Apache Design Solutions, Inc.

PCB/Package Data Exchange File (**XFL**) Format 2.0

(Last Revised 3/21/2011)

DRAFT COPY

NOT FOR REDISTRIBUTION

Copyright Notice and Proprietary Information

Disclaimer

Apache Design Solutions, Inc. makes no warranty of any kind, expressed or implied, with respect to software or documentation, its quality, or performance. The information in this document is subject to change without notice and does not represent a commitment on the part of Apache Design Solutions, Inc.

Trademarks

Apache Design Solutions, Inc.

PCB/Package Data Exchange File (**XFL**) Format

(Last Revised 3/10/2011)

1. Reserved Characters

#	Lines that begin with a pound (#) sign are treated as comments and ignored.
“ ”	any data between two double quotation marks is considered as a string.

2. Structure

The XFL consists of several sections and each section start with a dot (**.**) followed by a key word. The **.version**, **.unit**, and **.scale** sections consist of one line each. All other sections end with **.end** followed by the same key word. The key word in the **.end** line may be omitted. Except the .version, any section may be omitted.

.version x y

.unit [**inch** | **mm**]

.scale value

.material
definition
.end [material]

.material_frequency
definition
.end [material_frequency]

.layer
definition
.end [**layer**]

.shape
definition
.end [**shape**]

.board_geom
definition
.end [**board_geom**]

.padstack
definition
.end [**padstack**]

.part
definition
.end [**part**]

.component
definition
.end [**component**]

.netattr
definition
.end [**netattr**]

.netlist
definition
.end [**netlist**]

.via
definition
.end [**via**]

.bondwire
definition

.end [**bondwire**]

.route
definition
.end [route]

.ball
definition
.end [ball]

.bump
definition
.end [bump]

.analysisSettings
definition
.end [analysisSettings]

.spice
definition
.end [spice]

.pingroups
definition
.end [pingroups]

.cktPorts
definition
.end [cktPorts]

.SSOChannel
definition
.end [SSOChannel]

.gui
.net_color
definition
.end [net_color]
.end [gui]

3. Description

. version x y

Give the version of the XFL file.

x is the major version number, and y is a minor version number.

.version must be the first section in the XFL file.

x integer
y integer

.unit value

A geometric unit used throughout the XFL file. The value can be either **inch** or **mm**. If this section or the value is missing, then **inch** is assumed as the default.

value string (**inch** or **mm)**

.scale value

A geometric scale used throughout the XFL file. Actual dimension of the data in the file is determined by dividing the number by the scale value. For example, if the unit is set to **inch** and the scale is 1000, the geometric data represents mil or 1/1000 of inch. If the unit is set to mm and the scale is 1000, it is micron or micrometer. Default is 1.

value double

.material
definition
.end [**material**]

This is an optional section. Material properties. List materials used in the design and their properties. Each material is defined as:

C material name conductivity
or
D material name permittivity permeability loss tangent [conductivity causality]

where

C or **D**	**C** for conducting material, **D** for dielectric material
material name	Name of the material enclosed by double quotation marks.
conductivity	Electric conductivity (1/ohm.mm).
permittivity	Relative permittivity or dielectric constant
permeability	Relative permeability
loss tangent	Dielectric loss tangent
causality	Type of the causality. An integer number defines the causality type. 0 none 1 wideband debye 2 multipole debye

.material_frequency
definition
.end [material_frequency]

This is an optional section. Lists frequencies of used materials in the design. Each frequency dependent material is defined as:

C material name frequency conductivity
Or
D material name permittivity permeability loss tangent frequency conductivity

The definitions of material name, permittivity, permeability, loss tangent, conductivity are the same as the material's.

where,

frequency material frequency value in Hz.

.layer
definition
.end [**layer**]

Layer stack up. Describes a layer stack up from top to bottom (or from front to back). Each layer is defined as:

name thickness type conducting dielectric1

where

name — Name of the layer enclosed by double quotation marks. If unknown, write "".

thickness — Thickness of the layer. If unknown, write 0.

type — Type of the layer. A single character defines the layer type.
S for signal layer
D for dielectric layer
P for power or ground layer if it can be differentiated from the signal layer

conducting — Conducting material name entered as a string enclosed by double quotation marks.

dielectric1 — Dielectric material name entered as a string enclosed by double quotation marks.

