter label in the circuit in relation to the first reference position.

Line 89

`NOM 29.15`

This entry is the present nominal value of the parameter.

Line 90

`REF1 POLY 47 1`

This entry defines the first reference point for the parameter. The reference point is attached to polygon 47 and there is one point. The next statement identifies which point of the polygon is used which in this case is point 1 on polygon 47. The number of the polygon is the file ID, not the position of the polygon in the project file.

Line 91

`1`

As stated above, this identifies point 1 as the point of the polygon which is used as the first reference point.

Line 92

`REF2 POLY 46 1`

This entry defines the second reference point for the parameter. The reference point is attached to polygon 46 and there is one point. The next statement identifies which point of the polygon is used which in this case is point 39 on polygon 46. The number of the polygon is the file ID, not the position of the polygon in the project file.

Line 93

`39`

As stated above, this identifies point 39 as the point of the polygon which is used as the first reference point.

Line 94

`PS1 0`

This entry begins the specification of the first point set for reference 1. The first point set is not used for a radial dimension parameter, so the number of polygons which contain points in the point set is always set to zero.

Line 95 `END`
This END statement indicates the end of the first point set, started with the PS1 statement. This is required each time a PS1 statement is used.

Line 96 `PS2 1`
This entry begins the specification of the second point set of the radial dimension parameter. There is one polygon which contains points in the point set. The next 39 lines continue the specification of the second point set.

Line 97 `POLY 46 39`
The first point in the second point set is in polygon 46 and there are a total of 39 points in polygon 46 included in the point set. The points in the polygon are specified in the next set of entries.

Line 98 `0`
The first point in the second point set is point 0 in polygon 46.

Line 99 `1`
The second point in the second point set is point 1 on polygon 46. The next 37 entries specify the point numbers on polygon 46 included in the second point set.

Line 100	2
Line 101	3
Line 102	4
Line 103	5
Line 104	6
Line 105	7
Line 106	8
Line 107	9
Line 108	10
Line 109	11
Line 110	12
Line 111	13
Line 112	14
Line 113	15
Line 114	16
Line 115	17
Line 116	18
Line 117	19
Line 118	20

Line 119	21
Line 120	22
Line 121	23
Line 122	24
Line 123	25
Line 124	26
Line 125	27
Line 126	28
Line 127	29
Line 128	30
Line 129	31
Line 130	32
Line 131	33
Line 132	34
Line 133	35
Line 134	36
Line 135	37
Line 136	38 This is the last point in the second point set; point 39 in polygon 46.

Line 137 `END`

This END statement indicates the end of the second point set, started with the PS2 statement. This is required each time a PS2 statement is used.

Line 138 `END`

This END statement indicates the end of the GEOVAR statement. This is required whenever a GEOVAR statement occurs.

Line 139 `ENDEVIA1`

The EVIA statement block defines the locations of edge vias in the circuit. Vias are defined as extending from a vertex of a polygon to the next vertex. Only the originating vertex is specified in the POLY statement. The index for a polygon and its vertices are defined in the NUM statement which always appears last in a geometry file. The TOLEVEL statement defines the level to which the edge via extends. The level of origin is identified by the polygon to which the edge via is attached. For a detailed syntax description of EVIA1, see “EVIA1,” page 73. For a detailed syntax describing the NUM statement, see “NUM,” page 78.

Line 140 `POLY 15 1`

This line specifies that the edge via is attached to polygon 15. The “1” is required at the end of this line.

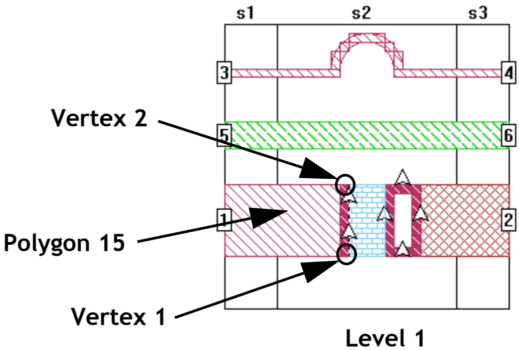
Line 141 `1`

This line identifies the edge of polygon 15 to which the edge via is attached. The edge via extends from point 1 to point 2 on polygon 15.

Line 142 `TOLEVEL 0`

This line specifies the level to which the edge via extends. The via originates on the level on which polygon 15 appears and extends to Level 0 of the circuit.

This edge via is defined as being on polygon 15 on level 1, extending from vertex 1. Pictured below is how this statement relates to the circuit as depicted in ***xgeom***.



Vertex 1 corresponds to the point (70,130) and extends to vertex 2 which is (70,90).

Note that for the file format the origin is the upper left hand corner. When the geometry is displayed in the project editor, the origin is the lower left hand corner.

Line 143 `POR1 STD`

Line 144 `POLY 15 1`

Line 145 `3`

Line 146 `1 50 0 0 0 0 110.0000027`

The next four lines of the project file define a port in the circuit. Definition of a port starts with the POR1 statement which defines a single port of the circuit. Each port in the circuit will have its own POR1 command. This is a standard box-wall port indicated by STD. The next two lines define the position of the port in relationship to the polygon on which it is placed. This port is on the polygon whose file ID is 15. Similar to a via, the port position is defined as extending from a vertex of a polygon to the next vertex, therefore the number of points on the polygon always defaults to 1 for a port as is shown in the end of line 144. Only the originating vertex is specified in the command. This port is between the 3rd and 4th vertices of the polygon. The last line defines the port number, port attributes and port location on the substrate. This is port #1 whose resistance is 50 ohms, the reactance, inductance and capacitance are all 0. The port is located at (0,110.0000027) on the substrate. For a detailed syntax description of POR1, see “POR1,” page 73.

Line 147 `POR1 STD`

Line 148 `POLY 18 1`

Line 149 `1`

Line 150 `2 50 0 0 0 160 110.0000027`

The next four lines, 147 - 150, define another port in the circuit. This is also a standard box-wall port on the polygon with the file ID 18. The port occurs between the 1st and 2nd vertex of the polygon. The port number is 2. The resistance is 50 ohms, and the reactance, inductance and capacitance are all 0. The port location on the substrate used for ***emvu*** is (160, 110.0000027).

Line 151 `POR1 STD`

Line 152 `POLY 23 1`

Line 153 `3`

Line 154 `3 50 0 0 0 0 27.4999991`

The next four lines, 151 - 154, define another port in the circuit. This is also a standard box-wall port on the polygon with the file ID 23. The port occurs between the 3rd and 4th vertex of the polygon. The port number is 3. The resistance is 50 ohms, and the reactance, inductance and capacitance are all 0. The port location on the substrate used for ***emvu*** is (0, 27.4999991).

Line 155 `POR1 STD`

Line 156 `POLY 25 1`

Line 157 `1`

Line 158 `4 50 0 0 0 160 27.4999991`

The next four lines, 155 - 158, define another port in the circuit. This is also a standard box-wall port on the polygon with the file ID 25. The port occurs between the 1st and 2nd vertex of the polygon. The port number is 4. The resistance is 50 ohms, and the reactance, inductance and capacitance are all 0. The port location on the substrate used for ***emvu*** is (160, 27.4999991).

Line 159 `POR1 STD`

Line 160 `POLY 27 1`

Line 161 `3`

Line 162 `5 50 0 0 0 0 62.5`

The next four lines, 159 - 162, define another port in the circuit. This is also a standard box-wall port on the polygon with the file ID 27. The port occurs between the 3rd and 4th vertex of the polygon. The port number is 5. The resistance is 50 ohms, and the reactance, inductance and capacitance are all 0. The port location on the substrate used for ***emvu*** is (0, 62.5).

Line 163 `POR1 STD`

Line 164 `POLY 27 1`

Line 165 `1`

Line 166 `6 50 0 0 0 160 62.5`

The next four lines, 163 - 166, define another port in the circuit. This is also a standard box-wall port on the polygon with the file ID 27. The port occurs between the 1st and 2nd vertex of the polygon. The port number is 5. The resistance is 50 ohms, and the reactance, inductance and capacitance are all 0. The port location on the substrate used for ***emvu*** is (160, 62.5).

Line 167 `NUM 11`

The NUM command defines the number of polygons in a circuit, which in this case is eleven. There are at least six lines per polygon following the NUM command line. These lines, which are detailed below, define the type and location of each polygon. For a detailed syntax description of NUM, see "NUM," page 78.

The next line identifies the polygon as metal ("MET POL"), dielectric brick ("BRI POL") or a via polygon ("VIA POLYGON"). In the case of a metal polygon, this entry is optional and as can be seen for the polygon below is usually omitted.

Line 168 `0 5 0 N 13 1 1 100 100 0 0 0 N`

The polygon appears on level 0, has 5 vertices, uses metal type 0 (any metal type ≥ 0 indicates a user defined metal; in this case, aluminum), uses normal (staircase) fill. Its Debug number is 13. This is the number used to identify the polygon when specifying a parameter. The minimum x and y dimension subsections are 1 cell and the maximum x and y dimension subsections are 100. The next three values are not presently used but are reserved for future development. The N indicates that edge meshing is not on for this polygon. If edge meshing were turned on, this field would be a Y. For a detailed syntax description, see “NUM,” page 78.

The next five lines of the geometry file specify the locations of the five vertices of the polygon. The number of locations and hence command lines always corresponds to the number of vertices in the polygon followed by the END statement to indicate the end of information on that particular polygon.

Line 169 `70 90`

Line 170 `90 90`

Line 171 `90 130`

Line 172 `70 130`

Line 173 `70 90`

Line 174 `END`

It is important to note that the index number assigned to a polygon is implicitly assigned by the location of its appearance in the geometry file. The index always starts with 0 and is restarted for each metal level. Similarly, the index number of a vertex is implicitly stated by the order of its appearance in the file also. For example, in this case, the polygon cited above is polygon 0 on level 0. Vertex 0 is (70, 90), vertex 1 is (90, 90), vertex 2 is (90, 130), vertex 3 is (70, 130), and vertex 4 is (70, 90) which is the same location as vertex 0. Note that the last vertex in a polygon must always be identical to the first vertex. For a detailed syntax description, see “NUM,” page 78.

Line 175 `0 41 -1 V 46 1 1 100 100 0 0 0 Y`

Line 176 This polygon appears on level 0, has 41 vertices, uses metal type -1, which is the default lossless metal, (any metal type ≥ 0 indicates a user defined metal; in this case, aluminum), and uses conformal fill. Its Debug number is 46. This is the number used to identify the polygon when specifying a dimension parameter. The minimum x and y dimension subsections are 1 cell and the maximum x and y dimension subsections are 100. The next three values are not presently used but are reserved for future development. The Y indicates that edge meshing is on for this polygon. For a detailed syntax description, see “NUM,” page 78

The next 41 lines identify the 41 vertices of the polygon. The number of locations and hence command lines always corresponds to the number of vertices in the polygon followed by the END statement to indicate the end of information on that particular polygon.

Line 177 `60.00297419 25.00371593`

Line 178 `60.00285093 23.254038`

Line 179 `60.61026261 19.8078504`

Line 180 `61.80687847 16.51952959`

Line 181 `63.55633367 13.48899116`

Line 182 `65.80546502 10.80831837`

Line 183 `68.48592643 8.558964006`

Line 184 `71.51626575 6.809275232`

Line 185 `74.80439966 5.612416811`

Line 186 `78.25041172 5.004755619`

Line 187 `81.74958828 5.004755619`

Line 188 `85.19560034 5.612416811`

Line 189 `88.48373425 6.809275232`

Line 190 `91.51407357 8.558964006`

Line 191 `94.19453498 10.80831837`

Line 192 `96.44366633 13.48899116`

Line 193 `98.19312153 16.51952959`

Line 194 `99.38973739 19.8078504`

Line 195 `99.99714907 23.254038`

Line 196	99.99702581 25.00371593
Line 197	94.99755047 25.00408074
Line 198	94.99764209 23.69180438
Line 199	94.54209214 21.10713555
Line 200	93.64464649 18.64087494
Line 201	92.33257698 16.36795709
Line 202	90.64575466 14.35744286
Line 203	88.63543805 12.67042064
Line 204	86.36271539 11.35814992
Line 205	83.89664844 10.46050367
Line 206	81.31217383 10.00475665
Line 207	78.68782617 10.00475665
Line 208	76.10335156 10.46050367
Line 209	73.63728461 11.35814992
Line 210	71.36456195 12.67042064
Line 211	69.35424534 14.35744286
Line 212	67.66742302 16.36795709
Line 213	66.35535351 18.64087494
Line 214	65.45790786 21.10713555

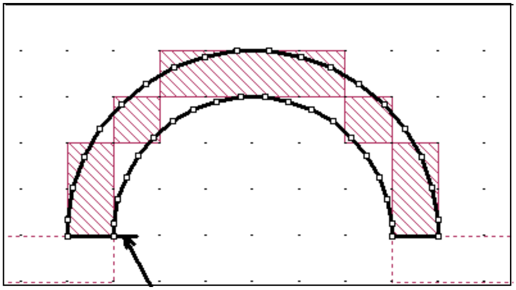
Line 215 `65.00235791 23.69180438`

Line 216 `65.00244953 25.00408074`

Line 217 `60.00297419 25.00371593`

Line 218 `END`

This concludes the definition of the polygon. This polygon with all 41 vertices highlighted in reshape mode is shown below.‘



Line 219 `1 5 -1 N 15 1 1 100 100 0 0 0 N`

Line 220 `0 90`

Line 221 `70 90`

Line 222 `70 130`

Line 223 `0 130`

Line 224 `0 90`

Line 225 `END`

This block of commands provides information on the next polygon, which is polygon 0 for level 1. This polygon has 5 vertices, uses the default lossless metal (indicated by the -1), and uses normal (staircase) fill. The minimum x and y dimension subsections are 1 cell and the maximum x and y dimension subsections are 100. The final three values are not presently used but are reserved for future development. The N at the end indicates that edge meshing is turned off. The locations of the five vertices are given in the next five statements followed by the end statement. For a detailed syntax description, see "NUM," page 78.

