P-CAD ASCII Reference Manual
Version 7.5

This is the P-CAD ASCII Reference Manual. P-CAD ASCII is a keyword-oriented structured language which represents a complete electrical design in a human-readable and fully documented machine-readable format. Because it is keyword-oriented, design files are easy to read, and not position-oriented. This format provides an interface for third-party applications, and represents P-CAD PCB designs, Schematic designs, and P-CAD libraries.

Document Syntax Conventions

This document uses a modified BNF (Backus-Naur Form) syntax to describe the P-CAD ASCII fileformat. It uses the following conventions:

- A production is of the form
  nonterminal ::= itemlist
  where *itemlist* is a list of terminals and nonterminals.
- Characters that appear within single quotes make up a literal string which appears directly in the file.
- A construct which is enclosed in [square brackets] is optional.
- A construct which is enclosed in { curly braces } may be repeated zero or more times.
- A list of items enclosed within (parentheses) and separated by vertical bars | indicates that one and only one of the items may be present.
- (PCB) or (SCH) following a production name indicates that a production is specific to P-CAD PCB or P-CAD Schematic, respectively.

ASCII File Overview

PCAD_ASCII is the top-level keyword for P-CAD ASCII files. Following this keyword is the name of the file and a header of generic information. The file then optionally contains: a *library* of style, pattern, symbol, and component definitions; a netlist of component instances and nets; and a *pcbDesign* or a *schematicDesign* with design-specific and placement information.

P-CAD ASCII files are case-insensitive and whitespace-insensitive. Comments may be added by inserting a semicolon; the comment continues from the semicolon to the end of the line.

Revision History

Changes for v7.5 (P-CAD EDA v19.0)

New productions:

Production	Brief description of production
boardCutoutObj	specifies a physical hole in the PCB board.
boardOutlineObj	specifies the outline of the PCB board.
enhancedPolygon	a polygon with straight and curved edges.
polyPoint	specifies a vertex of a enhancedPolygon and whether the following edge is straight or curved.

Changes for v7.4 (P-CAD EDA v18.4)

New productions:

Production	Brief description of production
drillSymColor	specifies the color used for a drill symbol display configuration.
drillSymDisplayConfigDef	defines the manner in which drill symbols can be displayed on a non-signal layer in the PCB workspace.
drillSymDisplayConfigRef	assigns a drill symbol display configuration to a non-signal PCB layer.

Changed productions:

Production	Description of change
drillSymSettings	added optional *drillSymDisplayConfigDef* list.
layerDef	added optional *drillSymDisplayConfigRef*.

Changes for v7.3 (P-CAD EDA v18.3)

New productions:

Production	Brief description of production
ncDrillMMFormat	describes the specific metric format for output of NC Drill data.
sheetOnlyNets	specifies that a net index table should include only those nets that reside on the sheet on which the table is placed.

Changed productions:

Production	Description of change
ncDrillSettings	added optional *ncDrillMMFormat.*
table	added optional *sheetOnlyNets* Specification.

Changes for v7.2 (P-CAD EDA v18.2)

New productions:

Production	Brief description of production
isRightReading	a flag specifying whether the attribute should always be readable from left to right and bottom to top.
numMajorColumns	describes the number of columns that a net index table will be divided into during placement.
onlineDrcSameCompPadsEnabled	specifies whether online DRC should process or ignore same-parent-component pads
orderedLayerList	specifies a list of layers in which the order of appearance in the list is significant.

table	added optional *numMajorColumns*.
variant	specifies a variant with unique name, unique name optioanal excluded components list and optional modified components list.
variantComponent	specifies a component which belongs to variant with unique name, and optional modified attribute list.
variantComponentName	specifies the name of a variant component.
variantDescription	specifies optional description of a new or existing variant.
variantExcludedComponentName	specifies the name of component which has been excluded from a variant.
variantName	specifies the name of a new or existing variant (depending on context used).

Changed productions:

Production	Description of change
attr	added optional *isRightReading*
drillTableInfo	added optional *dimensionPrecision*
onlineDrcState	added missing *onlineDrc*Enabled* productions along with optional *onlineDrcSameCompPadsEnabled*
outputItem	added optional *variantName*
printQueueEntry	modified to make *layerList* optional, and added optional *orderedLayerList*

Changes for v7.1 (P-CAD EDA v18.1)

New productions:

Production	Brief description of production
globalCopperPourCutoutBackoff Flag	specifies whether copper pours should back away from cutouts exactly to the edge of the cutout.

Changed productions:

Production	Description of change
pcbDesignHeader	added optional *globalCopperPourCutoutBackoffFlag*

Changes for v7.0 (P-CAD EDA v18.0)

New productions:

Production	Brief description of production
centerPoint	specifies the center point for triplePointArc.
endPoint	specifies the terminal point for triplePointArc.
layersStackup	specifies a list of layerStackupData.
layerStackupData	specifies data for a layer stackup entry
layerStackupDielectricConstant	specifies the dielectric constant for a layer stackup
layerStackupMaterial	specifies the material for a layer stackup entry
layerStackupName	specifies the name for a layer stackup entry
layerStackupThickness	specifies thickness of the copper or substrate for a layer stackup entry
layerPair	indicates the pairing between two layers.
layerPairs	specifies one or more layer pairings
startPoint	specifies the starting point for triplePointArc.
triplePointArc	specifies an arc with rigid control of start and end points

Changed productions:

Production	Description of change
accelAscii	default header name changed to *'PCAD_ASCII'*.
gridState	*plowGrid* and *plowViaGrid* are obsolete and are ignored.
odbLayerIncludedLayer	corrected to indicate use of stringToken, instead of booleanToken.
pcbDesign	added optional layerStackup and optional layerPairs

Deleted productions:

Production	Reason for change

Changes for v6.0 (Accel EDA v17.0)

New productions:

Production	Brief description of production
autoPlowCopperPours	flag indicates whether auto plowing of copper pours is enabled.
odbSettings	listing of ODB++ layer definitions
odbLayerDef	contains the output settings for a ODB++ layer definition
odbLayerName	*name of the ODB++ output layer, also the name of the design layer to be output.*
odbLayerContext	ODB++ layer context – can be "odbBoard", or "odbMisc"
odbLayerLayerType	ODB++ layer type – can be "odbSignal", "odbPowerGround", "odbSolderMask", odbSolderPaste", "odbSilkScreen", "odbDrill", "odbRout", "odbDocument", "odbComponent".
odbLayerPolarity	ODB++ layer polarity – can be, "odbPositive", "odbNegative".
odbLayerStartLayer	for drill layers this indicates the starting layer of the drill range.
odbLayerEndLayer	for drill layers this indicates the end layer of the drill range.
odbLayerIsSelected	indicates that this ODB++ layer has been selected for output.
odbLayerMirror	indicates that the ODB++ layer is to be mirrored.
odbLayerPads	indicates that pads are to be included in the output of the ODB++ layer.
odbLayerVias	indicates whether vias should be included during output of a ODB++ layer.
odbLayerRefdes	indicates whether a component's reference designator should be included during output of a ODB++ layer..
odbLayerType	indicates whether a component's type attribute should be included during output of a ODB++ layer.
odbLayerValue	indicates whether a component's value attributes should be included during output of a ODB++ layer.
odbLayerTitle	indicates whether the title information should be included with the output ODB++ layer.
odbLayerNoMtHoleCu	indicates that no copper should be drawn for mounting holes.
odbLayerPlatedHoles	indicates that plated holes are to be included with the drill layer.
odbLayerNonPlatedHoles	indicates that non-plated holes are to be included with the drill layer.
odbLayerIncludedLayer	name of a design layer to be included with the ODB++ output layer.
netColor	describes an optional color associated with a net.
viaThermalType	describes the thermal type for a via conneceted to a copper pour object..

viaThermalWidth	describes the thermal width for a via conneceted to a copper pour object.
viaThermalSpokes	describes the number of thermal spokes for a via conneceted to a copper pour object.

Changed productions:

Production	Description of change
CopperPour95	includes optional viaThermalType, viaThermalWidth, and viaThermalSpokes
Net	includes optional netColor
pcbDesign	includes optional odbSettings.
pcbDesignHeader	includesoptional autoPlowCopperPours
thermal	added optional thermalWidth to support different thermal widths in the same copper pour for vias and pads.

Deleted productions:

Production	Reason for change

Changes for v5.0 (Accel EDA v16.0)

New productions:

Production	Brief description of production
gluePointSize	
gluePointSizePrint	
infoPointSize	
infoPointSizePrint	
isAutoSwapPatternGraphics	indicates whether a component pattern automatically modifies its graphics when its orientation is modified.
junctionSize	
junctionSizePrint	
layerAttrs	
outputTestPoint	
patternDefExtended	defines a template used to create *patterns.* It can optionally contain multiple *patternGraphicsDef*s, which can be mapped to specific orientations using a *patternOrientationsMap*. This allows the pattern graphics to automatically change in concert with selected pattern orientations.
patternGraphicsDef	defines a template used for pattern graphics.
patternGraphicsNameDef	defines the name of a pattern graphics template.
patternGraphicsNameRef	refers to a previously defined *patternGraphicsNameDef.*
patternGraphicsRef	defines a pattern reference's pattern graphics. Its *attr* definitions can differ from its *patternGraphicsDef* template.
patternOrientation	identifies a specific orientation of a pattern.
patternOrientationAssignment	correlates a pattern orientation with its intended pattern graphics. This informs the system which pattern graphics to use for a specific orientation.
patternOrientationsMap	defines a list of *patternOrientationAssignment* definitions to inform the system which pattern graphics to use for various orientations.
pickPointSize	
pickPointSizePrint	

pourOrder	
plowGrid	
plowViaGrid	
refPointSize	
refPointSizePrint	
sheetOrderNum	
solderFlowDirection	defines wave solder flow direction during manufacturing process. Pattern graphics orientations are automatically deciphered relative to solder flow direction.
testPoint	defines a test point location on the board.
testPointAssociation	associates test points with free vias and pattern pads.
testPointSize	
testPointSizePrint	
testPointID	designates a unique identifier for a *testPointAssociation*.
testPointSide	specifies the side of the board from which the test point is accessed.
testPointSnapToCenter	specifies that a test point will snap to the center of pads/vias on which it placed and/or moved.
viaGridvisibility	

Changed productions:

Production	Description of change
arc	Added isFixed.
copperPour95	Added isFixed.
library	Added patternDefExtended.
line	Added isFixed.
pad	Added isFixed.
pattern	Added patternGraphicsNameRef, isAutoSwapPatternGraphics, and patternGraphicsRef list.
pcbDesignHeader	Added solderFlowDirection.
pcbDrawObj	Added testPoint.
via	Added isFixed and testPointAssociation.