Note: *Throughout the XFL, signal layer numbers are the numbers that are numbered sequentially from the top by counting only the signal/power/ground layers.*

.shape

definition

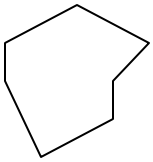
.end [**shape**]

Shape definition. This section defines shapes that can be referenced from other sections of the file. Supported shapes are polygon, rectangle, square, diamond, circle, annular, oblong, finger, bullet, and composite.

Each shape is defined as :

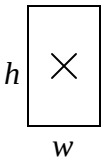
id keyword parameters

where id is an unique integer number starting from 1, and numbered sequentially. The keyword and parameters are described below.



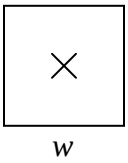
id **polygon {***x1 y1 x2 y2 …***}**

Vertices are enclosed by **{ }** and data can be written in more than one line. The last point does not need to be the same as the first point. Reference point is at (0,0).



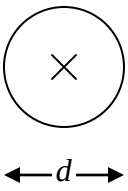
id **rectangle** *width height*

Each rectangle is defined by width and length and the definition must appear in one line. Reference point is at the center of the rectangle.



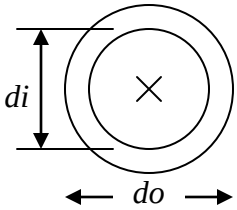
id **square** *width*

Each square is defined by width and the definition must appear in one line. Reference point is at the center of the square.



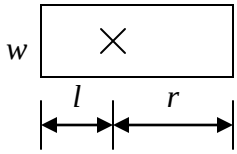
id **circle** *diameter*

Each circle is defined by radius and the definition must appear in one line. Reference point is at the center of the circle.



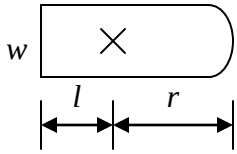
id **annular** *outer diameter* *inner diameter*

Each donut is defined by outer diameter and inner diameter and the definition must appear in one line. Reference point is at the center of the annular.



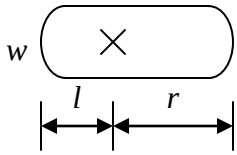
id **oblong** width left right

Each oblong is defined by width, left, and right. The definition must appear in one line. Reference point is shown at the left marked as X.



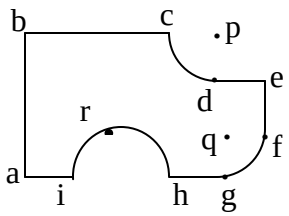
id **bullet** width left right

Each bullet is defined by width, left, and right. The definition must appear in one line. Reference point is shown at the left marked as X.



id **finger** width left right

Each finger is defined by width, left, and right. The definition must appear in one line. Reference point is shown at the left marked as X.



id **composite {***1st point segment1 segment2 ...* **}**

Composite shape consists of lines and/or arcs. The definition is enclosed by **{ }** and it can be shown in more than one lines. There are four types of segments – straight line, clockwise arc (**arc**), counter-clockwise arc (**rarc** or reverse arc) and arc defined by three points (**arc3**).

No keyword between two points indicates that they are connected by a straight line. A keyword **arc, rarc** or **arc3** between two points indicates that they are connected by a clockwise arc, a counter-clockwise arc or a three-point method arc, respectively. For **arc** and **rarc**, an arc origin appears after the arc ending point. For **arc3**, a middle point appears after the arc ending point.

The composite shape at the left can be described as:

2 **composite {**x_a y_a x_b y_b x_c y_c **rarc** x_d y_d x_p y_p x_e y_e x_f y_f **arc** x_g y_g x_q y_q x_h y_h **arc3** x_i y_i x_r y_r**}**

where the number 2 at the beginning is a shape id. Straight lines a-b and b-c, then a counter-clockwise arc from c to d with an origin at p. Straight lines d-e and e-f, then a clockwise arc from f to g with an origin at q. A straight line g-h and a three-point method arc from h to i with a middle point at r. Finally straight line from i to a. i to a is not defined but a straight line will connect the last point to the first point by default.