Line 226 `1 5 1 N 18 1 1 100 100 0 0 0 N`

Line 227 `110 90`

Line 228 `160 90`

Line 229 `160 130`

Line 230 `110 130`

Line 231 `110 90`

Line 232 `END`

This block of commands provides information on the next polygon, which is polygon 1 for level 1. This polygon has 5 vertices, uses the user defined metal type 1, brass, and uses normal (staircase) fill. The minimum x and y dimension subsections are 1 cell and the maximum x and y dimension subsections are 100. The final three values are not presently used but are reserved for future development. The N at the end indicates that edge meshing is turned off. The locations of the five vertices are given in the next five statements followed by the end statement.

Line 233 `BRI POLY`

Line 234 `1 5 0 N 20 1 1 100 100 0 0 0 N`

Line 235 `70 90`

Line 236 `90 90`

Line 237 `90 130`

Line 238 `70 130`

Line 239 `70 90`

Line 240 `END`

This block of commands provides information on the next polygon, which is polygon 2 for level 1. The first statement indicates that this polygon is a dielectric brick. This statement is REQUIRED for dielectric bricks. This polygon has 5 vertices, uses the default brick material air indicated by an index of 0 (user defined dielectrics start at index 1) and uses normal (staircase) fill. The minimum x and y dimension subsections are 1 cell and the maximum x and y dimension subsections are 100. The final three values are not presently used but are reserved for future development. The N at the end indicates that edge meshing is turned off. The locations of the five vertices are given in the next five statements followed by the end statement. For a detailed syntax descriptions, see “NUM,” page 78.

Line 241 `VIA POLYGON`

Line 242 `1 5 -1 N 21 1 1 100 100 0 0 0 Y`

Line 243 `TOLEVEL 0`

Line 244 `90 90`

Line 245 `110 90`

Line 246 `110 130`

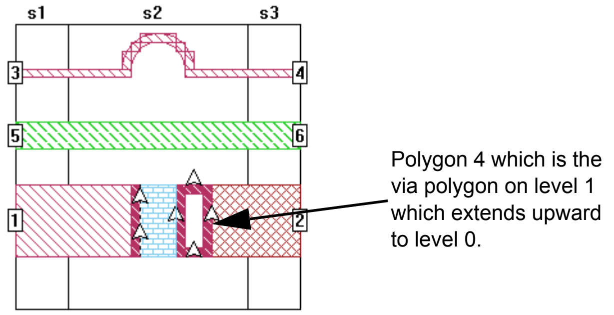
Line 247 `90 130`

Line 248 `90 90`

Line 249 `END`

This block of commands provides information on the next polygon, which is polygon 3 for level 1. The first statement indicates that this polygon is a via polygon. This statement MUST appear for via polygons. This via polygon has 5 vertices, uses the default lossless metal (indicated by the -1), and uses normal (staircase) fill. The minimum x and y dimension subsections are 1 cell and the maximum x and y dimension subsections are 100. The final three values are not presently used but are reserved for future development. The Y at the end indicates that edge meshing is turned on. The TOLEVEL statement indicates that this via extends from level 1, where the polygon originates, down to level 0. The locations of the five vertices are given in the next five statements followed by the end statement. For a detailed syntax descriptions, see “NUM,” page 78.

This via polygon is defined as being polygon 4 on level 1, extending up to level 0. Pictured below is how this statement relates to the circuit as depicted in ***xgeom***. For a detailed syntax description of EVIA, see “EVIA1,” page 73.



Level 1

Line 250 `1 41 -1 V 22 1 1 100 100 0 0 0 Y`

This block of commands provides information on the next polygon, which is polygon 4 for level 1. This is a metal polygon. This polygon has 41 vertices, uses the default lossless metal (indicated by the -1), and uses conformal fill. The minimum x and y dimension subsections are 1 cell and the maximum x and y dimension subsections are 100. The final three values are not presently used but are reserved for future development. The Y at the end indicates that edge meshing is turned on. The locations of the 41 vertices are given in the next forty-one statements followed by the end statement. For a detailed syntax descriptions, see “NUM,” page 78.

Line 251 60 24.99999819

Line 252 60 23.25022492

Line 253 60.60768988 19.80384435

Line 254 61.80460529 16.51534629

Line 255 63.55437856 13.48465009

Line 256 65.80384372 10.8038419

Line 257 68.4846519 8.554376748

Line 258 71.5153481 6.804603477

Line 259 74.80384616 5.607688068

Line 260 78.25022673 4.999998188

Line 261 81.74977327 4.999998188

Line 262 85.19615384 5.607688068

Line 263 88.4846519 6.804603477

Line 264 91.5153481 8.554376748

Line 265 94.19615628 10.8038419

Line 266 96.44562144 13.48465009

Line 267 98.19539471 16.51534629

Line 268 99.39231012 19.80384435

Line 269 100 23.25022492

Line 270 `100 24.99999819`

Line 271 `95 24.99999819`

Line 272 `95 23.68766824`

Line 273 `94.54423259 21.10288281`

Line 274 `93.64654603 18.63650927`

Line 275 `92.33421608 16.36348711`

Line 276 `90.64711721 14.35288097`

Line 277 `88.63651108 12.66578211`

Line 278 `86.36348892 11.35345215`

Line 279 `83.89711538 10.4557656`

Line 280 `81.31232995 9.999998188`

Line 281 `78.68767005 9.999998188`

Line 282 `76.10288462 10.4557656`

Line 283 `73.63651108 11.35345215`

Line 284 `71.36348892 12.66578211`

Line 285 `69.35288279 14.35288097`

Line 286 `67.66578392 16.36348711`

Line 287 `66.35345397 18.63650927`

Line 288 `65.45576741 21.10288281`

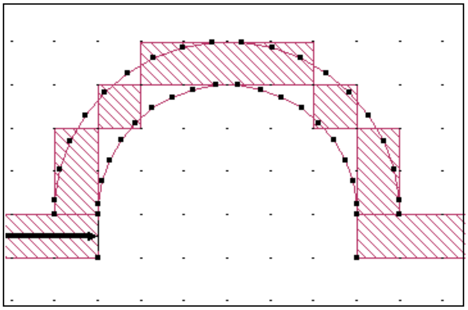
Line 289 `65 23.68766824`

Line 290 `65 24.99999819`

Line 291 `60 24.99999819`

Line 292 `END`

This concludes the definition of the polygon. This polygon with all 41 vertices highlighted in reshape mode is shown below.‘



Line 293 `1 5 -1 N 23 1 1 100 100 0 0 0 Y`

Line 294 `0 25`

Line 295 `65 25`

Line 296 `65 30`

Line 297 `0 30`

Line 298 `0 25`

Line 299 `END`

This block of commands provides information on the next polygon, which is polygon 5 for level 1. This polygon has 5 vertices, uses the default lossless metal (indicated by the -1), and uses normal (staircase) fill. The minimum x and y dimension subsections are 1 cell and the maximum x and y dimension subsections are 100. The final three values are not presently used but are reserved for future development. The N at the end indicates that edge meshing is turned off. The locations of the five vertices are given in the next five statements followed by the end statement.

Line 300 `1 5 -1 N 25 1 1 100 100 0 0 0 Y`

Line 301 `95 25`

Line 302 `160 25`

Line 303 `160 30`

Line 304 `95 30`

Line 305 `95 25`

Line 306 `END`

This block of commands provides information on the next polygon, which is polygon 6 for level 1. This polygon has 5 vertices, uses the default lossless metal (indicated by the -1), and uses normal (staircase) fill. The minimum x and y dimension subsections are 1 cell and the maximum x and y dimension subsections are 100. The final three values are not presently used but are reserved for future development. The N at the end indicates that edge meshing is turned off. The locations of the five vertices are given in the next five statements followed by the end statement.

Line 307 `1 5 2 N 27 1 1 100 100 0 0 0 Y`

Line 308 `0 55`

Line 309 `160 55`

Line 310 `160 70`

Line 311 `0 70`

Line 312 `0 55`

Line 313 `END`

This block of commands provides information on the next polygon, which is polygon 7 for level 1. This polygon has 5 vertices, uses the user defined metal type 2, thick silver, and uses normal (staircase) fill. The minimum x and y dimension subsections are 1 cell and the maximum x and y dimension subsections are 100. The final three values are not presently used but are reserved for future development. The N at the end indicates that edge meshing is turned off. The locations of the five vertices are given in the next five statements followed by the end statement.

Line 314 `END GEO`

This indicates the end of the GEO block and is required in a geometry project when the GEO block is used.

Line 315 `SUBDIV`

The SUBDIV block defines any subdividers in your project. If there are no subdividers in your geometry project this block is omitted. This block is only used in a geometry project.

Line 316 `MAIN $BASENAME_net.son`

This entry defines the name of the main netlist project which results from the subdivide command. If subdivision lines are present in a circuit, but no subdivide command has been performed this entry is "..\..\..\..\". In this case, the name is $BASENAME_net.son which for our example "format.son" is "format_net.son." If a new name has been entered as the name for the main netlist that is not a derivative of the source file, it appears here and does not use a variable.

Line 317 `REFPLANE A 80`

This entry defines the automatically added reference planes for the subprojects created by the subdivide command. These settings are entered in the Subproject Specifications dialog box. This entry uses the auto calculated reference planes whose lengths are 80 mils. For the other settings, see "REFPLANE," page 106. If a subdivide command has not been executed yet, the default value for this line is "REFPLANE A 0"

Line 318 `NAME 1 $BASENAME_net_s1.son`

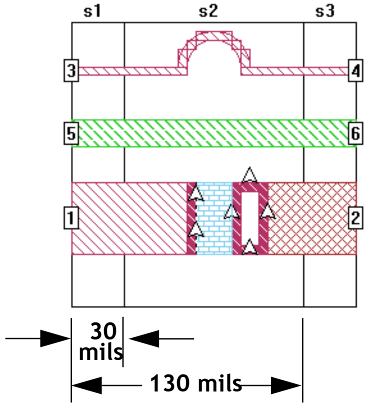
Line 319 `NAME 2 $BASENAME_net_s2.son`

Line 320 `NAME 3 $BASENAME_net_s3.son`

The next three entry lines specify the project file names to be used for the subprojects created by the subdivide command. In this case, all three files use the basename of the source file, i.e., the file names are format_net_s1.son, format_net_s2.son and format_net_s3.son. If other names had been entered in the Subproject Specifications dialog box by the user, they would appear here. These entries are omitted if no subdivide command has been executed yet. There are as many "NAME" entries as there are sections in your circuit created by the subdivision lines which also corresponds to the number of subprojects.

Line 321 `LINE 1 30 V`

This entry specifies the first subdivision line placed in the circuit. The line is 30 mils from the left box wall in the vertical direction.



Line 322 `LINE 2 130 V`

The entry specifies the second subdivision line placed in the circuit. The line is 130 mils from the left box wall in the vertical direction.

There is a “LINE” entry for each subdivision line in a geometry project.

Line 323 `END SUBDIV`

This indicates the end of the SUBDIV block and is required in a geometry project when the SUBDIV block is used.

Line 324 `OPT`

The OPT block is used to specify an optimization. Since no optimizations have been entered in our example project, the default entries are set.

Line 325 `MAX 100`

This entry is the maximum number of iterations for optimization. The 100 iterations set here is the default value.

Line 326 `VARS`

This entry is the beginning of the variables section of the optimization block. Each line after this entry defines a variable’s use and range of values.

Line 327 `Top N UNDEF UNDEF`

This is the default entry for a variable when no optimization is specified. The N indicates that Top is not being used and its data range is undefined. If you were using this variable the N would be a Y and the next two fields would define the minimum and maximum values to be used for the variable.

Line 328 `Radial N UNDEF UNDEF UNDEF`

This is the default entry for the variable Radial, the definition is the same as specified for the variable Top in the line above.

Line 329 `END OPT`

This is the end of the OPT block. There are additional entry lines used in the OPT block that were not used here since there is no optimization defined. For details about the complete syntax of the OPT block, see “OPT,” page 89.

Line 330 `VARSWP`

Line 331 `END VARSWP`

The VARSWP block is used to specify a parameter sweep. Since there is no parameter sweep entered in our example project, the block is empty except for the beginning and end statements. For the complete syntax of the VARSWP block, see “VARSWP,” page 91.

Line 332 `FILEOUT`

Line 333 `END FILEOUT`

The FILEOUT block is used to specify an optional output file. Since there is no optional output file specified in our example project, the block is empty except for the beginning and end statements. For the complete syntax of the FILEOUT block, see “FILEOUT,” page 93.

Line 334 `QSG`

The QSG block is used to specify the settings in the Quick Start Guide. The following seven lines specify which tasks in the Quick Start Guide have been done by the user. If the task is done a YES appears after the keyword. If the task has NOT been done, a NO appears after the keyword. For the complete syntax of the QSG block, see “Quick Start Guide Block for a Geometry Project,” page 107.

Line 335 `IMPORT NO`

This line specifies if a DXF or GDS import has been performed in the project editor.

Line 336 `EXTRA_METAL NO`

This lines specifies if extra metal has been removed from the circuit.

Line 337 `UNITS NO`

This line specifies if the user has changed the units used in the project.

Line 338 `ALIGN NO`

This line specifies if the user has aligned the circuit to the grid.

Line 339 `REF YES`

This line specifies if reference planes have been added to the circuit.

Line 340 `VIEW_RES NO`
This line specifies if the user had viewed response data.

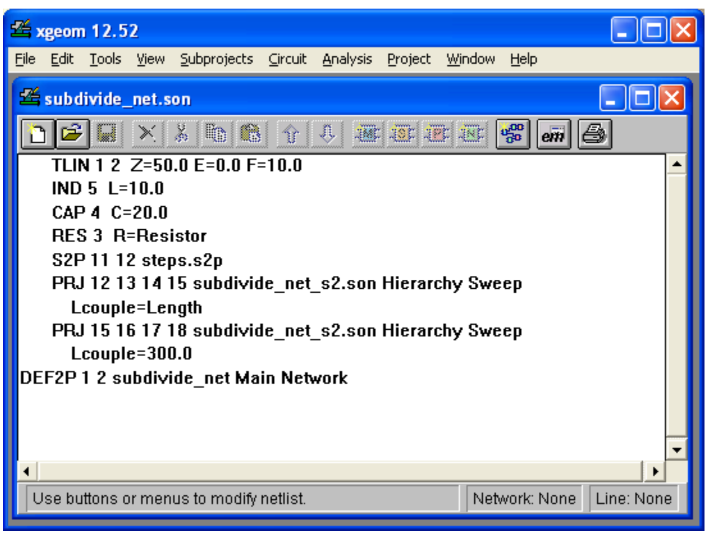
Line 341 `METALS YES`
This line specifies if the user has defined any new metal types.