Deleted productions:

Production	Reason for change

Changes for v4.0 (Accel EDA v15.0)

New productions:

Production	Brief description of production
chordHeight	The chord height to use when segmentizing filleted polygon corners. Used by *fillets*.
fillets	A list of fillet blocks one for each vertex of a filleted polygon.
filletDesc	The definition of a fillet. This includes the vertex point, radius and chord height.
noCopperPourConnect	Defines whether a *padShape* or v*iaShape* is forcibly prohibited from connecting to copper pours (requires a routed trace, instead)
infoPointRuleCategory	Defined the rule category type of the *infoPoint*

infoPointRuleType	Defines the rule type of the *infoPoint.*
InfoPointViolationType	Defines the violation type of the *infoPoint.*
pcbPoly	The new definition of a polygon in PCB. This supersedes *poly*.
reportColumnWidth	Defines the number of characters per page of a *reportDefinition.*
reportDefinition	Defines a custom report definition.
reportDefinitions	Defines the list of custom report definitions.
reportDestination	Defines whether the output of destination of a report.
reportExtension	Defines the file extension to be used when this report is written to a file.
reportDataFileName	Defines the name of the external file that contains MRP data to be imported into the Bill of Materials report.
reportMapFileName	Defines the name of the external file that contains mappings between ComponentName field and another field that exists in the MRP data file.
reportFieldColumnWidth	Defines the number of characters per each field in the Bill of Materials report.
reportFieldCondition	Defines a condition used in a *reportField.*
reportFieldConditions	A list containing multiple *reportFieldCondition* definitions.
reportFieldName	The name of a *reportField.*
reportField	Defines a specific report field used in a *reportDefinition.*
reportFields	A list containing multiple *reportField* definitions.
reportFieldSections	Defines an encapsulation of a *reportDefinition's reportFields*.
reportFieldSortOrder	Defines the sorting order of a specific *reportField.*
reportFieldSortType	Defines how a reportField should be sorted.
reportFieldShowFlag	Defines the flag for displaying or not displaying a *reportField.*
reportFieldType	Defines the type of a *reportField.*
reportHeader	Defines the header string for a *reportDefinition.*
reportFooter	Defines the footer string for a *reportDefinition.*
reportLinesPerPage	Defines the number of lines per page of a *reportDefinition.*
reportName	Defines the name of a *reportDefinition.*
reportPaginate	Defines the pagination flag for a *reportDefinition.*
reportSettings	Listing of report definitions for the design.
reportShowDate	Defines the flag for displaying the date of a *reportDefinition.*
reportShowFlag	Defines the flag for outputing a *reportDefinition.*
reportStyle	Defines the style of a *reportDefinition.*
reportType	Defines the type of a *reportDefinition.*
reportUserDefined	Defines whether the *reportDefinition is a user-defined.*
reportUseDesignInfo	Defines the flag for including design info in a *reportDefinition.*
reportUseFooter	Defines the flag for displaying the footer string in a *reportDefinition.*
reportUseHeader	Defines the flag for displaying the header string in a *reportDefinition.*
reportVariantName	Defines the specific variant name to be used in generating the report.

Changed productions:

Production	Description of change
copperPour95	Uses *pcbPoly* instead of *pourOutline*.
infoPoint	Added *infoPointRuleCategory, infoPointRuleType,* and *infoPointViolationType* tokens. Changed name of the *number* token to be *infoPointViolationNumber.*
islandRemoval	Added 'U*nconnected'* as a pour island removal option.
PadViaShapeType	Added *NoConnect* token.
padShape	Added optional *noCopperPourConnect* token.
plane	Now uses *pcbPoly* instead of *planeOutline.*
polyCutOut	Now uses *pcbPoly* instead of *poly*.
polyKeepOut	Now uses *pcbPoly* instead of *poly*.
viaShape	Added optional *noCopperPourConnect* token.
outputItem	Added optional *outputTitle* token.
poly	No longer includes net and tie information. This has been included in *pcbPoly*. No longer used in PCB and Relay. Use *pcbPoly*. Is used in Schematic, Symed, Pated, etc. This production is still needed to read older designs.

Deleted productions:

planeOutline	Can now be a *pcbPoly* and include fillet information. This production is still needed to read older designs
pourOutline	Can now be a *pcbPoly* and include fillet information. This production is still needed to read older designs

Changes for v3.0 (Accel EDA v14.0, Accel Document Toolbox)

New productions:

Production	Brief description of production
allComponents	Defines whether a power table includes all components.
allPins	Defines whether a power table includes all pins or hidden pins only.
arrowheadLength	Specifies the length of an arrowhead.
arrowheadWidth	Specifies the width of an arrowhead.
binaryData	Defines a sequence of binary data.
border	Defines the border of a title sheet.
column	Defines the contents of a column in a table.
columnWidth	Defines the width of a column in a table.
constraintComment	Defines a comment for an constraint attribute.
constraintFormula	Defines a formula for a constraint attribute.
constraintUnits	Defines the unit to be used for the constraintFormula.
currentPadStyle	Defines the name of the current pad style.
currentTextStyle	Defines the name of the current text style.
currentViaStyle	Defines the name of the current via style.
designView	Defines a design view object.
detail	Defines a detail object.

diagram	Defines a diagram object.
diagramInfo	Defines diagram-specific information.
diagramType	Defines the type of a diagram.
dimensionID	Designates a unique identifier for a dimension.
dimensionIDRef	Refers to a dimension object by its identifier.
dimensionIndex	Specifies the index for an object associated with a dimension.
dimensionOffsets	Specifies offset points for a dimension.
dimensionRef	Refers to a dimension object.
drillTableInfo	Defines drill table-specific information.
fieldDef	Defines a predefined or user-defined field.
fieldNameDef	Defines the name of a field.
fieldNameRef	Refers to a previously-defined field.
fieldSet	Defines a collection of field, note, and revision note definitions.
fieldSetNameDef	Defines the name of a field set.
fieldSetNameRef	Refers to a previously-defined field set name.
fieldSetRef	Refers to a previously-defined field set.
fieldValue	Defines the value of a field.
filename	Designates the name of an external file.
hexToken	Represents a 4-byte (32 bit) number in hexadecimal format.
horizontalZones	Defines the horizontal zoning information for a title sheet.
isHolePlated	A boolean flag that describes whether a *padStyleDef* or *viaStyleDef* has a plated or nonplated hole. Also differentiates holes of equivalent diameters when assigning drill symbols and tools to holes (*drillSym and toolAssn*)
isThinStrokeText	A boolean flag that describes whether a *printQueueEntry* is to use thin strokes to print text objects whose styles currently indicate stroke display mode.
isCopperTie	A boolean flag that indicates when a PCB design polygon is being used to tie two or more nets together.
layerStackupInfo	Designates layer stackup-specific information.
layerStackupStyle	Designates the style of a layer stackup diagram.
metafile	Describes an Accel Picture object.
noteAnnotation	Defines the graphical annotation of a specific note.
noteDef	Defines a numbered note.
noteNum	Defines the number of a specific note.
noteRef	Refers to a previously-defined note.
noteTableInfo	Defines note table-specific information.
noteValue	Defines the value of a specific note.
numBytes	Defines the number of bytes in a *binaryData* production.
numDirection	Allows specification of ascending or descending numbering.
outputDrillSymNonplated	A boolean flag that directs various output utilities to output drill symbols and/or hole locations for nonplated holes.

outputDrillSymPlated	A boolean flag that directs various output utilities to output drill symbols and/or hole locations for plated holes.
outputNoMountingHole	A boolean flag that directs various output utilities to suppress output of mounting hole copper.
powerTableInfo	Defines power table-specific information.
revisionNoteDef	Defines a numbered revision note.
revisionNoteRef	Refers to a previously-defined revision note.
room	Defines to a new pcb room object
roomAttrMgr	Refers to a previously-defined attrMgr.
roomFillPattern	Defines to the fill pattern of the room. Describes a enum used to indicate that a room's fill pattern is solid, clear, or hatched.
roomInclusionList	Refers to a previously-defined refDesRef. Describes a list of refDesRef.
roomPlacementSide	Defines to the placement side of the room. Describes a enum used to indicate that a room's placement side is top, bottom, or top-or-bottom.
row	Defines the contents of a row in a table.
subtitle	Designates the subtitle of a detail or diagram object.
table	Defines a table object.
tableInfo	Defines table-specific information.
tableType	Designates the type of a table.
textStyleAllowTType	A boolean flag that specifies whether a text style has a TrueType font mode.
textStyleDisplayTType	A boolean flag that specifies the current display mode for a text style (stroke or TrueType).
tieNetValue	A string value that specifies a unique name for associating nets that are tied together using a polygon tie.
title	Designates the title of an object.
verticalZones	Defines the vertical zoning information for a title sheet.
xRef	Defines a Schematic sheet connector cross reference.
zones	Defines the zoning information for a title sheet.

Changed productions:

Production	Description of change
arc	Added optional list of *dimensionRef*s.
attribute	Added three optionals - *constraintComment, constraintFormula, constraintUnits*
designInfo	Added a list of field sets. Made existing fields optional.
dimension	Added optional *arrowheadWidth, arrowheadLength, dimensionID,* and *dimensionOffsets*.
drillSym	Added *isHolePlated* token.
fieldType	Added designation of fields by string token. Added *noteRef* and *revisionNoteRef*.
layerDef	Added a list of attributes to signal and plane layers. Remove the hard-coded clearance values. Added optional *titleSheet* and *fieldSetRef*.
line	Added optional list of *dimensionRef*s.
netList	Added optional *globalAttrs*.

outputItem	Added optional *outputDrillSymPlated*, *outputDrillSymNonplated*, and *outputNoMountingHole* tokens.
pad	Added optional list of *dimensionRef*s.
padStyleDef	Added optional *isHolePlated* token.
pattern	Added optional list of *dimensionRef*s.
pcbDrawObj	Added *detail*, *diagram*, *metafile*, *room,* and *table* to the list of objects.
printQueueItem	Added *isThinStrokeText* token.
programState	Added optional *currentTextStyle*, *currentPadStyle*, and *currentViaStyle*. The *currentPadStyle* and *currentViaStyle* are not used by SCH.
schDrawObj	Added *table* and *xRef* to the list of objects.
sheet	Added optional *fieldSetRef* token.
textStyleDef	Added optional (second) *font*, *textStyleAllowTType*, and *textStyleDisplayTType* tokens.
titleSheet	Added optional *offset*, *border*, and *zones*. Added *pcbDrawObj* to list of contained objects.
toolAssn	Added *isHolePlated* token.
via	Added optional list of *dimensionRef*s.
viaStyleDef	Added optional *isHolePlated* token.