The same shape can also be described as:

2 **composite {**x_a y_a x_i y_i **arc3** x_h y_h x_r y_r x_g y_g **rarc** x_f y_f x_q y_q x_e y_e x_d y_d **arc** x_c y_c x_p y_p x_b y_b**}**

.board_geom
definition
.end [**board_geom**]

Exterior geometry of board or package is defined by one of the four methods shown below.

polygon {*x1 y1 x2 y2 …***}**
composite {*1st point segment1 segment2 …* **}**
shape *shapeID x y rotation mirror*
shape *shapeID x y mirror rotation*

The method of defining **polygon** and **composite** are the same as described in the shape section.
The **shape** is placed by defining shape ID, x and y coordinates (global coordinates) of the shape origin (local origin), counter-clockwise rotation in degrees, and mirror indicator. Mirror indicators are:

X	mirror about X-axis
Y	mirror about Y-axis
N	no mirror

If the mirror indicator appears after the rotation, the mirror is performed after the rotation. If the mirror indicator appears before the rotation, the mirror is performed before the rotation.

Both mirror and rotation operations are performed with respect to the shape origin (local origin).

Holes (cutouts or voids) are defined by using one or more of the following methods.

void_polygon {*x1 y1 x2 y2 …***}**
void_rectangle *width length x y*
void_square *width x y*
void_diamond *width x y*
void_circle *diameter x y*
void_composite {*1st point segment1 segment2 …* **}**
void_shape *shapeID x y rotation mirror*
void_shape *shapeID x y mirror rotation*

void_polygon, **void_composite**, and **void_shape** are defined the same way as **polygon**, **composite**, and **shape**, respectively.

void_rectangle, **void_square**, **void_diamond**, and **void_circle** are defined the same way as **rectangle**, **square**, **diamond**, and **circle** are defined in the shape section. They are placed at the coordinates x and y.

.padstack
definition
.end [**padstack**]

This section defines pad stacks and each pad stack is defined as:

pad stack id **{**
pad1
pad2
….
}

where the pad stack id is a number that will be referenced by vias in the via section and it is sequentially numbered from 1.

Each of the pads are defined only by using the predefined shapes in the shape section.

signal layer # shape id shape rotation apshape id ap shape rot

where

signal layer # Signal layer number as appeared in the layer section.

shape id Shape ID.

shape rotation Counter-clockwise rotation angle of the shape in degrees.

ap shape id Anti-pad shape ID. Anti-pad definition is optional.

ap shape rot Counter-clockwise rotation angle of the anti-shape in degrees.

.part
definition
.end [**part**]

Describes a part. Each part is defined as:

part name [shape llx lly urx ury height [type value [**noflip**] [material]]] **{**pin1 pin2 ….**}**

where part name is a name of the part. shape, llx, lly, urx, ury, height and **noflip** are optional. If the keyword **noflip** appears, part will not be flipped when it is place below the layer.

shape	Top view shape of the part. **R** for rectangle **C** for circle **D** for round-corner diamond
llx, lly	Lower-left coordinates of bounding box
urx, ury	Upper-right coordinates of bounding box
height	Height
type	Part type **R** for resistor **L** for inductor **C** for capacitor **S** for solder balls **D** for die **M** for molding compound **O** for other types
value	Part value (mOhm for Resister, nH for Inductor, pF for Capacitor). Set to zero for all other types. Part values R, L and C if the three values are in parentheses ( ). For example, (20.0 5 3.5)
material	Optional. Material name enclosed by " ".

Pins are defined as:

pin name x y i/o type [pad_stack_id] [relative layer]

where

pin name	Name of the pin. If the name is not known, a sequential number will be used as a name.

x y	Location of the pin with respect to local origin
i/o type	Pin I/O type. **D** for driver pin **R** for receiver pin **B** for bi-directional pin **DT** for driver terminator **RT** for receiver terminator
pad_stack_id	Optional. Pad stack ID (0 if pad stack is unknown)
relative layer	Optional. Default value is 0. Individual component pin layer is derived as pin_layer = component_layer + pin_relative_layer.