Line 342 `USED YES`
This line indicates if the Quick Start Guide is enabled for this project.

Line 343 `END QSG`
This is the end of the Quick Start Guide block. This concludes the example. Below is the example for a netlist project file. If you wish to see the complete syntax for the project file, see “Project File Syntax,” page 54.

Netlist Project Example

The file subdivide_net.son is pictured below as it appears in the project editor, ***xgeom***, followed by the file listing as it would appear in a text editor.



```
FTYP SONNETPRJ 4 ! Sonnet Netlist Project File
VER 12.56
HEADER
LIC sonnet9.aa.99999
DAT 04/30/2009 16:12:12
BUILT_BY_CREATED xgeom 12.56 04/30/2009 15:39:01
BUILT_BY_SAVED xgeom 12.56
MDATE 04/30/2009 16:12:12
HDATE 04/30/2009 16:12:12
END HEADER
DIM
FREQ GHZ
IND NH
LNG MIL
ANG DEG
CON /OH
CAP PF
RES OH
END DIM
VAR
Resistor = 50.0
Length = 200.0
END VAR
CKT
TLIN 1 2 Z=50.0 E=0.0 F=10.0
IND 5 L=10.0
CAP 4 C=20.0
RES 3 R= Resistor
S2P 11 12 steps.s2p
PRJ 12 13 14 15 subdivide_net_s2.son 4  1 DATE 04/30/2009 15:55:12 Lcouple= Length
PRJ 3 4 5 2 subdivide_net_s2.son 4  1 DATE 04/30/2009 15:55:12 Lcouple=300.0
DEF2P 1 2 subdivide_net R 50.00000

END CKT
FREQ
ABS 4.0 8.0
END FREQ
CONTROL
ABS
OPTIONS  -d
PUSH
SPEED 0
CACHE_ABS 1
TARG_ABS 300
Q_ACC N
```

```
46. END CONTROL
47. OPT
48. MAX 100
49. VARS
50. Resistor N UNDEF UNDEF UNDEF
51. Length N UNDEF UNDEF UNDEF
52. END OPT
53. VARSWP
54. END VARSWP
```

Line 1 `FTYP SONNETPROJ 4! Sonnet Netlist Project File`

This statement is mandatory at the beginning of a netlist project.

Line 2 VER12.56

This indicates the version of the project editor, which in this case is 12.56. Not required.

Line 3 `HEADER`

This command begins the HEADER block of the project file. This block contains licensing and date creation information on the project file.

Line 4 `LIC sonnet9.aa.99999`

License ID for the license used to create the project file. Not required.

Line 5 `DAT 04/30/2009 15:57:03`

The date the file was last saved. Format is MM/DD/YYYY.

Line 6 `BUILT_BY_CREATED xgeom 12.56 04/30/2009 15:39:01`

This indicates that this project file was created using the Sonnet project editor, version 12.56, at the date and time shown.

Line 7 `BUILT_BY_SAVED xgeom 12.56`

This indicates that the project file was last saved by the Sonnet project editor, version 12.56.

Line 8 `MDATE 04/30/2009 15:57:03`

Date that file was last saved with “Medium Importance” changes.

Line 9 `HDATE 04/30/2009 15:57:03`
Date that file was last saved with “High Importance” changes.

Line 10 `END HEADER`
This indicates the end of the HEADER block. Required.

Line 11 `DIM`
This indicates the beginning of the dimensions block. This provides the units used for various circuit elements.

Line 12 `FREQ GHZ`
The frequencies are in gigaHertz.

Line 13 `IND NH`
The inductor values are in nanoHenries.

Line 14 `LNG MIL`
Lengths in the geometry are in mils.

Line 15 `ANG DEG`
Angles are in degrees.

Line 16 `CON /OH`
The conductivity values are in siemens/meter.

Line 17 `CAP PF`
The capacitance values are in picofarads.

Line 18 `RES OH`
The resistance values are in Ohms.

Line 19 `END DIM`
This indicates the end of the dimensions block. Required if there is a DIM statement in the file.

Line 20 `VAR`
This indicates the beginning of the VAR block which is used in the netlist to define parameters. There is a single entry line for each parameter in the netlist. This block is used only for netlist projects.

If there are no parameters in the netlist the VAR entry appears followed directly by the END VAR statement.

Line 21 `Resistor = 50.0`
The first parameter is Resistor whose present nominal value is 50.

Line 22 `Length = 200.0`
The second parameter is Length whose present nominal value is 200.

Line 23 `END VAR`
This indicates the end of the VAR block.

Line 24 `CKT`
This entry is the beginning of the CKT block which defines the netlist circuit. This block is only used in a netlist project. There is an entry line in this block for each entry in your netlist file.

Line 25 `TLIN 1 2 Z=50.0 E=0.0 F=10.0`
This line is a transmission line modeled element. The element is connected to node 1 and 2 with an impedance of 50 ohms, an electrical length of 0 degrees and frequency = 10 GHz.

Line 26 `IND 5 L=10.0`
This line is an inductor modeled element connected at node 5 with a value of 10 nanoHenries.

Line 27 `CAP 4 C=20.0`
This line is a capacitor modeled element connected at node 4 with a value of 20 picoFarads.

Line 28 `RES 3 R= Resistor`
This line is a resistor modeled element connected at node 3. The value of the resistor is set equal to the variable Resistor.

Line 29

`S2P 11 12 steps.s2p`

This line is a data file element entry connected between nodes 11 and 12. The data file is steps.s2p.

Line 30

`PRJ 12 13 14 15 subdivide_net_s2.son 4  1 DATE 04/30/2009 15:55:12 Lcouple= Length`

This line is a project file element. The four ports in subdivide_net_s2.son correspond to nodes 12, 13, 14 and 15 in the netlist, respectively. The sweep control is set to use the subproject's sweep.

Subdivide_net_s2.son contains a variable, Lcouple. This subproject variable is set equal to the variable Length defined in the netlist.

Line 31

`PRJ 3 4 5 2 subdivide_net_s2.son 4  1 DATE 04/30/2009 15:55:12 Lcouple=300.0`

This line is a project file element. The four ports in subdivide_net_s2.son correspond to nodes 3, 4, 5 and 2 in the netlist, respectively. The sweep control is set to use the subproject's sweep. The variable in the project file, Lcouple is set equal to 300 mils.

Line 32

`DEF2P 1 2 subdivide_net R 50.00000`

This entry defines the main network. This netlist has 2 ports, port 1 is at node 1 of the netlist and port 2 at node 2 of the netlist. The port terminations are set to 50.0 ohms. The network name is subdivide_net.

Line 33

`END CKT`

This entry indicates the end of the CKT block. This is a mandatory entry.

Line 34

`FREQ`

This indicates the beginning of the frequency sweep block. This section details all the frequency sweeps specified for the project. If there is more than one type of sweep present in this block, you must refer to the Control block to see which type of sweep is presently being used in the project file. For a detailed syntax description of this block and any possible entries, see "FREQ," page 58.

Line 35

`ABS 4.0 8.0`

This is an ABS sweep from 4.0 GHz to 8.0 GHz.

Line 36 `END FREQ`
This indicates the end of the FREQ block.

Line 37 `CONTROL`
This indicates the beginning of the Control block which is a mandatory block. This section details the frequency sweep presently being used as well as run options and Speed/Accuracy control. The Hierarchy sweep setting is also included in this block for a netlist project. All possible entries are not present in this example file; for a detailed syntax description of this block and any possible entries, see “CONTROL,” page 83.

Line 38 `ABS`
This indicates that the presently defined sweep is ABS which corresponds to an Adaptive Band Synthesis Sweep in the analysis setup. This means that the ABS statement in the FREQ block is used.

Line 39 `OPTIONS -d`
The de-embedding option is selected as indicated by the “-d”. For the detailed syntax of the OPTIONS statement, see “OPTIONS,” page 84.

Line 40 `PUSH`
This entry indicates that the Hierarchy sweep option is on. The sweep defined in the netlist is used in the analysis of any subproject element in the netlist. This entry only appears when the Hierarchy sweep is on. If the Hierarchy sweep is off, this statement is omitted. This entry may only be used in a netlist project.

Line 41 `SPEED 0`
This entry indicates the position of the Speed/Memory Slider and is only valid for Geometry projects. The default of SPEED 0 appears in a netlist project but is ignored by ***em*** during the analysis.

Line 42 `CACHE_ABS 1`
This indicates the ABS caching level set in the Advanced Options dialog box. A value of 1 indicates that the ABS caching level is Stop/Restart which is the default value. For a detailed syntax description, see “CACHE_ABS,” page 87

Line 43 `TARG_ABS 300`
This indicates that the target for the number of frequencies for the ABS sweep is 300. This statement is for the Automatic setting for ABS Frequency Resolution found in the Advanced Options dialog box. Since the RES_ABS statement does not appear in this project, Automatic, which is the default, is selected. For a detailed syntax descriptions of both entries, see "TARG_ABS," page 88 and "RES_ABS," page 87.

Line 44 `Q_ACC N`
This statement determines if the Q-Factor Accuracy option in the Advanced Options dialog box is being used. "N" indicates that the value is not being used.

Line 45 `END CONTROL`

Line 46 `OPT`
This entry is the beginning of the OPT block which specifies optimization goals and data ranges for parameters. An optimization has not been input for this netlist, so the following entries are the default.

Line 47 `MAX 100`
The MAX entry specifies the maximum number of iterations for an optimization. 100 is the default value.

Line 48 `VARS`
This entry indicates the beginning of the list of variables. There is an entry for each variable defined in the netlist.

Line 49 `Resistor N UNDEF UNDEF UNDEF`
The first variable is "Resistor" which is not selected for optimization. The data range has UNDEF for both the starting and ending values and granularity since none have yet been entered.

Line 50 `Length N UNDEF UNDEF UNDEF`
The second and last variable is "Length" which is not selected for optimization. The data range has UNDEF for both the starting and ending values and granularity since none have yet been entered since no optimization has been input.

Line 51 `END OPT`
This indicates the end of the OPT block.

Line 52 `VARSWP`
This entry is the beginning of the parameter sweep block which specifies data ranges for variables and analysis frequencies for a parameter sweep. There is no parameter sweep defined for this project so this block contains no entries.

Line 53 `END VARSWP`
This indicates the end of the parameter sweep block.

This completes the example of the netlist project. The complete syntax for the Sonnet project follows.

Project File Syntax

Any line with a first non-space character of "!" unless immediately followed by a "<", is ignored. Any blank line is also ignored. "&" is used as a continuation character. When it occurs, anything after it in that line is ignored and the next entry line is added at the point at which the "&" was placed. Comments following any complete line of data are allowed. Comments are not allowed in the polygon vertex lists.

In the keywords that follow, only the specified letters (3 or 4) are significant. Additional letters may be used but they do not alter the program's execution. For example, "VER", "VERSION" and "VERTRFGH" all have the same effect. There may be no more than 255 characters per line.

The .son file is comprised of the following blocks. All blocks start with a keyword and are ended with the statement END <Keyword>.

HEADER
DIM
GEO (Geometry projects only)
FREQ
CONTROL
OPT

VARSWP
FILEOUT
VAR (Netlist projects only)
CKT (Netlist projects only)
SUBDIV (Geometry projects only)

The blocks and corresponding keywords appear below in the order in which they should appear in the project file. It is especially important to note that index numbers for metal type, dielectric type, polygons and polygon vertices in geometry netlists are implicitly assigned by where the entries appear in a file. The keyword is followed by a brief description with the complete syntax following.

FTYP File type statement identifies the file type: Geometry or Netlist.

Syntax `FTYP [SONPROJ|SONNETPROJ] 3`

FTYP is followed by SONPROJ for a geometry project and followed by SONNETPROJ for a netlist project. The 3 is mandatory and is reserved for future use by Sonnet.

VER Project Editor Version number. The VER line should be the first non-comment line in the file.

Syntax VER version

The version is a character string identifying the version of the project editor, for example, 7.0a. This entry is not required.

Header Block

The header block provides license and date information about the project file. **Special Note:** No Comments (!) or continuations (&) are allowed inside the header block since they will be read as standard characters. This is to allow any characters to be used in the ANN lines.

HEADER Beginning of Header Block

Syntax `HEADER`

Indicates the beginning of the header block. All statements following this entry are included in the Header block until you reach the END HEADER statement.

LIC License id number.

Syntax `LIC licenseID`
The License ID is a character string identifying the customer license ID.

DAT Date of last file change.

Syntax `DAT mm/dd/yyyy hh:mm:ss`
The DAT keyword is followed by a character string identifying the date and time the file was last saved.

BUILT_BY_CREATED Origin of project file.

Syntax `BUILT_BY_CREATED source version mm/dd/yyyy hh:ss`
The `BUILT_BY_CREATED` keyword is followed by the source program name, the version of the source program, then the date and time when the file was created. This entry is never updated. If the file source is unknown then "`source`" is set to "unknown" and `version` is set to "unknown version". The following are possible file sources:

Entry	Program
xgeom	Sonnet Project Editor
sonntgbr	Gerber Translator
ebridge	Agilent ADS Interface
gds	GDSII Translator
dxfgeo	DXF Translator
sonntawr	AWR Microwave Office
sonntcds	Cadence Virtuoso Interface

BUILT_BY_SAVED Origin for the last time the file was saved.

Syntax `BUILT_BY_SAVED source version`

The `BUILT_BY_SAVED` keyword is followed by the source program which last executed a save on the project file. This entry is updated each time the project file is saved. See the `BUILT_BY_CREATED` syntax in the previous entry for a description of the fields.

MDATE Date of last file change with "Medium Importance" changes.

Syntax `MDATE mm/dd/yyyy hh:mm:ss`

The MDATE keyword is followed by a character string identifying the date and time the file was last saved with "Medium Importance" changes. Not required.

HDATE Date of last file change with "High Importance" changes.

Syntax `HDATE mm/dd/yyyy hh:mm:ss`

The HDATE keyword is followed by a character string identifying the date and time the file was last saved with "High Importance" changes. High importance changes are those which cause analysis data to be invalid. Not required.