Changes for v2.3 (ACCEL EDA v13.0)

New productions:

Production	Brief description of production
entireDesign	Describes a boolean flag whether to print a defined region or the extents of the layers/sheet. Used in both PCB and Schematic.
isFixed	Describes a boolean flag used to indicate that a component's or room's location is fixed.
IsVisibleOnDrag	Describes a boolean flag used to indicate that an item that is otherwise hidden is temporarily displayed while its location or orientation is being edited in the workspace.
pageSize	Describes the output size for printing in Schematic.
pinDesignatorProperty	Describes the display properties for a pin designator on symbol pin.
pinNameProperty	Describes the display properties for a pin name on symbol pin.
printRegion	Describes the coordinates of a rectangle to be used when printing a specified region. Used in both PCB and Schematic.
polyDescriptor	Describes a regular polygon shape.
polyShape	Describes a polygon shape.
scaleFactor	Describes the user defined scale factor for printing. Schematic only.
scaleToFitPage	Describes the boolean flag whether to adjust the print output to fit exactly on one page. Used in both PCB and Schematic.
shapeOutline	Defines a polygon shape for a pad or via style.
shapeSidesDfn	Defines the number of sides in a parametric definition of a polygonal shaped pad or via.

Changed productions:

Production	Description of change
apertureDef	Added optional *polyShape* token.
dimensionStyle	Added stringToken *dim_datum* for representation of new dimension type.
field	Added optional *justify* token.
net	Added optional *isVisibleOnDrag* token.
padShapeDfn	Added optional description for a polygon shape.
padViaShapeType	Added token *polygon* to list of possible shapes.
pattern	Added optional *isFixed* token.
pin	Added optional pin designator or pin name descriptions.
printQueueEntry	Added optional *entireDesign* token. Added optional *scaleToFitPage* token. Added optional *printRegion* token.
schPrintSettings	The items *scaling*, *offset*, *isRotated*, *drawBorder* are now optional.
sheet	Sheets now contain their own printing information. Each sheet can now be printed with its own unique settings. Added optional *drawBorder*, *entireDesign*, *isRotated*, *pageSize*, *scaleFactor*, *offset*, *printRegion* tokens.
viaShapeDfn	Added optional description for a polygon shape.

Changes for v2.2 (Accel EDA v12.10)

Changed productions:

Production	Description of change
drillSym	added stringToken for representation of alpha-character drill symbols.
compType	added '*Module*' and '*Link*' tokens.

Changes for v2.1

New productions:

Production	Brief description of production
fontCharSet	describes the character set of a TrueType font.

Changed productions:

Production	Description of change
font	added TrueType font values.
gluePoint	added optional *isVisible* parameter.
pickPoint	added optional *isVisible* parameter.

Changes for v2.0

New productions:

Production	Brief description of production
area	specifies an area in square database units.
classToClass	defines rules between two *netClasses*.
copperPour95	describes the new, island-based copper pour object.
copperPourIsland	an island belonging to a *copperPour95*.
copperPourIslandCutout	a cutout belonging to a *copperPourIsland*.
cutoutOutline	the polygonal outline of a *copperPourIslandCutout*.

dimension	describes a dimension object.
endRange	describes the ending layer for a hole range.
endStyle	describes the *endStyle* of a line's end point.
fontClipPrecision	describes the clip precision of a TrueType font.
fontItalic	defines whether a TrueType font is italicized
fontOutPrecision	describes the output precision of a TrueType font.
fontPitchAndFamily	describes the pitch and family of a TrueType font.
fontQuality	describes the quality of a TrueType font.
fontWeight	describes the weight of a TrueType font.
islandOutline	the polygonal outline of a *copperPourIsland*.
islandRemoval	describes island removal modes for *copperPour95*.
layerSet	describes those layers contained in a layer set..
layerSets	describes a block of layerSet.
layerSetList	A list of one or more layerSet.
netClass	describes a collection of nets that share common attributes.
netClassNameDef	defines the name of a net class in a design.
netClassNameRef	provide a reference to an existing net class.
netPlaneColor	describes a color associated with a plane net.
plane	describes a plane object.
planeOutline	describes the polygonal outline of a plane object.
pourSmoothness	describes the smoothness factor used by *copperPour95*.
startRange	describes the starting layer for a hole range.
thermalSpokes	the number of thermal spokes used by *copperPour95*.
useDesignRules	the backoff method used by *copperPour95*.

Changed productions:

Production	Description of change
font	added TrueType font values.
fonttype	added new value *TrueType*.
fontFamily	added new value *Modern.*
net	the *isPlane* field is no longer necessary (the field is ignored, if present), and an optional *netPlaneColor* field describes the color for a plane net.
netList	added an optional *netClass* list and optional *classToClass* list.
pcbDrawObj	added an optional *plane* object.
padStyleDef, viaStyleDef	Both productions have two additional variables. Optional4 and Optional5, for startRange and EndRange respectively.
pcbDesign	One added variable for layerSets.

Changes for v1.2

New productions:

Production	Brief description of production
compAlias	describes an alias for a component name.
patternAlias	describes an alias for a pattern name.
portPinLength	defines the length of the pin or pins in a schematic port object.
symbolAlias	describes an alias for a symbol name.

Changed productions:

Production	Description of change
compPin	the value of the *partNum* field can now define common pins.
library	*library* now includes optional sections for name aliases for components, patterns, and symbols.
port	port now includes an optional *portPinLength* that defines the length of the pin or pins on a port object.
portType	*portType* now includes definitions for optional port shapes.

Changes for v1.1

New productions:

Production	Brief description of production
extent	represents the size of a bounding box.
onlineDrcEnabled	indicates the state of Online DRC checking.
onlineDrcReport	indicates whether Online DRC reports should be automatically generated.
onlineDrcState	describes the state of Online DRC settings.
port	defines a port object in a schematic design.
portType	*portType* defines the shape, the number of pins, and the pin orientation, for a schematic port object.
viaToLineClearance	defines the minimum DRC clearance between a via and a line.
viaToPadClearance	defines the minimum DRC clearance between a via and a pad.
viaToViaClearance	defines the minimum DRC clearance between a via and a via.

Changed productions:

Production	Description of change
layerDef	*layerDef* now supports optional clearance values for via-to-line, via-to-pad, and via-to-via.
programState	*programState* now includes an optional *onlineDrcState*. *onlineDrcState* is not used by P-CAD Schematic.
schDrawObj	*schDrawObj* now includes an optional port object.
text	text now includes an optional extent, which indicates the bounding box of the text. extent is always written by P-CAD PCB and P-CAD Schematic, but is ignored during File Open.

The alphabetized list of productions makes up the remainder of this document.

PCAD_ASCII

PCAD_ASCII ::= 'PCAD_ASCII'
asciiNameDef
asciiHeader
[library]
[netlist]
[pcbDesign | schematicDesign]

Description
PCAD_ASCII is the top-level keyword for PCAD ASCII files. Following this keyword is the name of the file and a header of generic information. The file then optionally contains: a *library* of style, pattern, symbol, and component definitions; a *netlist* of component instances and nets; and a *pcbDesign* or a *schematicDesign* with design-specific and placement information. Currently, PCAD ASCII supports only one *library,* one *netlist,* and one *pcbDesign* or *schematicDesign* per file.

Notes
To provide compatibility with TangoPRO ASCII designs, the keywords '*ACCEL_ASCII'* and *'TangoPRO_ASCII'* are also supported as a valid keyword.

Used By
none

allComponents (SCH)

allComponents ::= '(' 'allComponents'
booleanToken
[refDesPrefix]
')'

Description
allComponents defines whether a power table includes all components or only components of a specified RefDes prefix. If the boolean token is False, then the table is restricted to components of the specified RefDes prefix.

Used By
powerTableInfo

allPins (SCH)

allPins ::= '(' 'allPins'
booleanToken
')'

Description
allPins defines whether a power table includes all pins or hidden pins only.

Used By
powerTableInfo

alts

alts ::= '(' 'alts'
ieeeAlt
deMorganAlt
')'

Description
alts defines the existence or absence of IEEE and DeMorgan alternate symbol representations for a component.

Used By

compHeader

altType

altType ::= '(' 'altType'
('normal' | 'ieee' | 'demorgan')
')'

Description
altType indicates the representation type for a symbol: Normal, IEEE, or DeMorgan.

Used By
attachedSymbol, symbol

anotherEnd

anotherEnd ::= location

Description
anotherEnd is the second of two locations defining an item.

Used By
bus, fromTo, line, lineKeepOut

apertureAssn (PCB)

apertureAssn ::= '(' 'apertureAssn'
apertureRef
itemMnemonic
')'

Description
apertureAssn specifies the mapping between a Gerber photoplotter aperture definition and its mnemonic. This assignment reflects the **Item** and **Aperture** columns of the main list box for a specific aperture in the Aperture Assignments dialog.

Used By
gerberSettings

apertureDef (PCB)

apertureDef ::= '(' 'apertureDef'
dCode
apertureShape
width
height
holeDiam
desc
[apertureType]
[rotation]
[offset]
[polyShape]
')'

Description
apertureDef defines a PCB Gerber aperture. An aperture is defined by its type, shape, dimensions, hole diameter, and a description. References to aperture definitions are made by the definition *dCode*.

Notes
PolyShape is required if *apertureShape* indicates Polygon.

Defaults

apertureType: flash
rotation: 0 degrees
offset: none
polyShape: none

Used By

gerberSettings

apertureRef (PCB)

apertureRef ::= '(' 'apertureRef'
dCode
')'

Description

apertureRef is a specific reference to a previously-defined *apertureDef* . *apertureDefs* are referenced by the aperture's *dCode*.

Used By

apertureAssn

apertureShape (PCB)

apertureShape ::= '(' 'apertureShape'
('Ellipse' | 'Oval' | 'Rect' | 'RndRect' | 'Thrm2' |'Thrm2_90' | 'Thrm4' | 'Thrm4_45' |
'Target' |'MtHole' | 'DrlSym' | 'Polygon')
')'

Description

apertureShape describes the shape of an aperture. This value reflects the status of the **Shape** combo box in the Describe/Assign Apertures dialog for an aperture.