.component
definition
.end [**component**]

Describes a component placement. One component placement appears per line and each placed component is defined as:

U-name part name x y layer rotation [stack comp] [(R L C 0)] (combine value extraction comp type 0 0)

where

U-name	U-name. Also known as location identifier or reference designator. No space is allowed in the name.
part name	Part name. The name is enclosed by double quotation marks (**“ ”**).
x y	Location of the component origin with respect to the board origin.
layer	Placement layer number. ***+n*** for above the layer ***-n*** for below the layer
rotation	Counter-clockwise rotation of the component in degrees with respect to the component origin (local origin).

stack comp — Optional. Name of the component on which this component is stacked. If this defined, the layer number is ignored.

R L C 0 — Optional. Component's R, L, C and other values in a pair of parentheses. For example, (50 10 250 0). The units are mOhm, nH, or pF. The 4th value is for future use.

combine value — Combine three values on the three binary bits .
The value
= NonMatch4PinGroup*4
+**RefComp4PinGroup***2
+**Analysis**.

NonMatch4PinGroup — If the component does not match the pin group rule of the part, set the value as 1, else set 0.

RefComp4PinGroup — If the component is the part reference component for pin group, set value as 1, else set 0.

Analysis — if the component is to be analyzed, set value as 1, else set 0.

extraction — if the component is to be extracted for PSI setting, set value as 1, else set 0.

comp type — Component analysis type for PSI.
0 COMP_TYPE_LUMPED
1 COMP_TYPE_GAP

0 0 — 0 0 is dummy for distinguish from (R L C 0) field.

.netattr
definition
.end [**netattr**]

Net attribute definition. This section defines net attributes such as net class, allowable delays, and other user properties.

Each attribute group is defined as:

id **{**attribute name1=value1 attribute name2=value2 ….**}**

where the id is an attribute group number that will be referenced by nets in the netlist section and it is sequentially numbered from 1. There may be more than one attribute defined for one group. Each attribute group definition may appear in more than one line. Attribute assignments appear in the braces **{ }**. Each attribute assignment consists of attribute name, equal(=) sign, and the value.

.netlist
definition
.end [**netlist**]

This section defines net lists and each net is defined as:

net name net ype attribute id analysis npe type anlandbch**{**node1 node2 ….**}**

where

net name	Net name.
net type	Net type. **S** for signal net (**s** if broken) **P** for power net (**p** if broken) **G** for ground net (**g** if broken)
attribute id	Net attribute ID. Zero indicates that the attribute is unknown.
analysis	If the net is to be analyzed, set the value as 1, else set 0.
npe type	NPE net type. -1 NET_POWER 0 NET_GROUND 1 NET_SIGNAL.
anlandbch	The decade is the **NPEBranch** value, the single digit is the abs value of **NPEAnalysis**, the sign of the anlandbch is same as the sign of **NPEAnalysis**.
NPEAnalysis	NPE analysis type. 1 NET_ANALYZE

2	NET_ANALYZE_NO_CPL
-1	NET_NOT_ANALYZE
-2	NET_NOT_ANALYZE_NO_CPL

NPEBranch — Default value is 0. It keeps the net G(mho) value.

Each node is defined as:

U-name pin number i/o type NPE group num NPE node usage[{x y layer}]

where

U-name — U-name. "-" (minus sign) indicates unknown U-name.

pin number — Pin number or pin name.

i/o type — Pin I/O type. Available types are **D**, **R**, **B**, **DT**, and **RT** as described in the part section. A place holder "**-**" (minus sign) must appears if the type defined for the pin in the part section is to be used.

NPE group num — Default value is -5. Node group number, used for NPE multi-port, and will affect the .gp2 file if the value > 0.

NPE node usage — Node usage type. It will affect the .nod file, if the value > 2.

1	NODE_SOURCE
0	NODE_FLOAT
-1	NODE_SOURCE

Pin location may appear at the end of the node definition. It is optional and must be enclosed by { }. Set the layer number to negative for solder balls.