ANN Comment Statements

Syntax `ANN text`

This keyword is followed by any comments about the file. This is the only location in the header block that special comments are allowed. If you comment is longer than one line, you may use multiple ANN lines.

END End statement

Syntax `END HEADER`

Indicates the end of the header block. Required.

Dimensions Block

The dimensions block provides all the units to be used in the project file. There is an entry line for each type of unit.

DIM Beginning of dimensions block

Syntax `DIM`

Indicates the beginning of the dimensions block. All statements following this entry are included in the dimensions block until you reach the END DIM statement.

FREQ Frequency Units

Syntax `FREQ <unit>`

The FREQ keyword is followed by a character string identifying the frequency unit. Choices for <unit> include HZ, KHZ, MHZ, GHZ, THZ, and PHZ. The default is GHZ.

IND Inductance Units

Syntax `IND <unit>`

The IND keyword is followed by a character string identifying the inductance unit. Choices for <unit> include H, MH, UH, NH, PH, and FH. The default is NH.

LEN Length Units.

Syntax `LEN <unit>`

The LEN keyword is followed by a character string identifying the length unit. Choices for <unit> include MIL, UM, MM, CM, IN, and M. The default is MIL.

ANG Angle Units.

Syntax `ANG <unit>`

The ANG keyword is followed by a character string identifying the angle unit. Choices for <unit> include only DEG (degrees) at the present time.

CON Conductivity Units.

Syntax `CON <unit>`

The CON keyword is followed by a character string identifying the conductivity unit. The conductivity unit defaults to siemens/meter.

RES Resistance Units.

Syntax `RES <unit>`

The RES keyword is followed by a character string identifying the resistance unit. Choices for <unit> include OH, KOH, and MOH. The default is OH.

END End statement

Syntax `END DIM`

Indicates the end of the dimensions block. Required.

Geometry Block for Geometry Project

The geometry block specifies the circuit geometry in a geometry project.

GEO Beginning of geometry block

Syntax `GEO`

Indicates the beginning of the geometry block. All statements following this entry are included in the geometry block until you reach the END GEO statement.

SYM Enables symmetry for the circuit.

Syntax `SYM`

If the circuit geometry AND excitation are symmetric about the center line parallel to the X axis, a significant reduction in computation time can be realized. To indicate this, include a line with “SYM” as the first non-space characters. In this case, a mirror image of the geometry of the half substrate above the symmetry line is used on both halves of the substrate.

PSB1 Parallel Subsections

Syntax `PSB1 boxside distance`

This statement defines the parallel subsections in the geometry project. <boxside> is the side of the box to which the parallel subsections are attached. The value is a character string; possible values are “LEFT” “RIGHT” “TOP” “BOTTOM”. <distance> is a floating point number which provides the distance from the box wall to which the parallel subsections extend. There may be up too four PSB1 statements, one for each box wall, in the GEO block.

DRP1 Reference Planes

Syntax

```
DRP1 Position type [LEN|
POLY iPolygon #Points
iVertex]
```

Each side of the Box may have a reference plane offset associated with it. Each DRP1 statement defines one reference plane, so that there may be up to four DRP1 statements in a project file. All ports on a given side have the same offset. The Position defines the box wall from which the reference plane extends. Values for Position are LEFT, RIGHT, TOP, and BOTTOM.

If <type> is FIX, then the LEN field is a floating point number specifying the length of the reference plane in the units presently specified in the DIM block. The actual reference plane is moved to the nearest cell location (see BOX).

If <type> is LINK, then the DRP1 entry is following by additional entry lines specifying the polygon vertex to which the reference plane is linked. See the POLY statement syntax below.

Syntax

```
POLY iPolygon #Points
iVertex
```

This statement is used to specify the vertex to which the reference plane is linked. The keyword POLY is followed by an integer which identifies the polygon. This is the file ID, not the position in the GEO block. This is followed by the number of points to be specified. In the case of a linked reference plane this value is always 1.

The next line consists of an integer number which identifies the vertex of the polygon specified in the POLY statement above. The first polygon vertex is number zero.

TMET Top Cover Metal- only one allowed per file.

Syntax

```
TMET name patternid type value1 value2 ...
```

`name` This is the metal name. This name must be in quotes if it contains spaces. Quotes may be used when they are not required.

`patternid` The index of the pattern to be displayed. Starts at 0.

`type` This identifies the type of metal definition. This is one of the following character strings: WGLOAD, FREESPACE, NOR, RES, NAT, SUP, SEN. These keywords and the values associated with them are defined below.

`WGLOAD`
Wave Guide Load. There are no values associated with this type.

`FREESPACE 376.7303136 0 0 0`
Free Space. The values model no top being present on the box.

`NOR conductivity currentratio thickness`
Normal metal. The conductivity is a floating point number in S/m. The `currentratio` is a floating point number which is the ratio of current on the top surface to current on the bottom surface. Thickness is a floating point number for the thickness of the metal.

`RESISTOR rdc`
Resistor. R_{dc}, the DC resistance, is a floating point number in Ohm/sq.

`NAT rdc rrf`
Native. R_{dc}, the DC resistance, is a floating point number in ohms/sq. R_{rf}, the skin effect coefficient, is a floating point number.

`SUP rdc rrf xdc ls`
General. R_{dc}, the DC resistance, is a floating point number in ohms/sq. R_{rf}, the skin effect coefficient, is a floating point number. X_{dc}, the DC Reactance, is a floating point number in ohms/sq. Ls, the kinetic inductance, is a floating point number in pH/sq.

`SEN xdc`
Sense metal. Xdc, the DC reactance, is a floating point number in ohms/sq.

Example 1 `TMET “Freespace” 0 FREESPACE 376.7303136 0 0 0`

The four floating point numbers define the metal loss for the top of the circuit enclosure. This is a special case. When FREE SPACE is selected as the top metal in ***xgeom***, the circuit editor, then these loss values are used.

Example 2 `TMET “WG Load” 0 WGLOAD`

When WGLOAD is selected as the top metal in ***xgeom***, the circuit editor, this syntax is used. When WG Load is chosen, ***em*** models a perfect matched waveguide load whose values are used as the metal’s parameters.

BMET Box Bottom Metal - Only one allowed per file.

Syntax `TMET name patternid type value1 value2 ...`

The syntax for the BMET keyword is the same as that of the TMET command detailed above.

MET Metal Type - Unlimited number allowed per file.

`MET name patternid type value1 value2 ...`

`name` This is the metal name. This name must be in quotes if it contains spaces. Quotes may be used when they are not required.

`patternid` The index of the pattern to be displayed. Starts at 0.

`type` This identifies the type of metal definition. This is one of the following character strings: NOR, RES, NAT, SUP, SEN, TMM, ROG. These keywords and the values associated with them are defined below.

`NOR conductivity currentratio thickness`
Normal metal. The conductivity is a floating point number in S/m. The `currentratio` is a floating point number which is the ratio of current on the top surface to current on the bottom surface. Thickness is a floating point number for the thickness of the metal.

`RESISTOR rdc`
Resistor. R_{dc}, the DC resistance, is a floating point number in Ohm/sq.

`NAT rdc rrf`
Native. R_{dc}, the DC resistance, is a floating point number in ohms/sq. R_{rf}, the skin effect coefficient, is a floating point number.

`SUP rdc rrf xdc ls`
General. R_{dc}, the DC resistance, is a floating point number in ohms/sq. R_{rf}, the skin effect coefficient, is a floating point number. X_{dc}, the DC Reactance, is a floating point number in ohms/sq. Ls, the kinetic inductance, is a floating point number in pH/sq.

`SEN xdc`
Sense metal. Xdc, the DC reactance, is a floating point number in ohms/sq.

`TMM conductivity currentratio thickness numsheets`
Thick Metal Model. The conductivity is a floating point number in S/m. The `currentratio` is unused for this metal type. Thickness is a floating point number for the thickness of the metal. The number of sheets is an integer value indicating the number of sheets to be used to model the thick metal.

`ROG thickness RMSroughness numsheets currentratio top-surface`
Roger's Surface Roughness Model. Thickness is a floating point number for the thickness of the metal. The RMSroughness is a floating point number for the number of microns that are the average variation (root mean square) in the peaks and valleys in the surface of the metal. Numsheets is an integer value of 1 or 2 indicating the number of sheets to be used to model the thick metal. The currentratio is a floating point number which is used to adjust the total effective surface impedance. For the effect of this value, please refer to the beta documentation on the Roger's Surface Roughness. Note that this value is only used if numsheets = 1. topsurface is one of two character strings: "RR" for rough or "SR" for smooth. This field applies to the top level of the thick metal; the

bottom surface of the thick metal is always rough. For the effect of this value, please refer to the beta documentation on the Roger's Surface Roughness. Note that if numsheets = 1, this value is always set to "SR."

Example 1 `MET "Capbot" 1 SUP 0.02439 3.1e-007 0 0`

Example 2 `MET "Metal1" 2 RES 13`

DIM Dimensions

Syntax

```
DIM STD direction sign
POS xcoord ycoord
NOM nvalue
REF1 POLY idfile 1
ivertex
REF2 POLY idfile 1
ivertex
END
```

The DIM statement defines a dimension in your geometry project. The name, type and orientation of the parameter appear on the first line. Subsequent statements define the label position, nominal value, and reference points for the dimension. This set of statements appears for each dimension in the project. The syntax is detailed below.

`STD` This is the type of dimension. Presently, there is only a standard dimension so that this field is always "`STD`".

`direction` This field is either "`XDIR`" for a parameter in the x plane (horizontal) or "`YDIR`" for a parameter in the y plane (vertical).

`sign` This field is either "`1`" for a parameter whose reference position is greater than its anchor position and "`-1`" for a parameter whose reference position is less than its anchor position. The coordinates are based on the upper left hand corner of the substrate being point (0,0).

POS Label Position

Syntax

```
POS xcoord ycoord
```

The POS entry provides the position of the dimension label. The label always appears in the center of the dimension in the direction of the dimension. For an x directed dimension, <ycoord >is a floating point number which provides the position of the label as an offset from the first reference point in the y direction. For a y directed dimension, <xcoord> is floating point number which provides the position of the label as an offset from the first reference point in the x direction.

NOM Nominal Value of Dimension

Syntax

```
NOM nomvalue
```

This entry is the nominal value of the dimension. The keyword NOM is followed by a floating point number which is the nominal value of the dimension.

REF1 Reference Point 1

Syntax

```
REF1 POLY idfile 1
vertex
```

This entry identifies the first reference point of a dimension. The REF1 keyword is followed by the POLY statement on the same line. <idfile> is the file ID for the polygon which contains the first reference point. This is followed by the number of points on the polygon which for a dimension is always 1. The next line lists which vertex of the polygon is used as the first reference point.

REF2 Reference Point 2

Syntax

```
REF2 POLY idfile 1
vertex
```

This entry identifies the second reference point of a dimension. The REF2 keyword is followed by the POLY statement on the same line. <idfile> is the file ID for the polygon which contains the second reference point. This is followed by the number of points on the polygon which for a dimension is always 1. The next line lists which vertex of the polygon is used as the second reference point.

END End Dimension

Syntax `END`

This end statement indicates the completion of defining a dimension.

BRI The specification for an isotropic dielectric material.

Syntax BRI "name" pattern_id erel loss_tan diel_cond

A BRI line is used to specify an isotropic dielectric brick material. The BRI keyword is followed by the name of the dielectric in quotes and may contain spaces.This is followed by the index of the fill-pattern, then three floating point values. These values are the relative dielectric constant, the loss tangent and the bulk conductivity of the dielectric material.

BRA The specification for an anisotropic dielectric material.

Syntax BRA "name" pattern_id x_erel x_loss_tan x_diel_cond y_erel y_loss_tan y_diel_cond z_erel z_loss_tan z_diel_cond

A BRA line is used to specify an anisotropic dielectric brick material. The BRA keyword is followed by the name of the dielectric in quotes and may contain spaces. This is followed by the index of the fill-pattern, then nine floating point values. In order, these values are the following:

- X relative dielectric constant
- X loss tangent
- X bulk conductivity
- Y relative dielectric constant
- Y loss tangent
- Y bulk conductivity
- Z relative dielectric constant
- Z loss tangent
- Z bulk conductivity

VALVAR Variables

Syntax `VALVAR varname unittype value “description”`

The VALVAR statement defines a variable in your project. The name, units and present nominal value of the variable and a brief description are defined as detailed below.

`varname` The name of the variable. This is a character string.

`unittype` This field is one of six choices shown in the table below. The type of unit is shown here. The actual units used are the default for that type of unit in the project. For example, if the unittype is "LNG" and the project is presently using mils, then the variable's value is in mils.

unittype	Type of Units
LNG	Length
RES	Resistance
CAP	Capacitance
IND	Inductance
FREQ	Frequency
OPS	Ohms/sq
SPM	Siemens/meter
PHPM	picoHenries/meter
RRF	R_{rf}
NONE	Undefined

`value` This field is the present nominal value of the variable and is a floating point number. If the variable definition is an equation, the equation is a character string which appears in this field in quote marks. For example, you have a variable, Length that is defined as 2*Width. In that case "2*Width" would appear in the `value` field. For details on the equation syntax, please refer to "Equation Syntax" in Help.

`"description"` This field is a description of the variable and is a character string.

GEOVAR Parameters

Syntax

```
GEOVAR parname partype direction sign scaletype
POS xcoord ycoord
NOM nvalue
REF1 POLY idfile numpol
ivertex
REF2 POLY idfile numpol
ivertex
EQN "vareqn"
PS1 numpt
POLY idfile numpt
ivertex(1)
.
.
ivertex(n)
END
PS2 numpt
POLY idfile numpt
ivertex(1)
.
.
ivertex(n)
END
END
```

The GEOVAR statement defines a dimension parameter in your geometry project. The name, type and orientation of the parameter appear on the first line. Subsequent statements define the label position, nominal value, and point sets for the parameter. This set of statements appears for each parameter in the project including linked parameters. The syntax is detailed below.

`parname` The name of the parameter. This is a character string.

`partype` This field is either ANC for an anchored parameter, SYM for a symmetric parameter or RAD for a radial parameter.

`direction` This field is either "`XDIR`" for a parameter in the x plane (horizontal) or "`YDIR`" for a parameter in the y plane (vertical).