Used By

apertureDef

apertureType (PCB)

apertureType ::= '(' 'apertureType'
('flash' | 'draw' | 'flash/draw')
)'

Description

apertureType defines the type of an aperture. This value reflects the status of the **Type** combo box in the Describe/Assign Apertures dialog for an aperture.

Used By

apertureDef

area (PCB)

area ::= '(' 'area'
numberToken
')'

Description

area represents a number of square database units.

Used By

islandRemoval

arc

arc ::= '(' 'arc'
location
radius
startAngle
sweepAngle
width
[isFlipped]
[netNameRef]
{ dimensionRef }
[isFixed]
')'

Description
arc represents an arc object. The arc is defined as the curve spanning from a start angle, sweeping through *sweepAngle* degrees in a counter-clockwise direction, given a center point and a radius. The definition also includes the line width.

Notes
Arcs in the *library* section of P-CAD PCB design files do not have net references. These references, if present, are ignored. Arcs in Schematic designs do not have net references and will be ignored if present. For all other arcs, the arc does not belong to a net if the net reference is not present.

Defaults
isFlipped: False
isFixed: False

Used By
pcbDrawObj, schDrawObj

arrowheadLength (PCB)

arrowheadLength ::= '(' 'arrowheadLength'
dbNumber
')'

Description
arrowheadLength describes the length of an arrowhead.

Used By
dimension

arrowheadWidth (PCB)

arrowheadWidth ::= '(' 'arrowheadWidth'
dbNumber
')'

Description
arrowheadWidth describes the width of an arrowhead.

Used By
dimension

asciiHeader

asciiHeader ::= '(' 'asciiHeader'
asciiVersion
[written]
{ headerString }

fileUnits
')'

Description
asciiHeader provides ASCII format version and design file information.

Used By
PCAD_ASCII

asciiNameDef

asciiNameDef ::= nameDef

Description
asciiNameDef denotes the design file name.

Used By
PCAD_ASCII

asciiVersion

asciiVersion ::= '(' 'asciiVersion'
majorVersion
minorVersion
')'

Description
asciiVersion denote the P-CAD ASCII file format version.

Used By
asciiHeader

attachedPattern

attachedPattern ::= '(' 'attachedPattern'
patternNum
patternName
numPads
padPinMap
')'

Description
attachedPattern describes a pattern attached to a component. The definition relates a pattern number with the name given a pattern. This pattern name is the same as the **Pattern Name** field in the Create Component and Modify Component dialogs.

Used By
compDef

attachedSymbol

attachedSymbol ::= '(' 'attachedSymbol'
partNum
altType
symbolName
')'

Description

attachedSymbol describes a symbol attached to a component. The definition relates a part number with the name given a symbol. This symbol name is the same as the **Symbol Name** field in the Create Component and Modify Component dialogs.

Used By

compDef

attr

attr ::= '(' 'attr'
attributeNameDef
attributeStringValue
[location]
[rotation]
[isFlipped]
[isVisible]
[justify]
textStyleRef
[constraintFormula]
[constraintComment]
[constraintUnits]
[isRightReading]
')'

Description

attr represents an attribute object. Attributes may be associated with other objects in a P-CAD design, or may be created independent from other objects.

Notes

The presence or absence of the optional data within an attribute is dependent on whether the attribute is associated with another object, and may be specific to the type of object to which it is associated.

location is always present for attributes not associated to other objects.

Defaults

The following defaults may be overridden by special handling for specific object types.
location: (0,0)
rotation: 0 degrees
isFlipped: False
isVisible: False
isRightReading: False
justify: LowerLeft

Used By

compDef, compInst, globalAttrs, net, pattern, pcbDrawObj, room, schDrawObj, symbol

See Also

compDef, compInst, pattern, patternDef, symbol, symbolDef

attributeNameDef

attributeNameDef ::= stringToken

Description

attributeNameDef is the name assigned to an attribute. This name reflects the **Name** field in the Place Attribute and Modify Attribute dialogs.

Used By

attr

attributeStringValue

attributeStringValue ::= stringToken

Description

attributeStringValue is the value assigned to an attribute. This value reflects the **Value** field in the Place Attribute and Modify Attribute dialogs.

Used By

attr

autoClear (PCB)

autoClear ::= '(' 'autoClear'
booleanToken
')'

Description

The *autoclear* flag indicates whether tool or aperture assignments are to be cleared prior to automatic assignment. This value reflects the state of the **Clear Current Tools** and **Clear Current Apertures** check boxes in the File N/C Drill Tool Assignments and Gerber Out Aperture Assignments dialogs, respectively.

Used By

gerberSettings, ncDrillSettings

autoDrawApertureSize (PCB)

autoDrawApertureSize ::= '(' 'autoDrawApertureSize'
dbNumber
')'

Description

autoDrawApertureSize indicates the draw aperture size given to all automatic aperture assignments. This value reflects the value found in the **Draw Aperture Size** edit box of the Gerber Out Aperture Assignments dialog.

Used By

gerberSettings

autoPlowCopperPours

autoPlowCopperPours ::= '(' 'autoPlowCopperPours'
booleanToken
')'

Description

autoPlowCopperPours indicates the whether the Auto Plow Pours option under Options Configure is enabled for that design..

Used By

pcbDesignHeader

binaryData (PCB)

binaryData ::= '(' 'binaryData'
numBytes
{ hexToken }
')'

Description

binaryData is used to store data that cannot be otherwise stored in textual format. This production simply contains a specified number of bytes in hexadecimal format.

Used By
metafile

boardCutoutObj (PCB)

boardCutoutObj ::= '(' 'boardCutoutObj'
enhancedPolygon
[boardWidth]
dimensionRef
')'

Description
boardCutoutObj is used to describe a hole in the board.

Used By
pcbDrawObj

boardOutlineObj (PCB)

boardOutlineObj ::= '(' 'boardOutlineObj'
enhancedPolygon
[boardWidth]
dimensionRef
')'

Description
boardOutlineObj is used to describe the outline of the board.

Used By
pcbDrawObj

booleanToken

booleanToken ::= ('True' | 'False')

Description
A *booleanToken* describes a value which may have one of two possible values: 'True' or 'False'.

Used By
many productions

border

border ::= '(' 'border'
isVisible
height
width
offset
')'

Description
border is used to describe the border of a title sheet.

Used By
titleSheet

bus (SCH)

bus ::= '(' 'bus'
busNameDef

oneEnd
anotherEnd
[dispName]
[text]
')'

Description

bus represents a bus object. The bus is defined between the locations *oneEnd* and *anotherEnd*. The *text* item, if present, indicates the text style and placement of the bus name text.

Defaults

dispName: False
text: The default text style and bus name location are used

Used By

schDrawObj

busEntry (SCH)

busEntry ::= '(' 'busEntry'
busNameRef
busPoint
orient
')'

Description

busEntry indicates the placement and orientation of the graphical representation for a bus entry point. This is written for information only; these values are re-calculated by P-CAD programs during File Open, and are ignored when read from the input file.

Used By

schDrawObj

busNameDef (SCH)

busNameDef ::= nameDef

Description

busNameDef defines the name for a bus object. This is the name input in the **Bus Name** edit box from the Create Bus dialog.

Used By

bus

busNameRef (SCH)

busNameRef ::= '(' 'busNameRef'
nameRef
')'

Description

A *busNameRef* references a bus item by the bus name, *nameRef.*

Used By

busEntry

busPoint (SCH)

busPoint ::= pt

Description

A *busPoint* indicates a location on a bus.

Used By

busEntry

camQueueEntry (PCB)

camQueueEntry ::= '(' 'camQueueEntry'
entryName
layerList
outputItem
')'

Description

A *camQueueEntry* describes the output parameters for a CAM output job, specifically Gerber and N/C Drill output, based on the fields and selections in the File Gerber Out and File N/C Drill dialogs, respectively. There is one *camQueueEntry* for each output file in the **Output Files** list box.

Used By

gerberSettings, ncDrillSettings

centerPoint

centerPoint ::= location

Description

centerPoint is a center point locatoin.

Used By

triplePointArc

chordHeight (PCB)

chordHeight ::= '(' 'chordHeight'
dbNumber
')'

Description

Used for segmentization threshold of filleted polygon corners. *chordHeight* defines the minimum spacing between the center of a segment on the curve and the ideal arc.

Used By

filletDesc

classToClass

classToClass ::= '(' 'classToClassRules'
netClassNameRef
netClassNameRef
{ attr }
{ LayerAttrs }
')'

Description

classToClass defines a set of attributes between two net classes.

Used By

netlist

clearance

clearance ::= '(' 'clearance'
dbNumber
')'

Description
clearance specifies the distance value that has been calculated where a DRC netlist clearance voilation has been detected. This is the clearance value listed in the DRC report and is displayed as part of the InfoPoint information. This value is read and ignored if present in a Schematic design file.

Used By
infoPoint

codeFormat (PCB)

codeFormat ::= '(' 'codeFormat'
('eiaOdd' | 'asciiEven' | 'asciiNone')
')'

Description
codeFormat specifies the ASCII code format to be used for an N/C Drill output file. This value reflects the status of the **Output Code Type** radio buttons in the N/C Drill Format dialog.

Used By
ncDrillSettings

column

column ::= '(' 'column'
title
columnWidth
{ row }
')'

Description
column defines the contents of a column in a table.

Used By
drillTableInfo

columnWidth

columnWidth ::= '(' 'columnWidth'
integerToken
')'

Description
columnWidth defines the width of a column in a table.

Used By
column, noteTableInfo

compAlias

compAlias ::= '(' 'compAlias'
compNameDef
compNameRef
')'

Description
compAlias defines a name alias *compNameDef* for the component name *compNameDef.*

Notes

There may be more than one *compAlias* for a given component name; additional *compAliases* will be created for each component name alias.

Used By

library

compDef

compDef ::= '(' 'compDef'
componentNameDef
originalName
compHeader
{ compPin }
{ attachedSymbol }
{ attachedPattern }
{ attr }
')'

Description

A *compDef* is a component template. The information in this template is used to create components of the same type in a Schematic or PCB design by providing the logical design information.

Notes

There may be more than one *compDef* for a given component type. Additional *compDefs* are created if two components of the same type are not equivalent. The *componentNameDef* uniquely identifies the template; components reference templates by this name.

originalName is the component Type as displayed in the **Type** field of component create and modify dialogs, and P-CAD reports.

compPin data is present if the component has component pins.

attachedPattern and *attachedSymbol* data is written if it exists for the given component type in the active design. For P-CAD Schematic designs, *attachedSymbol* data is always present and will be written to the design file. Similarly, for PCB designs, *attachedPattern* data is always present for each component and will always be written.