.via
definition
.end [**via**]

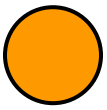
This section defines vias. One via definition appears per line and each via is defined as:

via name pad stack id pad stack rot shape id shape rot [thickness] [material]

where

via name	Via name that will be referenced by routing in the route section.
pad stack id	Pad stack ID. Zero indicates that no pad stack exists for the via.
pad stack rot	Counter-clockwise rotation angle of the pad stack
shape id	Shape ID of the via barrel.
shape rot	Counter-clockwise rotation angle of the via barrel shape.
thickness	Optional. Via barrel wall thickness "t".
material	Optional. Name of the material enclosed by double quotation marks.

t > 0



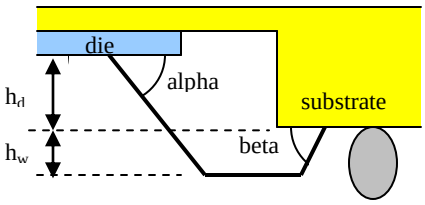
t = 0

.bondwire
definition
.end [**bondwire**]

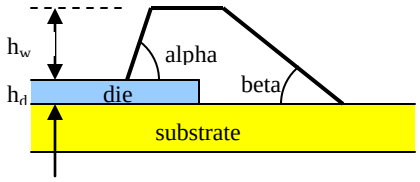
This section defines bond wire geometry. One bond wire definition appears per line and each bond wire is defined as:

id type material diameter h_w h_d alpha beta *[profile_name]*

where the id is a number that will be referenced by routing in the route section and it is sequentially numbered from 1. Others are explained below.



Die-down Configuration



Die-up Configuration

type	Bond wire type. **D** for die-down configuration **U** for die-up configuration

material	Material name or electric conductivity (1/ohm.mm). If it is a name, it must be enclosed by double quotation marks (**“ ”**).
diameter	Wire diameter.
h_w	Wire loop height.
h_d	Die height. $H_{die\_pad} - H_{top\_of\_top\_metal\_layer}$ for die-up $H_{die\_pad} - H_{bottom\_of\_bottom\_metal\_layer}$ for die-down
alpha	Die side angle in degrees.
beta	Substrate side angle in degrees.
profile_name	Name of the bond wire profile associated with the id.

.route
definition
.end [**route**]

This section defines routed nets and each routed net is defined as:

net name **{**segment1 segment2 ….**}**

where net name is one of the nets appeared in the netlist section. Each routed net consists of one or more of segments and each segment is defined by one of the following methods. All of them are in the form of

segment type signal layer number(s) segment definition

path *layer width* **{***1st point segment1 segment2 …* **}**
via *begin layer end layer via name x y rotation [mirror]*
bondwire *begin lyr end lyr bondwire id* x_b y_b x_e y_e *[die1 die2]*

polygon *layer* **{***x1 y1 x2 y2 …***}**
rectangle *layer width length x y*
square *layer width x y*
diamond *layer width x y*
circle *layer diameter x y*
annular *layer outer diameter inner diameter x y*
composite *layer* **{***1st point segment1 segment2 …* **}**

shape *layer shapeID x y rotation mirror*
shape *layer shapeID x y mirror rotation*

void_polygon *layer* **{***x1 y1 x2 y2 …***}**
void_rectangle *layer width length x y*
void_square *layer width x y*
void_diamond *layer width x y*
void_circle *layer diameter x y*
void_composite *layer* **{***1st point segment1 segment2 …* **}**
void_shape *layer shapeID x y rotation mirror*
void_shape *layer shapeID x y mirror rotation*

The **path** segment is defined the same way as the composite shape is defined in the shape section except it has a *width* and the last point does not automatically connect to the first point.

The **via** segment requires *begin layer* and *end layer* numbers while other segments requires only one layer number. The via segment also requires

via name	Name of the via defined in the via section
x y	Location of the via
rotation	Counter-clockwise rotation angle of the pad stack in degrees
mirror	Optional. Padstack mirror flag. **Y** mirror padstack **N** do not mirror padstack

The **bondwire** segment requires

begin lyr	Signal layer number of a beginning point. In order to indicate a die, enter a negative wire group ID.
end lyr	Signal layer number of an ending point. In order to indicate a die, enter a negative wire group ID.
bondwire id	Bond wire ID. (0 if unknown)
x_b y_b	Beginning point coordinate
x_e y_e	Ending point coodinate
die1 die2	Optional die component names. The name(s) appear here only if the *begin_lyr* and/or *end_lyr* are negative.