`sign` This field is either "`1`" for a parameter whose reference position is greater than its anchor position and "`-1`" for a parameter whose reference position is less than its anchor position. The coordinates are based on the upper left hand corner of the substrate being point (0,0).

`scaletype` This field indicates if and how scaling is applied to a dimension parameter when its size is changed. This field is only used for anchored and symmetric dimension parameters. The three possible values are `NSCD`, `SCUNI`, and `SCXY`. `NSCD` is used if no scaling is being applied; for a radial parameter this value is always used. `SCUNI` indicates that scaling is only applied in the direction in which the dimension parameter is oriented. `SCXY` indicates that the dimension parameter is scaled in both the x and y direction.

POS Label Position

Syntax `POS xcoord ycoord`

The POS entry provides the position of the parameter label. The label always appears in the center of the parameter in the direction of the parameter. For an x directed parameter, ycoord is a floating point number which provides the position of the label as an offset from the first reference point in the y direction. For a y directed parameter, xcoord is floating point number which provides the position of the label as an offset from the first reference point in the x direction.

NOM Nominal Value of Parameter

Syntax `NOM nomvalue`

This entry is the nominal value of the parameter. The keyword NOM is followed by a floating point number which is the nominal value of the parameter.

REF1 Reference Point 1

Syntax

```
REF1 POLY idfile 1
vertex
```

This entry identifies the first reference point of a symmetric parameter or the anchor point of an anchored parameter. The REF1 keyword is followed by the POLY statement on the same line. <idfile> is the file ID for the polygon which contains the first reference point. This is followed by the number of points on the polygon which for the reference point is always 1. The next line lists which vertex of the polygon is used as the first reference point. The vertices are numbered starting at 0.

REF2 Reference Point 2

Syntax

```
REF2 POLY idfile 1
vertex
```

This entry identifies the second reference point of a symmetric parameter or the reference point of an anchored parameter. The REF2 keyword is followed by the POLY statement on the same line. <idfile> is the file ID for the polygon which contains the second reference point. This is followed by the number of points on the polygon which for the reference point is always 1. The next line lists which vertex of the polygon is used as the second reference point.

EQN Equation or variable

This entry is optional and only appears if the nominal value of the variable assigned to the dimension parameter is defined by another variable or equation. The keyword is followed by a character string in quote marks which defines the variable or equation. For details on equation syntax and available functions, please see “Equation Syntax” in Help.

PS1 Point Set 1

Syntax

```
PS1 numpol
POLY idfile numpt
ivertex(1)
.
.
ivertex(n)
END
```

This entry starts the specification of the first point set associated with a parameter. The PS1 keyword is followed by an integer of the number of polygons which contain points in the point set. For an anchored or radial dimension parameter, there is no first point set, so numpol is always 0.

For each polygon with points in the point set, there is a POLY entry followed by vertices statements.<idfile> is the file ID for the polygon. numpt is the number of points on the polygon which are in the point set. Each point in the point set has a line which contains the vertex number on the polygon of the point. When all the points in the point set have been entered an END statement appears to indicate the end of the point set specification.

For example. If vertices 2 and 4 of polygon 13 and vertex 4 of polygon 17 are included in the point set, the input in the file would be as follows:

```
PS1 2
POLY 13 2
2
4
POLY 17 1
4
END
```

PS2 Point Set 2

Syntax

```
PS2 numpol
POLY idfile numpt
ivertex(1)
.
.
ivertex(n)
END
```

This entry starts the specification of the second point set associated with a parameter. The PS1 keyword is followed by an integer of the number of polygons which contain points in the point set. The second point set is specified using the same syntax as the first point set documented above.

END End of parameter specification.

Syntax

```
END
```

Once both point sets have been specified for a GEOVAR statement, an END statement appears to indicate the end of this GEOVAR statement. There is a GEOVAR statement for each parameter in a project including linked parameters.

BOX The dimensions of the box and the parameters of the enclosed substrates.

Syntax

```
BOX nlev xwidth ywidth xcells2 ycells2 nsubs eeff
    thickness erel mrel eloss mloss esignma nzpart name
    thickness erel mrel eloss mloss esignma nzpart name
                    .
                    .
    thickness erel mrel eloss mloss esignma nzpart name
```

If the dielectric is anisotropic then the following syntax is appended to the end of the line and these values are used for the dielectric properties in the Z direction:

```
    A erel mrel eloss mloss esignma
```

The dimensions of the box and the parameters of the enclosed substrates are described with the multi-lined BOX statement. BOX is followed by at least five numbers with two more optional. First, <nlev>, is the number of metalization levels (= 1 if using standard microstrip). Next are the X dimension, <xwidth> (left to right) and Y dimension, <ywidth> (top to bottom), dimensions of the box given in the units specified by the LEN statement. These are followed by two integers: 2 times the number of cells in the X-dimension, <xcells>, and then 2 times the number of cells in the Y-dimension, <ycells2>. The first optional number is an integer which historically set the minimum density of subsections per wavelength but is no longer used. The default value of 20 should be used as a placeholder. The second optional number is the effective dielectric constant (E_{eff}) used to calculate the wavelength for satisfying the subsections/wavelength parameter. If not specified, or if it is less than 1.0, the parameter is ignored and a simple estimate of E_{eff} is used. Please note that you must specify the first optional number in order to use the second number. The last entry is the name of the dielectric used for the dielectric layer. The name needs to appear in quotes if there is a space. You may use quotes around the name when they are not required.

The lines following BOX provide information on each of the dielectric layers in the box. Note that the number of layers is one more than the number of levels, <nlev>. Each line has the thickness of the layer (in the units previously specified) followed by the relative dielectric constant and the relative permeability of the layer. The fourth and fifth parameters are the dielectric and magnetic loss tangents, respectively. The sixth parameter is the bulk conductivity for the dielectric layer. The seventh and final parameter is the number of z-partitions. This parameter is used for subsectioning dielectric bricks. All items except the substrate thickness are optional. The default is lossless free space. Only one dielectric layer per line is allowed.

EVIA1 Defines an edge via in the circuit by identifying the polygon edge of origin and the level to which it extends.

Syntax

```
EVIA1
POLY poly_id 1
edge_number
TOLEVEL to_level
```

The POLY statement identifies which polygon the edge via is attached to. The poly_id is the number which identifies the polygon. The “1” is required.

The edge_number indicates the index number of the polygon vertex. The via is placed on the polygon edge specified by this vertex to the next vertex. For example, if vertex 3 is specified, the via extends from vertex 3 to vertex 4 on the polygon.

The TOLEVEL statement indicates to which level the via extends. The edge via extends from the level the polygon it is attached to is on to the level identified in to_level.

POR1 Defines a port in the circuit by identifying the polygon edge of origin and specifying its parameters.

Syntax

```
POR1 type groupid
POLY ipolygon #points
ivertex
portnum resist react induct capac xcoord ycoord [reftype
rpcallen]
```

There is an POR1 statement for each port in the circuit. The POR1 statement identifies the type of port. <type> is either STD for standard, AGND for autogrounded, or CUP for co-calibrated port.

If <type> is AGND, then the terms <reftype> and <rpcallen> are used. These values do not appear for a STD port.

If <type> is CUP, then the type is followed by groupid which is a character string identifying the co-calibrated port group. This character string is a letter or letters that is unique for each group of co-calibrated ports in the project. The CUPGRP block is used for this type to define the properties of a co-calibrated group of ports. See “CUPGRP,” page 75 below for details.

The POLY statement which follows identifies the polygon. <ipolygon> is the file ID for the polygon on which the port is placed. The number of points, <#points>, is always 1 for a POR1 statement.

The <ivertex> indicates the index number of the polygon vertex. The port is placed on the polygon edge specified by this vertex to the next vertex. For example, if vertex 3 is specified, the port is placed between vertex 3 and vertex 4 on the polygon.

A port is usually placed at the edge of the substrate so that a connection to ground is readily available (the sidewall). Otherwise a port is placed at the junction of two polygons (each polygon has a line segment colinear with the other). In this case, the two terminals of the port are formed by the two polygons.

The next value, <portnum>, is the port number. Port numbers are automatically assigned starting at 1 and incrementing upward as ports are added to the circuit. This number cannot be zero, but may be any other value. It need not be consecutive. If the number is the same as the number of any other port, the ports will be connected together electrically. If the number is negative, it will be connected together with any other negative numbers of the same value and the total current going out of the negative ports will be made equal to the total current going into all the ports with the positive number.

Four optional floating point numbers follow representing the impedance to which the port S-parameters will be normalized. The first is resistance (ohms), the second is reactance (siemens), the third is inductance (henries) and the fourth is capacitance (farads). The final two values are location of the port on the substrate; the x-coordinate is first, followed by the y-coordinate. The origin (0,0) is the upper left hand corner of the substrate.

If <type> is AGND, then <reftype> is a character string which identifies is a reference plane is used for the autogrounded port. <reftype> is FIX for a reference plane and NONE for a calibration length. <rpcallen> is a floating point number which provides the length of the reference plane when <reftype> is FIX and provides the calibration length when <reftype> is NONE. If there is no reference plane or calibration length, then the entry reads “NONE 0”.

CUPGRP Defines a co-calibrated group of ports by identifying the ground reference, the terminal width and reference planes and/or calibration lengths.

Syntax

```
CUPGRP groupid
ID idnum
GNDREF grndref
TWTYPE termwidth
TWVALUE width
DRP1 position type reflength callength
END
```

<groupid> is a character string consisting of a letter or letters which provides a unique identifier for a calibration group of ports. The <idnum> is an integer number used to identify the group in the project. This should be unique in the project. <grndref> defines what the calibration group uses for a ground reference. Values for this field are “F” for floating and “B” for Sonnet Box. <termwidth> defines the terminal width for the calibration group. Values for this field are “FEED” for feedline width, “1CELL” to use the cell size and “CUST” if the terminal width is a value input by the user. If “CUST” is the TWTYPE, then the TWVALUE entry also appears with <width> being a floating point value entered by the user; otherwise, the TWVALUE entry is omitted.

The DRP1 entry line defines the reference planes and/or calibration lengths used for the calibration group. This entry only appears if a single reference plane or calibration length is entered for the calibration group. Each side of the Box may have a reference plane/calibration length offset associated with it, therefore, there may be up to four DRP1 statements for a calibration group. All ports on a given side have the same offset. The Position defines the box wall from which the reference plane extends. Values for Position are LEFT, RIGHT, TOP, and BOTTOM. <type> is always set to FIX for a calibration group reference plane, then the <reflength> field is a floating point number specifying the length of the reference plane. If there is no reference plane, then <reflength> is the character string "NONE." If no calibration length is specified for the position then <calength> does not appear. If a calibration length is entered <calength> is a floating point number specifying the length of the calibration length. The END statement appears to indicate the end of the entry for a calibration group.

SMD Defines a component in the circuit by identifying the type of component, location of the schematic box, port properties, label and reference planes.

Syntax

```
SMD levelnum label
ID =
GNDREF B|F|P
TWTYPE FEED|CUST|1CELL
TWVALUE termwidth
DRP1 ...
SBOX leftpos rightpos toppos bottompos
PBSHW Y|N
PBOX leftpos rightpos toppos bottompos
PKG length width height
LPOS xpos ypos
TYPE ...
SMDP levelnum x y orientation portnum pinnum
```

There is an SMD statement for each component in the circuit. <levelnum> is a positive integer value which identifies the level on which the SMD is located and <label> is an alphanumeric string in quotes, i.e, "comp1" which is the label of the component displayed in the project editor.

The next entry in the SMD entry is the ID where the <objectid> is an integer number which identifies the SMD. This ID number is used elsewhere in the project file to reference this component.

GNDREF defines the ground reference where B is for the Sonnet box, F is for floating (default for ideal components), or P for user defined. If GNDREF is set to P additional port(s) must be defined using the SMDP keyword (see below). These ports will be assigned to a <pinnum> of zero. This setting is not allowed for a Ports Only component (TYPE NONE).

The TWTYPE defines the type of terminal width defined for the component where FEED is for feedline width, CUST is for user defined, or 1CELL for one cell wide.

The TWVALUE entry only appears if the TWTYPE is defined as CUST. This is a floating point number which defines the width of the cell.

If there are reference planes defined for the component, there will be DRP1 entries. For details on the syntax, please see "DRP1" on page 60.

SBOX determines the location of the schematic box or ideal component symbol. The four values are floating string values which represent the location of the four corners of the schematic box, left, right, top, and bottom. Note that these values are relative to the upper left hand corner of the substrate being the origin.

PBSHW indicates if the user has selected to display a package size; Y for yes, or N for no.

The PBOX entry only appears if PBSHW is set to Y. This entry determines the location of the package box using the same syntax as the SBOX entry discussed earlier.

The PKG entry only appears if PBSHW is set to Y and the user has entered dimensions for the package. The <length>, <width> and <height> are all floating point numbers which define the length, width and height of the package respectively.

LPOS provides the location of the center of the component label. <xpos> and <ypos> are floating point numbers which indicate the distance from the origin (upper left hand corner of the substrate) in the x and y direction.

The TYPE entry identifies the type of component and other parameters as detailed below:

- TYPE NONE
  This entry indicates a Ports Only component type. No further information is included.
- TYPE IDEAL idealtype compval
  This entry indicates an ideal component. <idealtype> is either RES for resistor, CAP for capacitor, or IND for inductor. <compval> is a floating point number for the value of the ideal component.
- TYPE SPARAM <paramfileindex>
  This entry indicates a Data File component. <paramfileindex> is an integer value which identifies the file using the index defined in the SMDFILES block, see "SMDFILES" on page 109.

```
SMDP levelnum x y orientation portnum pinnum
```

The SMDP entry defines a component port. These is an SMDP statement for each port of the component. <levelnum> is a positive integer identifying the level on which the port is placed. <x> and <y> provide the location of the port relative to the origin of the upper left hand corner. <orientation> is either T for top, B for bottom, R for right or L for left, indicating in which direction the port is oriented. This field is not used for all component ports. <portnum> is an integer value identifying the port; this number is not displayed in the project editor. <pinnum> is an integer value identifying which pin on the physical component is connected to this component port. This value is set to zero if the port is a ground reference.

NUM The NUM statement is used to define the number and location of polygons in the circuit.