Attribute data is written for PCB components only. The attributes written, if present, will include global predefined attributes associated to this component type. The attribute Value string is written as an empty string (two double-quotes). These global, predefined attributes will be repeated as part of the *compInst* definition, and will include the actual value for the attribute.

Used By

library

See Also

compInst

compHeader

compHeader ::= '(' 'compHeader'
[sourceLibrary]
[compType]
numPins
numParts
[composition]
[alts]
refDesPrefix
[numType]

')'

Description

compHeader describes the summary data for a component including the number of pins, number of parts, existence of alternate symbol representations, and component composition.

Notes

sourceLibrary is written only for Schematic designs.

Defaults

compType: Normal
composition: Homogeneous
alts: Component has only Normal symbol representations if not written
numType: Alpha

Used By

compDef

compInst

compInst ::= '(' 'compInst'
refDesDef
compRef
[originalName]
[compValue]
[patternName]
{ attr }
')'

Description

compInst represents a component in the active design. This instance of a component includes the component RefDes, and references the component template (*compRef*) that provides the logical definition for the component.

Notes

If the component references a *compDef* which is not equivalent to a component of the same type already placed in the design, an error will be output and the component will not be added.

originalName is the component Type as displayed in the **Type** field of component create and modify dialogs, and P-CAD reports. It is written for clarity, but since this information is also written as part of the *compDef,* the name is not read during File Open.

The *compValue* is written if the component has the Value attribute defined.

patternName is written only for Schematic design files the user has added a pattern name using the **Pattern** edit box in the Modify Part dialog.

attr is written for all local component attributes in PCB, and for all component attributes except RefDes, Type, and Value in Schematic.

Used By

netlist

componentNameDef

componentNameDef ::= nameDef

Description

componentNameDef provides a unique name for a component template that is referenced by components in the design.

Used By

compDef

componentNameRef

componentNameRef ::= nameRef

Description

componentNameRef is a reference to a component template previously defined in the library section.

Used By

compRef

componentPinDesDef

componentPinDesDef ::= nameDef

Description

componentPinDesDef defines the pin designator associated with a component pin. Each pin designator is unique within a given component. This is the same designator located in the Library Manager spreadsheet **Pin Designator** column, and referenced from P-CAD netlist node names.

Used By

compPin

componentPinDesRef

componentPinDesRef ::= nameRef

Description

componentPinDesRef is a reference to a component pin previously defined within a component in the *library* section.

Used By

compPinRef

compPin

compPin ::= '(' 'compPin'
componentPinDesDef
[pinName]
partNum
symPinNum
gateEq
pinEq
[pinType]
')'

Description

compPin specifies the logical definition of a component pin. The definition includes the pin designator, pin name, part number, symbol pin number, gate equivalence value, pin equivalence value, and pin type. *pinName* is written if the pin name exists.

Notes

The *partNum* field defines the gate for this *compPin*. Two special values are defined for the *partNum* field: if *partNum* is 0, then the *compPin* is defined as a global hidden power pin. If *partNum* is -1, then the *compPin* is defined as a common pin.

The *gateEq* for a common pin defines the scope of the common pin. A *gateEq* of 0 defines a common pin for all gates in the component. All other *gateEq* values define the common pin for gates with that value.

Defaults

pinType: Unknown

Used By

compDef

compPinRef

compPinRef ::= '(' 'compPinRef'
componentPinDesRef
')'

Description

compPinRef is a reference to a previously defined component pin, by its pin designator.

Used By

node

compRef

compRef ::= '(' 'compRef'
componentNameRef
')'

Description

compRef is a reference to a component template previously defined in the *library* section.

Used By

compInst

compType

compType ::= '(' 'compType'
('Normal' | 'Power' | 'SheetConnector')
')'

Description

compType specifies a component type for a component.

Used By

compHeader

composition

composition ::= '(' 'composition'
('Homogeneous' | 'Heterogeneous')
')'

Description

A component *composition* is homogeneous if all of the gates defining the component are identical, and heterogeneous if not.

Used By

compHeader

compValue

compValue ::= '(' 'compValue'
stringToken
')'

Description

compValue is the Value attribute associated with a component. This is the component value that may be assigned to a component by setting the **Value** edit box in the Modify Component and Modify Part dialogs.

Used By

compInst

constraintComment

constraintComment ::= '(' 'constraintComment'
stringToken
')'

Description

constraintComment is a user defined note explaining the associated constraintFormula.

Used By

attr

constraintFormula

constraintFormula ::= '(' 'constraintFormula'
stringToken
')'

Description

constraintFormula is a formula that defines a numeric or logical constraint.

Used By

attr

constraintUnits

constraintUnits ::= '(' 'constraintUnits'
'mil' | 'inch' |
'millimeter' | 'centimeter' | 'micrometer' | 'nanometer' | 'picometer' | 'meter' |
'layername' |
'viastyle' |
'radian' | 'degree' |
'ohm' | 'mho' |
'volt' | 'millivolt' | 'microvolt' | 'nanovolt' | 'picovolt' |
'ampere' | 'milliampere' | 'microampere' | 'nanoampere' | 'picoampere' |
'henry' | 'millihenry' | 'microhenry' | 'nanohenry' | 'picohenry' |
'farad' | 'millifarad' | 'microfarad' | 'nanofarad' | 'picofarad' |
'second' | 'millisecond' | 'microsecond' | 'nanosecond' | 'picosecond' |
'bool' |
'hertz' | 'kilohertz' | 'megahertz' | 'gigahertz' |
'watt' | 'milliwatt' | 'microwatt' | 'nanowatt' | 'picowatt' |
'quantity' |
'string' |
'celsius' | 'fahrenheit' | 'app'
')'

Description

constraintUnits is used to define the units of the associated constraintFormula.

Used By

attr

copperPour (PCB)

copperPour ::= '(' 'copperPour'
pourType
[netNameRef]
width
pourSpacing
pourBackoff
[thermalType]
[thermalWidth]
pourOutline
[fillPoly]
{ vertex }
{ thermal }
')'

Description

copperPour represents a copper pour object. The pour definition includes the external pour boundary, internal boundary polygons and vertex lists, thermals within the pour, and pour parameters.

Notes

netNameRef is omitted if the pour does not belong to a net.

If the pour has not been poured, only the pour outline data is included in the definition. This is the original polygon used to define the pour.

For poured pours, the definition includes a list of *fillPoly* polygons and a vertex list. The polygons and vertices define the actual boundaries of the poured region and items that intersect the poured outline. These polygons and vertices reflect the copper pour backoff and indicate which portions of the vertex list are inside or outside the pour boundary.

Defaults

thermalType: NoTherm
thermalWidth: 0

Used By

pcbDrawObj

copperPour95 (PCB)

copperPour95 ::= '(' 'copperPour95'
pourType
[netNameRef]
width
pourSpacing
pourBackoff
[useDesignRules]
pourSmoothness
[thermalType]
[thermalWidth]
[thermalSpokes]
islandRemoval
[viaThermalType]
[viaThermalWidth]
[viaThermalSpokes
pourOutline
{ copperPourisland }
[isFixed]
')'

Description

copperPour95 represents a new, island-based copper pour object and superscedes *copperPour*. This new version represents a copper pour as a series of islands rather than an unrelated collection of edges. The new pour also supports island removal criteria, a smoothness factor and the ability to define the backoff from the system design rules.

Notes

copperPour95 should be used in place of *copperPour* if the design system supports it.

netNameRef is omitted if the pour does not belong to a net.
thermalSpokes currently is fixed at 4 and cannot be changed.
viaThermalSpokes currently is fixed at 4 and cannot be changed.

If the pour has not been poured, only the pour outline data is included in the definition. This is the original polygon used to define the pour.

For poured pours, a list of islands follows the definition data. Islands inherit the same characteristic, such as line width, hatch and thermal types, as the parent pour. Islands may in turn have cutouts that represent void areas - holes - that lie completely inside the island.

Defaults

pourSmoothness: 1 (low)
thermalType: NoTherm
thermalWidth: 0
useDesignRules: True
isFixed: False
viaThermalType: thermalType
viaThermalWidth: thermalWidth

Used By

pcbDrawObj

copperPourIsland (PCB)

copperPourIsland ::= '(' 'island'
islandOutline
{ copperPourIslandCutout }
{ thermal }
')'

Description

copperPourIsland represents an island belonging to a *copperPour95* object. A *copperPourIsland* may also have cutout regions - holes - as well as a list of thermals.

Used By

copperPour95

copperPourIslandCutout (PCB)

copperPourIslandCutout ::= '(' 'cutout'
cutoutOutline
')'

Description

copperPourIslandCutout represents cutout regions - holes - inside a *copperPourIsland*.

Used By

copperPourIsland

copyright

copyright ::= '(' 'copyright'
stringToken
')'

Description
copyright is string available to denote the copyright statement output by the application generating the design file.

Used By
written

currentAbsGrid

currentAbsGrid ::= '(' 'currentAbsGrid'
gridString
')'

Description
currentAbsGrid is a string that describes the design file absolute grid setting, as listed in the Options Grids dialog. The string includes the grid units.

Used By
gridState

currentLayer

currentLayer ::= '(' 'currentLayer'
layerNumRef
')'

Description
currentLayer denotes the PCB layer or Schematic sheet that was current at the time the design was saved.

Used By
layerState

currentPadStyle (PCB)

currentPadStyle ::= '(' 'currentPadStyle'
stringToken
')'

Description
CurrentPadStyle is a string that is the name of a pad style in the design. The pad style will be used to indicate the current pad style used when placing pads.

Notes
currentPadStyle is read but ignored if present in a Schematic design file.

Used By
programState

currentRelGrid

currentRelGrid ::= '(' 'currentRelGrid'
gridString
')'

Description
currentRelGrid is a string that describes the design file relative grid setting, as listed in the Options Grids dialog. The string includes the grid units.

Used By
gridState

currentTextStyle

currentTextStyle ::= '(' 'currentTextStyle'
stringToken
')'

Description
currentTextStyle is a string that is the name of a text style in the design. The text style will be used to indicate the current text style used when placing text.

Used By
programState

currentViaGrid

currentViaGrid ::= '(' 'currentViaGrid'
gridString
')'

Description
currentViaGrid is a string that describes the design file via grid setting, as listed in the Route Configure dialog. The string includes the grid units.