All other segment types are defined the same way as they are defined in the shape section or in the board_geom section.

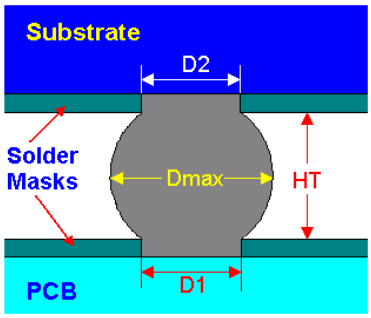
.ball
default ball
ball1
ball2
...
.end [ball]

This section is optional, and it defines ball info as:

default ball:
Dmax D1 D2 HT direction material

ball:
comp DMax D1 D2 HT direction material

Where



Dmax	The max diameter of ball.
D1	The bottom circle diameter.
D2	The top circle diameter.
HT	The height of ball.
direction	Ball direction 0 up 1 down
material	Name of the material, enclosed by double quotation marks.
comp	Name of component.

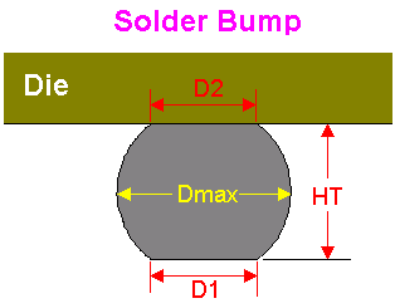
Note: *Currently,only the default ball size is used as all solder balls must be the same size.*

.bump
default bump
bump1
bump2
…
.end [bump]

This section is optional, and it defines bump info as:

default bump:
Dmax D1 D2 HT direction material

bump:
comp DMax D1 D2 HT direction material



Dmax	The max diameter of bump.
D1	The diameter of bottom circle.
D2	The diameter of top circle.
HT	The height of bump.
direction	Bump direction 0 up 1 down
material	Name of the material, enclosed by double quotation marks.
comp	Name of component.

.analysisSettings
definition
.end [analysisSettings]

This section is optional. It defines analysisSettings as:
.analysisSettings
temp_folder folder
analysis_type type
analysis_setup ACAFS FWCM use cores optimize level solve net seperatly floating voltage sources use unequal voltage ignore void size time duration error tolerance PG skin effect signal skin effect
full_wave_spice_model_setup passivity margin dbTol num terations skimTol asymEigval spiceName IsRefImpedence IsAsymptoticPassivity IsMaxSij FreqMin FreqMax EnableExactDC
emi_data {
    fields NearFieldRequired NearFieldSpecified FarFieldRequired FarFieldSpecified
    emi_plane X_Size Y_Size X_Sample Y_Sample Z_Above_Below Z_Location
    emi_box X_Spacing Y_Spacing Z_Spacing X_Sample Y_Sample Z_Sample
    emi_sphere radius
    emi_cylinder radius height
    emi_point radius theta phi

}
ckt_obs_data {
pwr_grid XPwrGrids YPwrGrids UseGridByCells GridLenAlongX GridLenAlongY
time_interval TimeInterval
v_pin_pair comp positive pin negative pin
…
i_pin comp pin
…
v_plane_pair above layer below layer
…
v_points above layer below layer x y
…
}
obs_data {
pwr_grid XPwrGrids YPwrGrids UseGridByCells GridLenAlongX GridLenAlongY
time_interval time interval
v_pin_pair comp positive pin negative pin
…
i_pin comp pin
…
v_plane_pair above layer below layer
…
v_points above layer below layer x y
…
}
freq_band start end type point number
…
via_iconstraint net name via constraint
…
.end [analysisSettings]

Where

folder Name of temp folder.

type Type of analysis.