Syntax

```
NUM npoly
type
ilevel nvertices mtype filltype debugid xmin ymin xmax
ymax conmax res res edgemesh
TOLEVEL to_level
xvertex yvertex
    .
    .
xvertex yvertex
```

The number of polygons in the file, <npoly>, follows NUM on the same line. The lines which follow the NUM statement until the END line define an individual polygon. This is the last section of the GEO block.

The first line for each polygon section is the type. This line is optional for a metal polygon and would be set to MET POL if used. If the polygon is a dielectric brick, then this line is BRI POL If the polygon is a via polygon, then this line is VIA POLYGON.

The next line defines the metalization level, number of vertices, metal type, fill type, subsectioning constraints and edge meshing setting. The entry <ilevel> is the circuit metalization level index, which begins with the index 0. The entry <nvertices> is the number of vertices which make up the polygon. The entry <mtype> is the index number which identifies the metal or dielectric brick type for the polygon. If the polygon is metal or a via, an index of -1 indicates the default lossless metal; user-defined metals start at index 0. Indices are assigned implicitly by the location of the appropriate RES statement in the file. The first RES statement is index 0, the next index 1, and so on. If the polygon is a dielectric brick, an index of 0 indicates the default dielectric, air; user defined dielectrics start at index 1. Again, indices are assigned implicitly by the location of the appropriate PRI or PRA statement in the file. The next field, <filltype>, identifies the fill type used for the polygon. N indicates staircase fill, T indicates diagonal fill, V indicates conformal mesh, X indicates Anisotropic X fill and Y indicates AnisotropicY fill. Note that filltype only applies to metal polygons; this field is ignored for dielectric brick polygons. The next two values, <xmin> and <ymin>, define the minimum subsection size in number of cells for each dimension, followed by <xmax> and <ymax> which define the maximum subsection size in number of cells. The default for the minimum in both dimensions is 1, with a default for maximum of 100 for both dimensions. The next value, <conmax>, is the maximum length for a conformal mesh subsection. If this value is zero, the maximum length for a conformal mesh subsection is automatically calculated. The next two values are not presently used but are reserved for future development. Set these to 0. The last field is the edge mesh setting. Y indicates edge meshing is on for this polygon. N indicates edge meshing is off. The subsectioning information, fill type, and edge mesh setting are input in the Metalization Attributes dialog box in the project editor.

The next statement, TOLEVEL, is only used for a via polygon. This is the level to which the polygon extends. It originates on the level identified in the header line above.

The following lines are x,y point pairs that define the location in terms of the substrate grid of all the vertices of the polygon. There is one line per vertex; also note that the first and last vertices are the same location in order to close the polygon off. Vertex indices are assigned implicitly by the location of the appropriate point pair statement in the file. Vertex index numbers start at zero. The data are floating point numbers in the previously specified units (see LEN). The origin point of the grid (0,0) is located at the upper left hand corner of the substrate when viewed in the project editor. Immediately following the last line is an END statement. Subsequent polygons are presented in an identical manner, no blank lines allowed.

Once a NUM line is encountered, the other keywords, described above, are not recognized.

Polygons which have edges in common are electrically connected. Polygons with an edge in common with the edge of the substrate are automatically shorted to ground on that edge, unless there is a port at that point. The polygons may overlap; however it is recommended that they do not. Likewise, a polygon may be complex (it crosses over itself). However this is of little utility.

END End statement

Syntax `END GEO`
Indicates the end of the geometry block. Required.

Frequency Block

The frequency block contains all the frequency sweeps which have been input in a project. Which sweep is presently being used is specified in the CONTROL block. For details about the control block, see “CONTROL,” page 83.

FREQ Beginning of frequency block

Syntax

```
FREQ
```

Indicates the beginning of the frequency block. All statements following this entry are included in the frequency block until you reach the END FREQ statement. The statements in this block are generated in the Frequency Sweep dialog box in the analysis setup.

SWEEP Linear frequency sweep with stated interval.

Syntax

```
SWEEP f1 f2 fstep
```

Linear frequency sweep. <f1> is the starting frequency. <f2> is the ending frequency. <fstep> is the interval between frequencies.

ESWEEP Exponential frequency sweep.

Syntax

```
ESWEEP f1 f2 Nfreq
```

Exponential frequency sweep from <f1> to <f2> with a common ratio between the <Nfreq> frequency points. <f1> is the starting frequency. <f2> is the ending frequency. <Nfreq> is the number of points in the sweep.

LSWEEP Linear frequency sweep with number of points.

Syntax

```
LSWEEP f1 f2 Nfreq
```

Linear frequency sweep from <f1> to <f2>. <f1 is the starting frequency. <f2> is the ending frequency. <Nfreq> is the number of analysis frequencies. <Nfreq> is used to calculate the step size in the following manner:

```
Step Size = (f2-f1)/(Nfreq-1)
```

STEP Discrete analysis frequencies

Syntax

```
STEP f1
```

Discrete frequency. The value is a floating point number. Only one frequency value may be entered per Step statement.

ABS_ENTRY Adaptive Band Synthesis Sweep

Syntax `ABS_ENTRY startfreq stopfreq`

An Adaptive Band Synthesis (ABS) from <startfreq> to <stopfreq>. These two values define the band across which the ABS is being performed.

ABS_FMIN Find the minimum frequency response.

Syntax `ABS_FMIN NET= param f1 f2`

ABS_FMIN finds the frequency at which the minimum frequency response occurs. The <param> field specifies the parameter whose minimum you wish to find. Only S-Parameters may be used. The format is S followed by a pair of port indices. For example, S_{24} would be S24. If one of the two port indices has more than one digit, you use an underscore to separate the two port indices. For example, S10_27.

<f1> is the starting frequency and <f2> is the ending frequency for the ABS sweep that is performed before determining the minimum.

ABS_FMAX Find the maximum frequency response.

Syntax `ABS_FMAX NET= param f1 f2`

ABS_FAX is identical to ABS_FMIN except that it finds the frequency at which the maximum frequency response occurs.

DC_FREQ Analyze at a DC frequency point.

Syntax `DC_FREQ fcalc frequency`

DC_FREQ specifies an analysis at a DC Point. The fcalc field is “AUTO” if you wish to allow ***em*** to automatically calculate the analysis frequency. When AUTO is used, there is no `frequency` entry. The fcalc field is set to “MAN” if the analysis frequency has been entered by the user. This option is followed by the desired analysis frequency in KHz.

END End statement

Syntax `END FREQ`

Indicates the end of the frequency block. Required.

Control Block

The control block specifies the type of analysis sweep presently defined for the project. All the analysis controls which have been input to the project are available in the FREQ block. The control block identifies which sweep would be used in the actual analysis of the project.

The CONTROL block contains a statement which indicates the type of sweep presently selected in the Analysis Control drop list in the Analysis Setup dialog box. Only one type of sweep is entered in this block at any one time. Choices include SIMPLE, STD, ABS, RES_ABS, OPTIMIZE, VARSWP, and EXTFILE. See the keyword entry below for details.

CONTROL Beginning of control block

Syntax `CONTROL`

Indicates the beginning of the control block. All statements following this entry are included in the control block until you reach the END CONTROL statement. The statements in this block are generated in the Analysis Setup dialog box.

SIMPLE Simple Sweep

Syntax `SIMPLE`

This selects a simple sweep. The simple sweep values are entered under the keyword SIMPLE in the FREQ block.

STD Standard Sweep

Syntax `STD`

This selects the standard sweep. The standard sweep values are entered under the SWEEP keyword in the FREQ block. There may be multiple STD statements in the frequency block.

ABS Adaptive Band Synthesis (ABS)

Syntax This selects an Adaptive Band Synthesis. If a Manual resolution has been entered by the user, then it appears in the RES_ABS statement in this block.

RES_ABS ABS Resolution

Syntax `RES_ABS [N|Y] resolution`

This controls the ABS resolution. If the resolution is automatically calculated by em, then an N appears in the second field. If the manual resolution is selected in the Advanced Options dialog box, then Y appears in the second field. <resolution> is a floating point number which is the minimum gap between data points to be used in the ABS analysis. This value is input by the user.

OPTIMIZE Optimization

Syntax `OPTIMIZE`

This selects an optimization. The optimization is specified in the OPT block. For details on the OPT block, see “OPT,” page 89.

VARSWP Parameter Sweep

Syntax `VARSWP`

This selects a parameter sweep. The parameter sweep is specified in the VARSWP block. For details on the VARSWP block, see “VARSWP,” page 91.

EXTFILE External Frequency File

Syntax `EXTFILE`

This selects an external frequency file. The external frequency file is identified in the FILENAME statement which also appears in the control block. For details about the FILENAME statement, see “FILENAME,” page 86.

OPTIONS Analysis Control options

Syntax `OPTIONS [-<option identifiers>]`

Command line options for the analysis run. Options appear depending on which checkboxes are selected in the Analysis Setup dialog box and Advanced Options dialog box. Multiple analysis options can be combined. For example, if both Generate Current Density and De-embedding are selected then the entry would be

OPTIONS -dj. The table below shows all the options along with the checkboxes that generate them. In addition, any options entered in the Additional Options text entry box in the Advanced Options dialog box appear here. Note that run options are case sensitive.

Checkbox	Option
Generate Current Density	j
Multi-Frequency Caching	A
Memory Saver	m
Box Resonance	b
De-embedding	d
Q-factor Accuracy	

SUBSPLAM Subsections per Lambda

Syntax `SUBSPLAM use subslambda`

This statement provides the maximum subsections/lambda and is optional. If no SUBSPLAM entry appears, the default value of 20 is used. This value is entered in the Advanced Subsectioning dialog box. <use> indicates if the value is being used. The field is “Y” for yes and “N” for no. <subslambda> is an positive integer value which defines the maximum number of subsections/lamba. The minimum legal value is 6. In release 6.0a, this value was input with the Box parameters.

EDGECHECK Polygon Edge Checking

Syntax `EDGECHECK use numlevels`

This statement provides the number of levels for which polygon edge checking is performed. If no EDGECHECK entry appears, the default value of 1 is used. This value is entered in the Advanced Subsectioning dialog box. <use> indicates if the value is being used. The field is “Y” for yes and “N” for no. <numlevels> is a positive integer value which defines the number of levels for which polygon edge checking is performed.

CFMAX Maximum Subsectioning Frequency

Syntax `CFMAX use subfreq`

This statement provides the frequency used to calculate the maximum subsection size. If no CFMAX entry appears, the value is automatically calculated by ***em***. This value is entered in the Advanced Subsectioning dialog box. <use> indicates which frequency is being used. The field is “N” for the highest frequency in the present analysis only, “Y” for a fixed frequency which is entered in the <subfreq> field, “L” for the highest frequency in a previous analysis only, and “B” for the highest frequency in the present or a past analysis. <subfreq> is a floating point value which defines the frequency used to calculate the maximum subsection size when a Fixed Frequency is selected; this would be indicated by a “Y” in the use field. If the use field is another value then <subfreq> is not used. The units for the frequency are those stated in the DIM statement of the GEO block.

CEPSY Estimated Epsilon Effective

Syntax `CEPSY use epsilon`

This statement provides the estimated epsilon effective. If no CEPSY entry appears, the value is automatically calculated by ***em***. This value is entered in the Advanced Subsectioning dialog box. <use> indicates if the value is being used. The field is “Y” for yes and “N” for no. <epsilon> is a floating point value which defines the estimated epsilon effective. This value must be greater than zero and should be greater than one.

FILENAME External Frequency File

Syntax `FILENAME filename`

The filename statement provides the name of the external frequency file used to control the analysis. This file is only used if the EXTFILE statement also appears in the control block. The filename is a character string. This character string may include an absolute or relative path name.

SPEED Analysis Speed/Memory Control - Valid only for geometry projects.

Syntax `SPEED setting`

This entry is the position of the Speed/Memory slider in the Analysis/Speed Control dialog box. This entry is only valid for geometry projects. See the table below for values of setting.

Setting Value	Meaning
0	Fine/Edge Meshing (Left on Slider)
1	Coarse/Edge Meshing (Middle on Slider)
2	Coarse/No Edge Meshing (Right on Slider)

RES_ABS Target for Manual Frequency Resolution for ABS Sweep

RES_ABS [Y|N] number

This entry is the number of frequencies entered as the target for an ABS sweep when Manual is selected for the ABS frequency resolution in the Advanced Options dialog box. The number of frequencies is an integer value. For a new project in which the Manual option has not been selected, this entry does not appear. Once the Manual option has been selected for the Frequency Resolution, then the RES_ABS statement appears with a “Y” indicating that Manual is presently selected and a “N” indicating that Automatic is presently selected. If Manual is selected, the TARG_ABS statement is ignored.

CACHE_ABS ABS Caching Level

```
CACHE_ABS setting
```

This entry is the selected setting of the ABS caching level in the Advanced Options dialog box. See the table below for values of setting:

Setting Value	Meaning
0	None
1	Stop/Restart
2	Multi-sweep plus Stop/Restart

TARG_ABS Target for Automatic Frequency Resolution for ABS Sweep

```
TARG_ABS number
```

This entry is the number of frequencies entered as the target for an ABS sweep when Automatic is selected for the ABS frequency resolution in the Advanced Options dialog box. The number of frequencies is an integer value. For a new project, this is the only entry which appears for the ABS frequency resolution. If the Manual option has even been selected for the Frequency Resolution, then the RES_ABS statement also appears. In that case, this statement is only used, if the RES_ABS statement indicates that “Manual” is not selected. See “RES_ABS,” page 87 for details.

Q_ACC Q-Factor Accuracy

```
Q_ACC use
```

This statement determines if the Q-Factor Accuracy option in the Advanced Options dialog box is being used. <use> indicates if the value is being used. The field is “Y” for yes and “N” for no.

PUSH Hierarchy Sweep - Valid only for netlist projects.

Syntax `PUSH`

The presence of this keyword indicates that the Hierarchy sweep option has been selected in the Analysis Setup dialog box for a netlist project. A Hierarchy sweep imposes its frequency sweep on all subprojects during an analysis. This option is only available for a netlist project.

END End statement

Syntax

```
END CONTROL
```

Indicates the end of the control block. Required.