Notes
currentViaGrid is read but ignored if present in a Schematic design file.

Used By
gridState

currentViaStyle (PCB)

currentViaStyle ::= '(' 'currentViaStyle'
stringToken
')'

Description
currentViaStyle is a string that is the name of a via style in the design. The via style will be used to indicate the current via style used when placing vias.

Notes
currentViaStyle is read but ignored if present in a Schematic design file.

Used By
programState

cutoutOutline (PCB)

cutoutOutline ::= '(' 'cutoutOutline'
{ pt }
')'

Description
cutoutOutline lists the vertices of the outline of a *copperPourIslandCutout.*

Used By
copperPourIsland

date

date ::= yearNumber
monthNumber
dayNumber

Description
A *date* is represented as three numeric values: year month date.

Used By
timeStamp

dayNumber

dayNumber ::= integerToken

Description
dayNumber is an integer number that ranges from 1 to 31.

Used By
date

dbNumber

dbNumber ::= numberToken
[dbUnit]

Description
Design object widths, lengths, and locations are represented as *dbNumbers*. These numbers are written in user-units, and will specify the unit type *as mil, mm*, or *in* if the unit type is different from what is specified as the *fileUnits* for the design file.

Notes
If it is not possible to represent a database values exactly in the current design user units, a unit will be chosen that can represent the value and the *dbUnit* notation will be added to the number .

Used By
many productions

dbUnit

dbUnit ::= ('mil' | 'mm' | 'in')

Description
dbUnit denotes a unit type for numeric values. Supported units are mils, millimeters, and inches.

Used By
dbNumber, fileUnits

dCode (PCB)

dCode ::= '(' 'dCode'
integerToken
')'

Description
dCode represents the draft code for a Gerber aperture. This code is specified in the **D Code** edit box in the Describe/Assign Apertures dialog.

Used By
apertureDef, apertureRef

deMorganAlt

deMorganAlt ::= '(' 'deMorganAlt'
booleanToken
')'

Description
deMorganAlt indicates whether or not a DeMorgan Alternate representation for a symbol exists. This value reflects the **Alternate Views DeMorgan** checkbox in the P-CAD Library Manager Edit Attach Symbols dialog.

Used By
alts

depth (PCB)

depth ::= '(' 'depth'
integerToken
')'

Description
depth classifies a copper pour vertex based on its relationship to the poured area. *depth* signifies whether this vertex begins a segment outside of the pour (1), no change (0), or begins a segment entering the pour (-1) .

Used By
vertex

desc

desc ::= '(' 'desc'
stringToken
')'

Description
desc is a text string providing descriptive information or an item mnemonic.

Used By
apertureDef, infoPoint

designAuthor

designAuthor ::= '(' 'designAuthor'
stringToken
')'

Description
designAuthor represents the contents of the **Author** field in the Design Info dialog box.

Notes
This production is no longer generated by P-CAD, but is included for compatibility with older designs. Use of the *fieldDef* production to define field names and values is preferred.

Used By
designInfo

designDate

designDate ::= '(' 'designDate'
stringToken
')'

Description

designDate represents the contents of the **Date** field in the Design Info dialog box.

Notes

This production is no longer generated by P-CAD, but is included for compatibility with older designs. Use of the *fieldDef* production to define field names and values is preferred.

Used By

designInfo

designInfo

designInfo ::= '(' 'designInfo'
{ fieldSet }
[designTitle]
[designAuthor]
[designDate]
[designTime]
[designRevision]
')'

Description

designInfo lists the contents of the Design Info dialog fields. The *fieldSet* production is the preferred way to designate field values, although the older *designTitle, designAuthor, designDate, designTime,* and *designRevision* productions are retained for compatibility.

Used By

pcbDesignHeader, schematicDesignHeader

designRegion

designRegion ::= '(' 'designRegion'
pt
pt
')'

Description

designRegion describes a specified area to be drawn. The first point describes the lower left corner of a rectangle, the second point describes the opposing corner.

Used By

designView

designRevision

designRevision ::= '(' 'designRevision'
stringToken
')'

Description

designRevision represents the contents of the **Revision** field in the Design Info dialog box.

Notes

This production is no longer generated by P-CAD, but is included for compatibility with older designs. Use of the *fieldDef* production to define field names and values is preferred.

Used By

designInfo

designTime

designTime ::= '(' 'designTime'
stringToken
')'

Description
designTime represents the contents of the **Time** field in the Design Info dialog box.

Notes
This production is no longer generated by P-CAD, but is included for compatibility with older designs. Use of the *fieldDef* production to define field names and values is preferred.

Used By
designInfo

designTitle

designTitle ::= '(' 'designTitle'
stringToken
')'

Description
designTitle represents the contents of the **Title** field in the Design Info dialog box.

Notes
This production is no longer generated by P-CAD, but is included for compatibility with older designs. Use of the *fieldDef* production to define field names and values is preferred.

Used By
designInfo

designView

designView ::= '(' 'designView'
location
textStyleRef
scaleFactor
title
[subTitle]
designRegion
layerList
')'

Description
designView represents a visual reproduction which can be scaled to aid in showing a more detailed depiction of a particular area.

Notes
DesignView objects can only be created by P-CAD Document Toolbox.

Used By
layerContent

detail (PCB)

detail ::= '(' 'detail'
filename
location
[extent]
title
subtitle
textStyleRef

metafile
')'

Description
detail is used to designate a Detail object.

Notes
Detail objects can only be created by P-CAD Document Toolbox.

Used By
pcbDrawObj

diagram (PCB)

diagram ::= '(' 'diagram'
diagramType
location
[extent]
[scaleFactor]
title
subtitle
textStyleRef
[diagramInfo]
')'

Description
diagram is used to designate a Diagram object.

Notes
Diagram objects can only be created by P-CAD Document Toolbox.

Used By
pcbDrawObj

diagramInfo

diagramInfo ::= layerStackupInfo

Description
diagramInfo is used to add diagram-specific information to a *diagram* production.

Notes
At this time, layer stackup diagrams are the only supported diagram type.

Used By
diagram

diagramType (PCB)

diagramType ::= 'layerStackup'

Description
diagramType is used to specify the type of diagram in a *diagram* production.

Notes
At this time, layer stackup diagrams are the only supported diagram type.

Used By
diagram

dimension (PCB)

dimension::= '(' 'dimension'
dimensionStyle
pt
rotation
isFlipped
dimensionOrient
dimensionTextOrient
dimensionPrecision
dimensionDisplayUnits
dimensionSuppressLeadingZeros
dimensionSuppressTrailingZeros
dimensionUnits
dimensionLineWidth
dimensionLeaderStyle
dimensionLeaderSize
dimensionCenterSize
dimensionPlusLinearTol
dimensionMinusLinearTol
dimensionPlusDegTol
dimensionMinusDegTol
dimensionShowTol
dimensionShowDiaSymbol
dimensionDimLineGraphics
dimensionExtLineGraphics
dimensionTextGraphics
dimensionPoints
[arrowheadWidth]
[arrowheadLength]
[dimensionID]
[dimensionOffsets]
')'

Description
dimension contains the data that describes a dimension object. Its style, location, rotation, etc, and the graphical objects.

Used By
pcbDrawObj

dimensionCenterSize (PCB)

dimensionCenterSize::= '('
numberToken
')'

Description
dimensionCenterSize descibes the width and height of the center dimension.

Used By
dimension

dimensionDimLineGraphics (PCB)

dimensionDimLineGraphics ::= '('
dimensionDimLineGraphicsList
')'

Description
dimensionDimLineGraphics is a two dimensional array of graphical objects for displaying the dimension lines.

Used By
dimension

dimensionDimLineGraphicsList (PCB)

dimensionDimLineGraphicsList ::= '('
NIL
| dimensionGraphics
| dimensionDimLineGraphicsList
')'

Description
dimensionDimLineGraphicsList is an array of graphical objects for displaying the dimension lines.

Used By
dimensionDimLineGraphics

dimensionDisplayUnits (PCB)

dimensionDisplayUnits::= '('
booleanToken
')'

Description
dimensionDisplayUnits is a flag for displaying the units of measurement with the text.

Used By
dimension

dimensionSuppressLeadingZeros (PCB)

dimensionSuppressLeadingZeros::= '('
booleanToken
')'

Description
dimensionSuppressLeadingZeros is a flag for suppress the leading zeros for the dimension text.

Used By
dimension

dimensionSuppressTrailingZeros (PCB)

dimensionSuppressTrailingZeros::= '('
booleanToken
')'

Description
dimensionSuppressTrailingZeros is a flag for suppress the trailing zeros for the dimension text.

Used By
dimension

dimensionExtLineGraphics (PCB)

dimensionExtLineGraphics ::='('
dimensionExtLineGraphicsList
')'

Description

dimensionExtLineGraphics is a two dimensional array of graphical objects for displaying the extension lines.

Used By

dimension

dimensionExtLineGraphicsList (PCB)

dimensionExtLineGraphicsList ::= '('
NIL
| dimensionGraphics
| dimensionExtLineGraphicsList
')'

Description

dimensionExtLineGraphicsList is an array of graphical objects for displaying the extension lines.

Used By

dimensionExtLineGraphics

dimensionGraphic (PCB)

dimensionGraphic ::= '('
| line
| arc
| poly
| tangoText
| triplePointArc
')'

Description

dimensionGraphic is a graphical object such as a line, polygon, arc, or text.

dimensionId (PCB)

dimensionId ::= '(' 'dimensionID'
integerToken
')'

Description

dimensionID is used to designate a unique identifier for a *dimension* object. This identifier is used to maintain information about associative dimensions.

Notes

Associative dimensioning is only available in P-CAD Document Toolbox.

Used By

dimension

dimensionIDRef (PCB)

dimensionIDRef ::= integerToken

Description

dimensionIDRef is used to refer to an identifier for a *dimension* object.

Notes

Associative dimensioning is only available in P-CAD Document Toolbox.

Used By

dimensionRef

dimensionIndex (PCB)

dimensionIndex ::= integerToken

Description
dimensionIndex specifies the index for an object associated with a *dimension*.

Notes
Associative dimensioning is only available in P-CAD Document Toolbox.

Used By
dimensionRef

dimensionLeaderSize (PCB)

dimensionLeaderSize::= '('
numberToken
')'

Description
dimensionLeaderSize descibes the width and height of the leader dimension symbol.