0	Extraction Resistance
1	Extraction S-Parameter
2	COSIM IR
3	COSIM Dynamic
4	COSIM EMI

Definition of **analysis_setup**:

Where

ACAFS Switch of Advanced frequency sweep for AC-cosimulation.
- 0 afs off, fast sweep off, adaptive sampling off.
- 1 afs on, fast sweep off, adaptive sampling off.
- 2 value 2 is reserved, do not use.
- 3 afs on, fast sweep on, adaptive sampling off.
- 4 afs on, fast sweep off, adaptive sampling on.
- 5 afs on, fast sweep on, adaptive sampling on .

FWCM Switch for Full-Wave Circuit Model for AC S-parameter extraction.

use cores Number of cores used for analysis.

optimize level Optimization setting.
- 0 Fast
- 1 Balanced
- 2 Accurate

solve net seperatly Flag to tell if each net should be solved seperatly for DC IR drop simulation.

floating voltage sources Flag to connect -ve voltage source terminals to global ground for DC IR drop simulation.

use unequal voltage This is for DC IR drop simulation
1: Divide voltage value between pins.
0: Divide current value between pins.

time duration Time Duration value in ns for PSI Dynamic analysis.

error tolerance Folat value of error tolerance, default: -1.

PG skin effect True: Skin-effect resistance enhancement used by Sentinel-PSI

signal skin effect True: Skin-effect resistance enhancement used by Sentinel-PSI

Definition of **full_wave_spice_model_setup**:
Refer to the help document.

Definition of **emi_data**:

Where

NearFieldRequired	Set to 1 if near-field to be calculated. Otherwise set to 0.
NearFieldSpecified	Ignored. Set to 0 always
FarFieldRequired	Set to 1 if far-field to be calculated. Otherwise set to 0.
FarFieldSpecified	Ignored. Set to 0 always
X_Size	Size of near-field plane along x-axis
Y_Size	Size of near-field plane along y-axis
X_Sample	Number of samples on near-field plane
Y_Sample	Number of samples on near-field plane
Z_Above_Below	1 indicates near-field plane is above the package.
Z_Location	Distance between top-most (or bottom-most) metal layer of the package and the near-field observation plane
X_Spacing	Size of emi box along x-axis
Y_Spacing	Size of emi box along y-axis
Z_Spacing	Size of emi box along z-axis
Z_Sample	Number of samples along z-axis
Radius	Radius of the far-field sphere or cylinder
Height	Height of the far-field cylinder
Theta	Far-field theta in spherical coordinates
phi	Far-field phi in spherical coordinates

Definition of **obs_data** and **ckt_obs_data**:
Where

XPwrGrids	Power grid number along X, set 0 to disable outputting power loss.
YPwrGrids	Power grid number along Y, set 0 to disable outputting power loss.
UseGridByCells	
GridLenAlongX	Grid length along X, unit:mm.
GridLenAlongY	Grid length along Y, unit:mm.
TimeInterval	Time interval to output voltage distribution between plane pair.
comp	Name of component.
positive pin	Name of positive pin.
negative pin	Name of negative pin.
pin	Name of pin.
above layer	Name of above layer.
below layer	Name of below layer.
x	X value of point.
y	Y value of point.

Definition of **freq_band**:
Where

Start	Start frequency in Hz
End	Stop frequency in Hz
type	0=Linear and 1=Log sampling
point number	Number of samples (or points/decade for log sampling)

Definition of **via_iconstraint**:
Where

net name	Name of via net.

via constraint Via constraint value.

.spice
definition
.end [spice]

This section is optional. List spices that used in design, it defines the spice as:

For imported spice:

part cktModel **path** {
    filename
}
Or
part cktModel **path** {
    filename
    scb filename
}

For user input spice:

part cktModel **text** {
    cktModel text
}

Where

part	Name of part.
cktModel	Full name of Spice circuit model.
filename	Full path name of circuit model file.
scb filename	Full path name of .scb file.
cktModel text	Circuit mode text that user input.

.pingroups
definition
.end [pingroups]

This section is optional. List all netlists node groups for PSI. It defines each pingroup as:

name number is cpm group spice node {
node {
node1
node2
…
}

[**hookup** {
hookup1
hookup2
…
}]

[**sense** {
sense1
sense2
…
}]

[**port** {
port1
port2
…
}]
}

Where

name Name of pingroup.

number Group number of pingroup.

is cpm group If the group is a CPM group, set the value as 1, else set 0.

spice node Name of spice node.