Optimization Block

The optimization block specifies the sweep, parameter data range and optimization goals for an optimization. This block is only used when the control block contains the OPTIMIZE entry. Any frequency sweep which may be defined for an ***em*** analysis may be used for an optimization. For details about these entries, see the "Frequency Block" on page 80.

OPT Beginning of optimization block

Syntax

```
OPT
```

Indicates the beginning of the optimization block. All statements following this entry are included in the optimization block until you reach the END OPT statement. The statements in this block are generated in the Analysis Setup dialog box.

MAX Maximum number of iterations.

Syntax

```
MAX num
```

The keyword MAX is followed by an integer number, <num>, which is used as the maximum number of iterations for an optimization.

VARS Optimization Parameter Settings

Syntax

```
VARS
varname(1) [N|Y] minval maxval granularity
varname(2) [N|Y] minval maxval granularity
.
.
varname(n) [N|Y] minval maxval granularity
```

The VARS command defines which parameters are to be used in an optimization and their data range. Each parameter in a project must have an entry line. The name of the parameter, <varname>, is followed by an N if it is not used in the optimization. If the parameter is used in the optimization this field is Y. The default is N. <minval> is the minimum value of the parameter and <maxval> is the maximum value. If values have not been entered both fields are UNDEF. The `granularity` field is set to UNDEF if Auto is entered. If a manual value has been entered for granularity this is a floating point number.

Analysis Sweeps

The following eight keywords are all the types of sweeps available with an optimization. There is some type of sweep statement associated with each optimization goal.

SWEEP Linear frequency sweep with stated interval. For details see the SWEEP keyword in the "Frequency Block" on page 80.

ESWEEP Exponential frequency sweep. For details see the ESWEEP keyword in the "Frequency Block" on page 80.

LSWEEP Linear frequency sweep with number of points. For details see the LSWEEP keyword in the "Frequency Block" on page 80.

STEP Discrete analysis frequencies. For details see the STEP keyword in the "Frequency Block" on page 80.

ABS Adaptive Band Synthesis Sweep. For details see the ABS keyword in the "Frequency Block" on page 80.

ABS_FMIN Find the minimum frequency response. For details see the ABS_FMIN keyword in the "Frequency Block" on page 80.

ABS_FMAX Find the maximum frequency response. For details see the ABS_FMAX keyword in the "Frequency Block" on page 80.

DC_FREQ Analyze at a DC frequency point. For details see the DC_FREQ keyword in the "Frequency Block" on page 80.

NET Optimization Goal

Syntax `NET=[GEO|netname] restype[respar] rel tartype tarvalue weight`

There is a net statement for each optimization goal entered in the Analysis Setup. If the project is a geometry project NET=GEO is used. NET=GEO also identifies the main network of a netlist. For a netlist project, <netname> is the name of the network whose response you wish to use. <restype> is the response type: choices are DB, ANG, MAG, RE and IM. <respar> is the response. <respar> consists of either S, Y, or Z followed by a pair of port indices. For example, S_{24} would be S24. If one of the two port indices has more than one digit, you use an underscore to separate the two port indices. For example, S10_27. <rel> is the relationship between the response and target: choices include "=", "<" or ">". <tartype> is the type of target: choices include VALUE, NET, and FILE.

If the <tartype> is VALUE then the <tarvalue> is a floating point number which specifies the desired target value.

If <tartype> is FILE it is followed by a pathname for the file using this syntax: "`<pathname>`". In this case, <tarvalue> is a response specified using the syntax of `restype[respar]` as described above.

If <tartype> is NET then tarvalue is specified using the "`NET[GEO|netname] restype[respar]`" specified above.

END End statement

Syntax `END OPT`

Indicates the end of the optimization block. Required.

Parameter Sweep Block

The parameter sweep block specifies a frequency sweep and data range for each parameter sweep. This block is only used when the control block contains the VARSWP entry.

VARSWP Beginning of parameter sweep block

Syntax

```
VARSWP
sweeptype(1) sweepparameters
```

```
parameter(1) [N|Ytype] min max step
parameter(2) [N|Ytype] min max step
.
.
sweeptype(n) sweepparameters
parameter(n) [N|sweeptype] min max step
```

Indicates the beginning of the parameter sweep block. All statements following this entry are included in the parameter sweep block until you reach the END VARSWP statement. The statements in this block are generated in the Analysis Setup dialog box.

The sweeptype entry defines a parameter sweep. There is a sweeptype entry for each parameter sweep defined in the Analysis Setup dialog box. Each sweeptype statement defines the frequency sweep. After each sweep statement there is a line for each parameter defined in the project.

The sweeptype entry can be any of the Frequency Control options followed by its parameters. The available entries are SWEEP, ESWEEP, LSWEEP, STEP, ABS_ENTRY, DC_FREQ. For details about each sweep type and its parameters, see "Frequency Block" on page 80.

The entry for each parameter starts with the name of the parameter. Then next field is N if the parameter is not used in the parameter sweep. In this case, the nominal value of the parameter is used for all analysis frequencies. If the next field is a `Ytype` value, then the parameter is used in the parameter sweep. The table below shows the possible values for Ytype and their definitions.

Ytype	Definition
Y	Linear sweep
YN	Linear sweep # steps
YC	Corner Sweep
YS	Sensitivity Sweep
YE	Exponential Sweep

<min> is a floating point number which is the minimum value of the parameter. <max> is a floating point number which defines the maximum value of the parameter. <step> is a floating point number which defines the interval between parameter values.

END End statement

Syntax `END VARSWP`
Indicates the end of the parameter sweep block. Required.

Output File Block

The output file block specifies an output file which allows you to store response data from your analysis outside your project file. After the initial FILEOUT statement in the beginning of the block, each line specifies an output file. These files are specified in the Output Files dialog box.

FILEOUT Beginning of output file block

Syntax `FILEOUT`
Indicates the beginning of the output file block. All statements following this entry are included in the output file block until you reach the END FILEOUT statement. The statements in this block are generated in the Generate Default Output Files dialog box. Each output file has its own entry. There are four different syntaxes for the output file; response data, PI Spice, N-coupled Line Spice and Broadband Spice. All are detailed below.

Response `filetype [NET=network] embed ABS_inc filename comments sig partype parform ports`

`filetype` The following options are available for this field.

filetype entry	Definition
TS	Touchstone
DATA_BANK	Databank
SC	SCompact
CSV	Spreadsheet
CADENCE	Cadence
MDIF	MDIF (S2P)
EBMDIF	MDIF (ebridge)

`network` The NET=network is omitted in a geometry project. For a netlist project, network is the network for which you wish to export data.

`embed` This field is "D" for de-embedded data or "ND" for non-de-embedded data.

`ABS_inc` This field is "Y" to include the ABS adaptive data or "N" to include only the discrete data. Default is to include the adaptive data.

`filename` The filename consists of a basename and extension. If the basename of the project file is used, the variable "$BASENAME" may be substituted in the filename. For example, in the project file steps.son if an output file steps.s2p is entered, the filename would appears as "$BASENAME.s2p" in the fileout block. The user may enter any filename they wish and are not restricted in their use of extensions.

`comments` This field is "NC" for no comments or "IC" to include comments.

`sig` This value is "15" if High precision is on and "8" if High Precision is not selected.

`partype` This field is "`S`" for S-Parameters, "`Y`" for Y-Parameters, and "`Z`" for Z-Parameters.

`parform` The following options are available for this field:

parform entry	Definition
MA	Mag-Angle
DB	DB-Angle
RI	Real-Imag

`ports` There are three different syntaxes for port information for output files.

If all ports in the circuit use real impedance with the same resistance and all other values 0, then <ports> is as follows:

```
R resist
```

where <resist> is a floating point number for the resistance.

If all ports in the circuit use complex impedance with the same resistance and all other values 0, then <ports> is as follows:

```
Z rresist iresist
```

where <rresist> is a floating point number for the real part of the resistance and <irresist> is a floating point number for the imaginary part of the resistance.

```
TERM resist(1) react (1) resist(2) react(2) ... resist(n) react(n)
```

where <`resist(1)`> is a floating point number for the resistance of the first port in the circuit, <`react(1)`> is a floating point number for the reactance of the first port in the circuit. Pairs of values, for resistance and reactance are repeated for each port in the circuit. If the number of ports is large, the continuation character (&) is used for additional lines that are part of this file's specification.

If a port or ports in the circuit have a non-zero value for either the inductance or capacitance, then each port displays four values using the following syntax:

```
FTERM resist(1) react(1) induct(1) cap(1) ... resist(n) react(n) induct(n) cap(n)
```

where <resist(1)> is a floating point number for the resistance of the first port in the circuit, <react(1)> is a floating point number for the reactance of the first port in the circuit, <induct(1)> is a floating point number for the inductance of the first port in the circuit, and <cap(1)> is a floating point number for the capacitance of the first port in the circuit. Four values, for resistance, reactance, inductance and capacitance are repeated for each port in the circuit. If the number of ports is large, the continuation character (&) is used for additional lines that are part of this file's specification.

This completes the syntax for the file specifications for response files.

PIMODEL This is the entry for an PI Model Spice file which is specified in the PI Model File Entry dialog box.

Syntax

```
PIMODEL [NET=network] embed ABS_inc filename comments sig
PINT=pint RMAX=rmax CMIN=cmin& LMAX=lmax KMIN=kmin RZERO=rzero
format
```

`network` The NET=network is omitted in a geometry project. For a netlist project, <network> is the network for which you wish to export data.

`embed` This field is "D" for de-embedded data or "ND" for non-de-embedded data.

`ABS_inc` This field is "Y" to include the ABS adaptive data or "N" to include only the discrete data. Default is to include the adaptive data.

`filename` The filename consists of a basename and extension. If the basename of the project file is used, the variable "$BASENAME" may be substituted in the filename. For example, in the project file steps.son if an output file steps.lib is entered, the filename would appears as "$BASENAME.lib" in the fileout block. The user may enter any filename they wish and are not restricted in their use of extensions.

`sig` This value is "15" if High precision is on and "8" if High Precision is not selected.

`pint` This is a floating point number for the percentage used to determine the intervals between the two frequencies used to determine each SPICE model.

`rmax` This is a floating point number for the maximum allowed resistance (Rmax). Default value is 1000.0

`cmin` This is a floating point number for the minimum allowed capacitance (Cmin). Default value is 0.1.

`lmax` This is a floating point number for the maximum allowed inductance (Lmax). Default value is 100.0.

`kmin` This is a floating point number for the minimum allowed mutual inductance (Kmin). Default value is 0.01.

`rzero` This is a floating point number for the resistor to go in series with all lossless inductors (Rzero). Default value is 0.0.

`format` This is the format for the PI Model output file. This field is "PSPICE" for PSpice and "SPECTRE" for Spectre.

NCLINE This is the entry for an N-coupled line Spice model file which is specified in the N-coupled Line Model File Entry dialog box.

Syntax `NCLINE [NET=network] embed ABS_inc filename comments sig format`

`network` The NET=network is omitted in a geometry project. For a netlist project, <network> is the network for which you wish to export data.

`embed` This field is "`D`" for de-embedded data or "`ND`" for non-de-embedded data.

`ABS_inc` This field is "Y" to include the ABS adaptive data or "N" to include only the discrete data. Default is to include the adaptive data.

`filename` The filename consists of a basename and extension. If the basename of the project file is used, the variable "`$BASENAME`" may be substituted in the filename. For example, in the project file steps.son if an output file steps.lib is entered, the filename would appears as "`$BASENAME.lib`" in the fileout block. The user may enter any filename they wish and are not restricted in their use of extensions.

`sig` This value is "15" if High precision is on and "8" if High Precision is not selected.

`format` Only one format is available for the n-coupled line model so that this field is always "GENERAL."

BBEXTRACT This is the entry for a Broadband Spice Model which is specified in the Broadband Spice Model File Entry dialog box. The entry specified above is followed by an OPTIONS block.

Syntax
```
BBEXTRACT [NET=network] embed ABS_inc filename comments sig
format
```

`network` The NET=network is omitted in a geometry project. For a netlist project, <network> is the network for which you wish to export data.

`embed` This field is "`D`" for de-embedded data or "`ND`" for non-de-embedded data.

`ABS_inc` This field is "Y" to include the ABS adaptive data or "N" to include only the discrete data. Default is to include the adaptive data.

`filename` The filename consists of a basename and extension. If the basename of the project file is used, the variable "`$BASENAME`" may be substituted in the filename. For example, in the project file steps.son if an output file steps.lib is entered, the filename would appears as "`$BASENAME.lib`" in the fileout block. The user may enter any filename they wish and are not restricted in their use of extensions.

`sig` This value is "15" if High precision is on and "8" if High Precision is not selected.

`format` This is the format for the PI Model output file. This field is "PSPICE" for PSpice and "SPECTRE" for Spectre.

OPTIONS Options block for Broadband Spice Model specification. These options are specified in the Broadband Model File Entry dialog box and the Advanced Broadband Model Options dialog box.

Syntax
```
OPTIONS
```

ErrThresh Error Threshold for Broadband Spice Model

Syntax
```
ErrThresh threshold
```

`threshold` This field is a floating point value used for the error threshold.

TotalOrder The maximum order of the rational polynomial which produces the Broadband Spice Model.

Syntax `TotalOrder order`

`order` This field is an integer value used as the maximum order.

CurveFit Generate Predicted S-parameter data file

Syntax `CurveFit Y|N`

This field indicates whether a Predicted S-parameter data file is generated when you create the Broadband Spice model. ‘Y” is for yes and “N” is for no.

DCPoint N Generate data at a DC Point for the Predicted S-parameter data

Syntax `DCPOINT Y|N`

This field indicates whether analysis data at a DC Point should be included in the predicted S-Parameter data. “Y” is for yes and “N” is for no.

SingleFile Generate a file for each parameter combination.

Syntax `SingleFile Y|N`

This field indicates whether a separate output file is created for each parameter combination or if a single output file contains the data for all parameter values. “Y” is for yes and “N” is for no.

FUNIT Frequency units

Syntax `FUNIT unit`

This field defines the frequency units used when requesting extra frequencies be output to the Predicted S-parameter data. “unit” is the frequency units being used. This unit is always the same as specified for the project file. Possible values are Hz, KHz, MHz, GHZ, THz,

FOLDER Identifies the output directory in which the output files are created.

Syntax `FOLDER pathname`

This field identifies the directory in which the output files are created. The pathname is relative to the project source directory.