Used By
dimension

dimensionLeaderStyle (PCB)

dimensionLeaderStyle::= '('
'dim_leader_style_text' |'dim_leader_style_circle' | 'dim_leader_style_square' | 'dim_leader_style_triangle'
')'

Description
dimensionLeaderStyle descibes which style a leader dimension is.

Used By
dimension

dimensionLineWidth (PCB)

dimensionLineWidth::= '('
numberToken
')'

Description
dimensionLineWidth descibes the width of the lines in the dimension.

Used By
dimension

dimensionMinusDegTol (PCB)

dimensionMinusDegTol ::= '('
numberToken
')'

Description
dimensionMinusDegTol descibes the tolerance of an angular dimension.

Used By
dimension

dimensionMinusLinearTol (PCB)

dimensionMinusLinearTol ::= '('
numberToken
')'

Description
dimensionMinusLinearTol descibes the tolerance of a linear dimension.

Used By
dimension

dimensionOffsets (PCB)

dimensionOffsets ::= '(' 'dimensionOffsets'
{ pt }
')'

Description
dimensionOffsets is used to store the offsets for a dimension.

Used By
dimension

dimensionOrient (PCB)

dimensionOrient::= '('
('dimension_horizontal' | 'dimension_vertical' |'dimension_diagonal')
')'

Description
dimensionOrient descibes the orientation of the dimension, either horizontal, vertical, or diagonal.

Used By
dimension

dimensionPlusDegTol (PCB)

dimensionPlusDegTol ::= '(' numberToken ')'

Description
dimensionPlusDegTol descibes the tolerance of an angular dimension.

Used By
dimension

dimensionPlusLinearTol (PCB)

dimensionPlusLinearTol ::= '('
numberToken
')'

Description
dimensionPlusLinearTol descibes the tolerance of a linear dimension.

Used By
dimension

dimensionPoints (PCB)

dimensionPoints ::= '('

dimensionPointsList
')'

Description
dimensionTextGraphics is an array of *pt* that are the points of measurement for the dimension.

Used By
dimension

dimensionPrecision (PCB)

dimensionPrecision ::= '('
integerToken
')'

Description
dimensionPrecision descibes the precision to which the text is displayed.

Used By
dimension, drillTableInfo

dimensionRef (PCB)

dimensionRef ::= '('
dimensionIDRef
dimensionIndex
')'

Description
dimensionRef is used to refer to a specific dimension. It is used to associate a graphical object with a dimension object.

Notes
Associative dimensioning is available only in P-CAD Document Toolbox.

Used By
arc, line, pad, pattern, via

dimensionShowDia (PCB)

dimensionShowDia ::= '('
booleanToken
')'

Description
dimensionShowDia is a flag for displaying the diameter symbol.

Used By
dimension

dimensionShowTol (PCB)

dimensionShowTol ::= '('
booleanToken
')'

Description
dimensionShowTol is a flag for displaying the dimension tolerance.

Used By
dimension

dimensionStyle (PCB)

dimensionStyle::= '('
('dim_angular' | 'dim_baseline' | 'dim_center' | 'dim_diameter' | 'dim_leader' |
'dim_pointtopoint' | 'dim_radius | dim_datum')
')'

Description
dimensionStyle descibes which style the dimension is.

Used By
dimension

dimensionTextGraphics (PCB)

dimensionTextGraphics ::= '('
dimensionTextGraphicsList
')'

Description
dimensionTextGraphics is a two dimensional array of graphical objects for displaying the text of the dimension.

Used By
dimension

dimensionTextGraphicsList (PCB)

dimensionTextGraphicsList ::= '('
| NIL
| dimensionTextGraphicsList
| dimensionGraphic
')'

Description
dimensionTextGraphicsList is an array of graphical objects for displaying the text of the dimension.

Used By
dimensionTextGraphics

dimensionTextOrient (PCB)

dimensionTextOrient::= '('
('dimension_horizontal' | 'dimension_vertical' | 'dimension_diagonal')
')'

Description
dimensionTextOrient descibes the orientation of the text within the dimension.

Used By
dimension

dimensionUnits (PCB)

dimensionUnits::= '(' 'dimension_units_inch' | 'dimension_units_mil' | 'dimension_units_mm' |
'dimension_units_cm' | 'dimension_units_cm_inch' |'dimension_units_cm_mil' |
'dimension_units_inch_cm' |'dimension_units_inch_mm' | 'dimension_units_mil_mm' |
'dimension_units_mm_inch' |'dimension_units_mm_mil'
')'

Description
dimensionDisplayUnits is a flag for displaying the units of measurement with the text.

Used By
dimension

dispName (SCH)

dispName ::= '(' 'dispName'
booleanToken
')'

Description
dispName indicates whether a bus or wire name is to be displayed or hidden. If the value is True, the name is displayed. This value reflects the current **Display Name** check box setting in the Modify Bus and Modify Wire dialogs.

Used By
bus, wire

dispPinDes (SCH)

dispPinDes ::= '(' 'dispPinDes'
booleanToken
')'

Description
dispPinDes indicates whether a component pin designator is to be displayed or hidden. If the value is True, the designator is displayed. This value reflects the current **Display Pin Designator** check box setting in the Modify Part dialog.

Used By
pinDisplay

dispPinName (SCH)

dispPinName ::= '(' 'dispPinName'
booleanToken
')'

Description
dispPinName indicates whether a component pin name is to be displayed or hidden. If the value is True, the name is displayed. This value reflects the current **Display Pin Name** check box setting in the Modify Part dialog.

Used By
pinDisplay

drawBorder (SCH)

drawBorder ::= '(' 'drawBorder'
booleanToken
')'

Description
drawBorder indicates whether the title sheet is to be included in the print output. This value reflects the current **Title** check box setting in the Print Page Setup dialog.

Used By
schematicPrintSettings

drawPolygons (PCB)

drawPolygons ::= '(' 'drawPolygons'

booleanToken
')'

Description
drawPolygons indicates whether polygon shaped pads and vias should be drawn or flashed when they are written as Gerber data.

Used By
gerberSettings

drawRotated (PCB)

drawRotated ::= '(' 'drawRotated'
booleanToken
')'

Description
drawRotated indicates whether the image is to be rotated 90 degrees in the Gerber output. This value reflects the current **Rotate** check box setting in the Aperture Assignment dialog.

Used By
gerberSettings

drillSym (PCB)

drillSym ::= '(' 'drillSym'
(drillSymShape | stringToken)
holeDiam
[isHolePlated]
')'

Description
drillSym associates holes in a PCB design with a drill symbol shape or a single alphabetic character label. A hole is identified by diameter and plating characteristic. These symbols are used for Printed, DXF, and Gerber Output, and reflect the assignments in the Drill Symbol Assignments dialog.

Defaults
isHolePlated: True

Used By
drillSymSettings

drillSymColor (PCB)

drillSymColor ::= '(' 'drillSymColor'
stringToken
')'

Description
drillSymColor is used to specify a color for drill symbol display configurations. The color is used for drawing drill symbols in the PCB workspace when the drill symbol configuration is assigned to a non-signal layer.

Used By
drillSymDisplayConfigDef

drillSymDisplayConfigDef (PCB)

drillSymDisplayConfigDef ::= '(' 'drillSymDisplayConfigDef'
nameDef
drillSymSize
outputDrillSymPlated

outputDrillSymNonplated
drillSymColor
outputPads
outputVias
startRange
endRange
')'

Description

drillSymDisplayConfigDef defines a display configuration for drill symbols, specifying the manner in which drill symbols are to be displayed in the PCB workspace. When a *drillSymDisplayConfigDef* is referenced by name in a non-signal PCB *layerDef,* drill symbols are displayed when the layer is enabled.

Used By

drillSymSettings

drillSymDisplayConfigRef (PCB)

drillSymDisplayConfigRef ::= '(' 'drillSymDisplayConfigRef'
nameRef
')'

Description

drillSymDisplayConfigRef is used to assign a *drillSymDisplayConfigDef* to a non-signal PCB *layerDef.*

Used By

layerDef

drillSymSettings (PCB)

drillSymSettings ::= '(' 'drillSymSettings'
{ drillSym }
{ drillSymDisplayConfigDef }
')'

Description

drillSymSettings lists the PCB design drill symbol assignments and drill symbol display configurations. The drill symbol assignments are used for Gerber output, print output, and drill symbol display in the PCB workspace. Drill symbol assignments are reflected in the Drill Symbol Assignments dialog, while drill symbol display configurations are reflected in the Drill Symbol Display Configurations dialog.

Used By

pcbDesign

drillSymShape (PCB)

drillSymShape ::= ('Cross' | 'X' | 'Y' | 'T' | 'Hour' | 'Side_Hour' |'Box_Line' | 'Diamond_Line' |
'Box_V' |'Diamond_V' | 'Box_X' | 'Diamond_Cross' |'Box_Cross' | 'Diamond_X' | 'Box_Y' | 'Diamond_Y' |'Box_T' |
'Diamond_T' | 'Circle_Line' |'Circle_V' | 'Circle_Cross' | 'Circle_X' |
'Circle_Y' | 'Circle_T')

Description

drillSymShape indicates display shape to be used during Gerber and print output for a drill symbol.

Used By

drillSym

drillSymSize (PCB)

drillSymSize ::= '(' 'drillSymSize'

dbNumber
')'

Description
drillSymSize defines the size of the drill symbols for printer or CAM output.

Used By
outputItem

drillTableInfo (PCB)

drillTableInfo ::= '(' 'drillTableInfo'
units
[dimensionPrecision]
{ column }
')'

Description
drillTableInfo is used to specify information specific to a drill table. This information includes the units used by the table, places of precision to right of decimal, and an optional list of user-defined columns.

Defaults
dimensionPrecision: 3

Used By
tableInfo

ecoRecording

ecoRecording ::= '(' 'ecoRecording'
booleanToken
')'

Description
ecoRecording indicates the status of the **ECO Recorder** radio button in the Utils Record ECOs dialog. True indicates ECO recording is ON; False indicates recording is OFF.

Used By
ecoState

ecoState

ecoState ::= '(' 'ecoState'
ecoRecording
')'

Description
ecoState indicates the status of the ECO recorder.

Used By
programState

embedApertures (PCB)

embedApertures ::= '(' 'embedApertures'
booleanToken
')'

Description

embedApertures indicates whether Gerber output is to include embedded aperture definitions as mass parameters. A value of True specifies that the embedded apertures are to be included, as indicated by the **Include Aperture Definitions** check box in theGerber Format dialog.