Definition of node:

comp net pin

Where

comp Name of node component.

net Name of node net.

pin Name of pin that node connected.

Definition of hookup:

This section is optional and is used for DC IR drop voltage and current hookups. Each hookup is defined as:

global ID sign srcHookupNet srcHookupGrp source srcval

where

global ID	Port number.
sign	If the connection is positive terminal, set the value as true, else if negative terminal set it as false.
srcHookupNet	Name of source hookup net.
source	Type of source -1 voltage 0 float 1 current
srcval	Source value of voltage or current.

The sense section is optional, and the definition is same as hookup.

Definition of port:

This section is optional. It defines the port for AC S-parameter extraction as:

globalID portName isVertUp isVertDown portSign portLinkNet portLinkGrp portType R L C radius

Where

globalID	Port number.
portName	Port Name.
isVertUp	If it is a vertical up port, set value as 1, else set 0.
isVertDown	If it is a vertical down port, set value as 1, else set 0.

portSign	If it is a positive terminal, set value as 1, else if is a negative terminal, set the value as 0.
portLinkNet	The net of the other terminal to which this port is linked to.
portLinkGrp	The other terminal group number to which this port is linked.
portType	Type of port: 0 PORT_TYPE_OPEN 1 PORT_TYPE_SHORT 2 PORT_TYPE_TERM 3 PORT_TYPE_TERM 4 PORT_TYPE_GAP 5 PORT_TYPE_LUMPED 6 PORT_TYPE_COAXIAL
R L C	Value of R L C, The units are mOhm, nH, or pF.
radius	Delta radius value in um for co-axial ports.

.cktPorts
definition
.end [cktPorts]

This section is optional. It defines each circuit port as:

id name comp positive pin negative pin type R L C radius

Where

id	Port number.
name	Port name.
comp	Component name.
positive pin	Positive pin name.
negative pin	Negative pin name.
type	Port type:

0	PORT_TYPE_OPEN
1	PORT_TYPE_SHORT
2	PORT_TYPE_TERM
3	PORT_TYPE_TERM
4	PORT_TYPE_GAP
5	PORT_TYPE_LUMPED
6	PORT_TYPE_COAXIAL

R L C — Value of R L C, The units are mOhm, nH, or pF.

radius — Delta radius value in um.

.SSOChannel
Use_FastSweep_mode DoesUseFastSweepMode
Min_Freq minFreq
Max_Freq maxFreq
Max_PG_Port portNum
SSO_Net netName netType groupName
…
SSO_Comp compName DoesIncludeSpiceModel refTerminal
…
.end SSOChannel

Where

DoesUseFastSweepMode	Flag for use fast sweep mode. Y - use N - do not use
minFreq	Minimal frequency. Unit: Hz, default value is 100.0.
maxFreq	Maximum frequency. Unit: GHz, default value is 5.0.
portNum	Maximum number of PG ports, default value is 1.
netName	Name of net, enclosed by double quotation marks.
netType	Type of net, by default, all Signal nets are of type IO. All power nets are of type

"Power" .All ground nets are of type "Reference".
I – IO
P – Power
R – Reference

groupName	Name of group which a IO net belongs to, enclosed by double quotation marks. It's empty when the net doesn't belong to any group.
compName	Name of component, enclosed by double quotation marks.
DoesIncludeSpiceModel	Indicate the component includes a spice model. By default, only the components that belong to DIE and BGA type parts will have "No" set here. All other types will have "Yes" set here. Y – include N – not include
refTerminal	Three options, enclosed by double quotation. "Ref Net" "Vertical Up" "Vertical Down"

.gui
.net_color
definition
.end [net_color]
.end [gui]

This the .gui section is optional, and contains the GUI settings on XFL items.

The .net_color section will list each net color as set by user. It defines each net color as:

net name R G B

Where

net name	Name of net, enclosed by double quotation marks.
R	Red color value, 0-255.
G	Green color value, 0-255.

B Blue color value, 0-255.

THIS

PAGE

IS

INTENTIONALLY

LEFT

BLANK