END End statement

Syntax `END OPTIONS`

Indicates the end of the Options block for the Broadband Spice Model file. Required

END End statement

Syntax `END FILEOUT`
Indicates the end of the output file block. Required.

Parameter Block for Netlist Project

The parameter block specifies parameters for a netlist project. After the initial VAR statement in the beginning of the block, each line specifies a parameter and its nominal value. The end of the VAR block is indicated by the END VAR statement.

VAR Parameters for a Netlist Project

Syntax
```
VAR
parname = nomvalue
```
Indicates the beginning of the parameter block in a netlist project. There is an entry line for each parameter defined in a netlist project. The END VAR statement indicates the end of the parameter block.

`parname` A character string which provides the name of the parameter.

`nomvalue` A floating point number which is the nominal value for the parameter.

END End statement

Syntax `END VAR`
Indicates the end of the VAR block. Required.

Circuit Block for Netlist Project

The circuit block specifies all the elements and networks in a netlist project. These entries correspond to the netlist which appears in the netlist editor window. This is equivalent to the GEO block of a geometry project.

CKT Circuit elements and netlist

Syntax

```
CKT
```

Indicates the beginning of the circuit block in a netlist project. The END CKT statement indicates the end of the parameter block.

RES Resistor Element

Syntax

```
RES nodenum1 [nodenum2] R = resvalue
```

The RES statement defines a resistor element in your netlist circuit. The <nodenum1> is the first, and possibly only, node number in the net to which the resistor is connected. <nodenum2> only appears if the resistor is connected between two nodes in the network, and is the second node to which the resistor is connected. The resistor is included in the first network whose entry (DEFnP) appears after the RES statement in the file. <resvalue> is the value of the resistor and can either be a floating point number or a character string representing a parameter defined in the netlist project.

IND Inductor Element

Syntax

```
IND nodenum [nodenum2] L = indvalue
```

The IND statement defines an inductor element in your netlist circuit. The <nodenum1> is the first, and possibly only, node number in the net to which the inductor is connected. <nodenum2> only appears if the inductor is connected between two nodes in the network, and is the second node to which the inductor is connected. The inductor is included in the first network whose entry (DEFnP) appears after the IND statement in the file. <indvalue> is the value of the inductor and can either be a floating point number or a character string representing a parameter defined in the netlist project.

CAP Capacitor Element

Syntax

```
CAP nodenum [nodenum2] C = capvalue
```

The CAP statement defines a capacitor element in your netlist circuit. The <nodenum1> is the first, and possibly only, node number in the net to which the capacitor is connected. <nodenum2> only appears if the capacitor is connected between two nodes in the network, and is the second node to which the capacitor is connected.The capacitor is included in the first network whose entry (DEFnP) appears after the CAP statement in the file. <capvalue> is the value of the capacitor and can either be a floating point number or a character string representing a parameter defined in the netlist project.

TLIN Transmission Line element

Syntax

```
TLIN prt1nd prt2nd Z=imped E=length F=freq
```

The TLIN statement defines a transmission line element in your netlist circuit. The transmission line is included in the first network whose entry (DEFnP) appears after the TLIN statement in the file. <prt1nd> is an integer number; port 1 of the transmission line is connected to this node in the netlist. <prt2nd> is an integer number; port 2 of the transmission line is connected to this node in the netlist. <imped> is a floating point number used for the impedance of the transmission line. <length> is a floating point number used for the electrical length in degrees of the transmission line. <freq> is a floating point number which is the frequency for the transmission line.

TLINP Physical Transmission Line element

Syntax

```
TLINP prt1nd prt2nd Z=imped L=length K=eeff F=freq A=atten
```

The TLINP statement defines a physical transmission line element in your netlist circuit. The physical transmission line is included in the first network whose entry (DEFnP) appears after the TLINP statement in the file. <prt1nd> is an integer number; port 1 of the transmission line is connected to this node in the netlist. <prt2nd> is an integer number; port 2 of the transmission line is connected to this node in the netlist. <imped> is a floating point number used for the impedance of the transmission line. <length> is a floating point number used for the length of the transmission line. <freq> is a floating point number which is the frequency for the transmission line. <atten> is a floating point number which is the attentuation (dB/length unit) of the physical transmission line.

SnP Data Response file element

Syntax

```
SnP prtnode(1) prtnode(2) .. prtnode(n) [gndnode] filename
```

The SnP entry defines a data response file element in your netlist circuit. The response file is included in the first network whose entry (DEFnP) appears after the SnP statement. The keyword changes depending upon the number of ports in the data file. A response file for a 2 port circuit would use the keyword S2P. A response file for a 4 port circuit would use the keyword S4P. A node number, <prtnode>, appears for each port in the data file. If there are four ports in a circuit, then there will be four integers representing the nodes to which the ports are connected. <gndnode> is optional. If the ground for the response file is connected to GND than this field is omitted. However, if ground for the response file is connected to a node, than that node appears here. Last in the statement is a character string, <filename>, identifying the response file you wish to include. This string may include an absolute or relative path.

PRJ Project File Element

Syntax

```
PRJ prt1node .. prt(n)node filename numprt swpcon [parameter = parvalue]
```

The PRJ entry defines a subproject element in your netlist circuit. The subproject is included in the first network whose entry (DEFnP) appears after the PRJ statement. A node number, <prt(n)node>, appears for each port in the project. If there are four ports in a circuit, then there will be four integers representing the nodes to which the ports are connected. <gndnode> is optional. If the ground for the project is connected to GND than this field is committed. However, if ground for the project is connected to a node, than that node appears here. Next in the statement is a character string, <filename>, identifying the subproject you wish to include. This string may include an absolute or relative path. <numprt> is an integer value for the number of ports in the project. <swpcon> is 0 to indicate that you use the sweep from this project or 1 to indicate that you use the sweep from the subproject. This setting is overridden if Hierarchy Sweep is on. If the subproject contains any parameters, then there is a `<parameter>=<parvalue>` entry for each parameter. <parameter> is a character string which is the name of the parameter. <parvalue> is a floating point number or a character string representing a parameter defined in the netlist project.

DEFnP Network or Network Element

Syntax `DEFnP prtnode(1) prtnode(2) .. prtnode(n) netname ports`

The DEFnP entry defines a network element or main network in your netlist circuit. The main network is identified by the last DEFnP statement which occurs in the project file. The keyword changes depending upon the number of ports in the network. A network for a 2 port circuit would use the keyword DEF2P. A network for a 4 port circuit would use the keyword DEF4P. A node number, <prtnode>, appears for each port in the network. If there are four ports in a network, then there will be four integers representing the nodes to which the ports are connected. Last in the statement is a character string, <netname>, identifying the network. In the case of a network element, this string may include an absolute or relative path. <ports> defines the port terminations for the network. The syntax is detailed below.

`ports` There are three different syntaxes for port information for output files.

If all ports in the circuit use real impedance with the same resistance and all other values 0, then <ports> is as follows:

```
R resist
```

where <resist> is a floating point number for the resistance.

If all ports in the circuit use complex impedance with the same resistance and all other values 0, then <ports> is as follows:

```
Z rresist iresist
```

where <rresist> is a floating point number for the real part of the resistance and <irresist> is a floating point number for the imaginary part of the resistance.

If a port or ports in the circuit have a non-zero value for the reactance, then each port displays two values using the following syntax:

```
TERM resist(1) react (1) resist(2) react(2) ... resist(n) react(n)
```

where <resist(1)> is a floating point number for the resistance of the first port in the circuit, <react(1)> is a floating point number for the reactance of the first port in the circuit. Pairs of values, for resistance and reactance are repeated for each port in the circuit. If the number of ports is large, the continuation character (&) is used for additional lines that are part of this file's specification.

If a port or ports in the circuit have a non-zero value for either the inductance or capacitance, then each port displays four values using the following syntax:

```
FTERM resist(1) react(1) induct(1) cap(1) ... resist(n) react(n) induct(n) cap(n)
```

where <resist(1)> is a floating point number for the resistance of the first port in the circuit, <react(1)> is a floating point number for the reactance of the first port in the circuit, <induct(1)> is a floating point number for the inductance of the first port in the circuit, and <cap(1)> is a floating point number for the capacitance of the first port in the circuit. Four values, for resistance, reactance, inductance and capacitance are repeated for each port in the circuit. If the number of ports is large, the continuation character (&) is used for additional lines that are part of this file's specification.

This completes the syntax for the file specifications for response files.

END End statement

Syntax `END CKT`

Indicates the end of the circuit block. Required.

Subdivider Block for Geometry Project

The subdivider block specifies the subdividers and resulting filenames for a geometry project. After the initial SUBDIV statement in the beginning of the block, entries specify the output file names and subdivider's positions. The end of the subdivider block is indicated by the END SUBDIV statement.

MAIN Main netlist name

Syntax `MAIN filename`

This entry defines the name for the resulting netlist file produced from performing the subdivide on this geometry project. This is the name entered in the Circuit Subdivision dialog box which appears when you select the *Tools ⇒ Subdivide*

Circuit command. The filename is a character string which identifies the main netlist. This string may include an absolute or relative path. The default for this filename is $Basename_net.son.

REFPLANE Reference Planes for Subprojects

Syntax REFPLANE reftype reflength

This statement defines the reference planes added to the geometry subprojects which result from performing the subdivide. This selection is made in the Subproject Specification dialog box. The reftype is the type of reference plane you wish to add to the geometry subprojects. The options and meanings are shown in table below.

reftype	Meaning
A	Automatic - Suggested Length used
F	Fixed - Entered length used
N	None - No reference planes added

<reflength> is a floating point number which is the length of the automatically added reference planes. If the reftype is set to "N" then reflength is equal to zero.

NAME Geometry Subproject Name

Syntax `NAME secnum prjname`

The NAME statement defines the name for a geometry subproject created when a subdivide is performed. There is a name statement for each geometry subproject created. A geometry subproject is created for each section of the geometry produced when the subdividers are added. These names are entered in the Subproject Specification dialog box.

<secnum> is the integer value for the section number from which the geometry subproject is produced. secnum always starts at a value of 1 and increments for each NAME statement. The <prjname> is a character string which contains the name for the resultant geometry subproject. This string may include an absolute

or relative path. The default is $BASENAME_net_s(secnum).son. For example, for section 3 of the file spiral.geo, the <prjname> would be $BASENAME_net_s3.son.

LINE Subdivider Line Location

Syntax `LINE linenum coordinate dir`

The LINE statement defines the location of a subdivider in the project. There is a LINE statement for each subdivider in a geometry project. The <linenum> is an integer number which identifies the subdivider. <linenum> starts at a value of 1 for each geometry file and increases sequentially for each subdivider. The <coordinate> is a floating point number which indicates the position of the subdivider on the substrate. If the subdivider is vertical, indicated by "`V`" in the dir field, then the subdivider occurs at this distance from the origin in the x (horizontal) direction. If the subdivider is horizontal, indicated by "`H`" in the <dir> field, then the subdivider occurs at this distance from the origin in the y (vertical) direction. The direction for all subdividers in a circuit must be the same. The origin (0,0) is located at the upper left hand corner of the substrate.

Quick Start Guide Block for a Geometry Project

The QSG block is used to specify the settings in the Quick Start Guide. The following seven lines specify which tasks in the Quick Start Guide have been done by the user. Each line corresponds to an entry in the Quick Start Guide. If the task is done a YES appears after the keyword. If the task has NOT been done, a NO appears after the keyword.

QSG Indicates the beginning of the Quick Start Guide block.

Syntax `QSG`

This statement is followed by statements detailed below which provide the state of entries in the Quick Start Guide. All the entries in this block are required.

IMPORT Specifies if a DXF or GDS import has been performed in the project editor.

Syntax `IMPORT [Y|N]`

A "Y" appears if an import has been done. A "N" appears if no import has been down.

EXTRA_METAL Specifies if extra metal has been removed from the circuit.

Syntax `EXTRA_METAL [Y|N]`

UNITS Specifies if the user has changed the units used in the project.

Syntax `UNITS [Y|N]`

A “Y” appears if the units have been changed. A “N” appears if the units have not been changed.

ALIGN Specifies if the user has aligned the circuit to the grid.

Syntax `ALIGN [Y|N]`

A “Y” appears if the circuit has been aligned to the gird. A “N” appears if the circuit has not been aligned.

REF Specifies if reference planes have been added to the circuit.

Syntax `REF [Y|N]`

A “Y” appears reference planes have been added to the circuit. A “N” appears if no reference planes have been added.

VIEW_RES Specifies if the user had viewed response data.

Syntax `VIEW_RES [Y|N]`

A “Y” appears if the user has viewed response using the response viewer, the current density viewer or the far field viewer. A “N” appears if the user has not viewed response.

METALS Specifies if the user has defined any new metal types.

Syntax `METALS [Y|N]`

A “Y” appears if the user has defined any new metal types. A “N” appears if no metal types have been defined.

METALS Specifies if the user has defined any new metal types.

Syntax `METALS [Y|N]`

A “Y” appears if the user has defined any new metal types. A “N” appears if no metal types have been defined.

USED Indicates if the Quick Start Guide is enabled for this project.

Syntax `USED [Y|N]`

A “Y” appears if the Quick Start Guide is being displayed. A “N” appears if the Quick Start Guide is not being displayed.

END QSG Indicates the end of the Quick Start Guide Block.

Syntax `END QSG`

This indicates the end of the Quick Start Guide block.

Component Data Files Block

The Component Data files block is used to specify the response data files associated with data file type components (SPARAM). Each line corresponds to data file and assigns an index number to the file that is used in the component definition specified in the SMD keyword. For more details about the SMD keyword, please see "SMD" on page 76.

SMDFILES Indicates the beginning of the Component Data Files block

Syntax `SMDFILES`

This statement is followed by an entry for each data file used by a component.

Data File Specifies a response data file used by a component and assigns an index number.

Syntax `<indexnumber> “<filename>”`

<indexnumber> is a non-zero positive integer. If the data file is used by more than one component, only one index number is assigned to that file. The same index number is used in the different component “SMD” entries. <filename> specifies the location of the response data file. Relative paths are used.

END SMDFILES Indicates the end of the Component Data Files Block.

Syntax

```
END SMDFILES
```

This indicates the end of the Component Data Files block.