Used By
gerberSettings

enhancedPolygon (PCB)

enhancedPolygon ::= '(' enhanced'Polygon'
{ polyPoint }
')'

Description
enhancedPolygon is a definition of polygon that contains straight and curved edges.

Used By
boardCutoutObj
boardOutlineObj

endPoint

endPoint ::= location

Description
endPoint is a terminal point locatoin.

Used By
triplePointArc

endRange (PCB)

endRange::= integerToken

Description
endRange represents the layer a pad/via's hole range end on. This is specified in the Modify Hole Range dialog..

Default value
LAYER_BOTTOM_SIGNAL

Used By
padStyleDef, viaStyleDef

endStyle

endStyle::= '(' 'endStyle' ('LeftLead' | 'RightLead' | 'Rounded' | 'TwoLeads') ')'

Description
endStyle represents a line's end point *endStyle*. It can be set to one of four values, *LeftLead*, *RightLead*, *Rounded*, or *TwoLeads*. If a line's end point *endStyle* is missing, the end style is assumed to be *Rounded.*

Default value
Rounded

Used By
line

entireDesign

entireDesign ::= '(' 'entireDesign'
booleanToken

')'

Description

entireDesign indicates whether the printing of a job or sheet should include the extents of the job or sheet, or only the defined region for the job or sheet, in the output.

Used By

sheet, printQueueEntry

entryName (PCB)

entryName ::= stringToken

Description

entryName represents the name given a either a Print, Gerber, or N/C Drill queue output job. This is the user-specified job or file name assigned to an output job in the File Print, Gerber Out, or N/C Drill dialogs.

Used By

camQueueEntry, printQueueEntry

extent

extent ::= '(' 'extent'
xPoint
yPoint
')'

Description

extent designates the size of a bounding box. It is written for informational purposes only, and is ignored when read from an input file.

Used By

metafile, table, text

field

field ::= '(' 'field'
fieldType
location
[rotation]
[isFlipped]
[justify]
[textStyleRef]
')'

Description

field represents a field object placed on a PCB or Schematic design. A *field* is defined by its type and location.

Notes

The default text style is used if *textStyleRef* is not present.

Defaults

rotation: 0 degrees
isFlipped: False
justify: JUSTIFY_LOWER_LEFT

Used By

pcbDrawObj, schDrawObj

fieldDef

fieldDef ::= '(' 'fieldDef'
fieldNameDef
fieldValue
')'

Description
fieldDef represents the logical value for a field in P-CAD, as defined in the Design Info dialog. This production is used for both predefined and user-defined field definitions.

Used By
fieldSet

fieldNameDef

fieldNameDef ::= nameDef

Description
fieldNameDef defines the name of a field definition.

Used By
fieldDef

fieldNameRef

fieldNameRef ::= nameRef

Description
fieldNameRef refers to an existing field.

Used By
fieldType

fieldSet

fieldSet ::= '(' 'fieldSet'
fieldSetNameDef
{ fieldDef }
{ noteDef }
{ revisionNoteDef }
')'

Description
fieldSet groups together field definitions, note definitions, and revision note definitions for a specific field set.

Notes
For files generated by P-CAD without the Document Toolbox option, all fields will be grouped into a single field set titled "(Default)."

Used By
designInfo

fieldSetNameDef

fieldSetNameDef ::= nameDef

Description
fieldSetNameDef defines the name of a field set. This name reflects the contents of the **Field Set Name** edit box in the Field Set dialog.

Notes
The Field Set dialog is only available with P-CAD Document Toolbox.

Used By
fieldSet

fieldSetNameRef

fieldSetNameRef ::= nameRef

Description
fieldSetNameRef refers to the name of a previously-defined field set.

Used By
fieldSetRef

fieldSetRef

fieldSetRef ::= '(' 'fieldSetRef'
fieldSetNameRef
')'

Description
fieldSetRef refers to a previously-defined field set.

Used By
layerDef, sheet

fieldType

fieldType ::= ('Date' | 'CurDate' | 'Time' | 'CurTime' | 'Author' | 'Rev' | 'Filename' | 'Title' |
'SheetNumber' | 'NumSheets'
| noteRef
| revisionNoteRef
| fieldNameRef)

Description
fieldType indicates one of the P-CAD predefined field types, or a user-defined field type. The preferred way to represent the type of a field is with *fieldNameRef*, instead of the enumerated values listed above, although these values are still valid for compatibility with older files.

Used By
field

fieldValue

fieldValue ::= stringToken

Description
fieldValue defines the value of a field definition.

Used By
fieldDef

fileAuthor

fileAuthor ::= '(' 'fileAuthor'
stringToken
')'

Description
fileAuthor represents the contents of the **Author** edit box in the Design Info dialog.

Used By

written

filename

filename ::= '(' 'filename'
stringToken
')'

Description
filename represents the name of an external file.

Used By
detail

fileUnits

fileUnits ::= '(' 'fileUnits'
dbUnit
')'

Description
fileUnits specifies the default database unit for all coordinate and size output values.

Used By
asciiHeader

See Also
dbNumber

fillets (PCB)

fillets ::= '(' 'fillets'
chordHeight
{ filletDesc }
')'

Description
fillets starts a block of fillet information used in the routing of polygon corners.

Used By
pcbPoly

filletDesc (PCB)

filletDesc ::= '(' 'filletDesc'
location
radius
')'

Description
filletDesc defines a single polygon vertex fillet. There is one *filletDesc* for every vertex in the polygon.

Used By
fillet

fillPoly (PCB)

fillPoly ::= '(' 'fillPoly'
rotation
{ poly }

')'

Description

fillPoly represents the boundary of an object either wholly or partially contained within a copper pour outline. *fillPoly* is present only for poured copper pours. The polygon is defined to reflect the current pour backoff values and may not be coincident with the object boundaries.

Used By

copperPour

first (PCB)

first ::= '(' 'first'
booleanToken
')'

Description

first indicates that a copper pour vertex begins a new vertex list.

Used By

vertex

font

font ::= '(' 'font'
fontType
fontFamily
fontFace
fontHeight
strokeWidth
[fontWeight]
[fontItalic]
[fontCharSet]
[fontOutPrecision]
[fontClipPrecision]
[fontQuality]
[fontPitchAndFamily]
')'

Description

font describes a text font to be used as part of a text style definition. The font definition includes the type of font, font family, the font face, height, and width. Character widths are defined individually per character and are maintained within the font file for stroke fonts; only font height is specified within the font definition. Font weight, whether or not a font is italized, font character set, font output precision, font clip precision, font quality, and font pitch and family is specified when using a TrueType font.

Defaults

fontItalic: False

Used By

textStyleDef

fontCharSet

fontCharSet ::= '(' 'fontCharSet'
integerToken
')'

Description

fontCharSet defines the type of character set (e.g. ANSI, Symbol, etc.) to use for TrueType fonts.

Used By
font

fontClipPrecision

fontClipPrecision ::= '(' 'fontClipPresision'
integerToken
')'

Description
fontClipPrecision defines how to clip characters that are partially outside the clipping region.

Used By
font

fontFace

fontFace ::= '(' 'fontFace'
stringToken
')'

Description
fontFace specifies the name of an P-CAD design font or a TrueType font.

Used By
font

fontFamily

fontFamily ::= '(' 'fontFamily'
('Serif' | 'Sanserif' | 'Modern')
')'

Description
fontFamily denotes the font family for a font. P-CAD currently supports Serif and Sanserif families. P-CAD supports Modern for TrueType fonts.

Used By
font

fontHeight

fontHeight ::= '(' 'fontHeight'
dbNumber
')'

Description
fontHeight specifies the height of a text font. Note that character widths are defined individually per character and are maintained within the font file; only font height is specified within the font definition.

Used By
font

fontItalic

fontItalic::= '(' 'fontItalic'
booleanToken
')'

Description

fontItalic, if set to TRUE, states that the text represented in this font will be italized.

Default
FALSE

Used By
font

fontOutPrecision

fontOutPrecision::= '(' 'fontOutPrecision'
integerToken
')'

Description
fontOutPrecision defines how closely the output must match the height, width, character orientation, escapement, and pitch of the requested font.

Used By
font

fontPitchAndFamily

fontPitchAndFamily ::= '(' 'fontPitchAndFamily'
integerToken
')'

Description
fontPitchAndFamily specifies the pitch and family of the font. The pitch desribes the slant of the text represted in this font. The font family describes the look of a font in a general way.

Used By
font

fontQuality

fontQuality ::= '(' 'fontQuality'
integerToken
')'

Description
fontQuality defines how carefully the graphics device interface (GDI) must attempt to match the specified font characteristics to the actual physical font.

Used By
font

fontType

fontType ::= '(' 'fontType'
('Stroke' | 'TrueType')
')'

Description
fontType indicates a font type. Currently Stroke and TrueType fonts are supported.

Used By
font

fontWeight

fontWeight ::= '(' 'fontWeight '
integerToken
')'

Description
fontWeight defines weight or thickness of a font. The weight can be thin, normal, bold, heavy, etc.

Used By
font

fromTo (PCB)

fromTo ::= '(' 'fromTo'
netNameRef
oneEnd
anotherEnd
')'

Description
fromTo represents an electrical connection between two PCB objects. The definition includes the locations of those objects, and the name of the net to which the connection belongs.

Used By
pcbDrawObj

g54Option (PCB)

g54Option ::= '(' 'g54Option'
booleanToken
')'

Description
g54Option indicates whether Gerber output is to include a G54 tool-select code with each command to change apertures. A value of True specifies that the codes are to be included, as indicated by the **G54 With Apertures** check box in the Gerber Format dialog.

Used By
gerberSettings

gateEq

gateEq ::= '(' 'gateEq'
integerToken
')'

Description
gateEq is an integer value indicating the equivalence value assigned to a pin and its gate. This value reflects the **gateEq** column in the P-CAD Library Manager spreadsheet for the pin.

Notes
gateEq is used for gate swapping if the **Automatic** option is checked in the Optimize Nets dialog. It is also used by the Renumber command, and for RefDes numbering during part placement.

Used By
compPin

gerberSettings (PCB)

gerberSettings ::= '(' 'gerberSettings'
units

numFormat
autoDrawApertureSize
outputPath
viewLog
autoClear
g54Option
useArcs
embedApertures
useApertureHoles
drawRotated
[drawPolygons]
{ apertureDef }
{ apertureAssn }
{ camQueueEntry }
')'

Description

gerberSettings describes the state of the Gerber output settings and lists the Gerber output queue entries for a design. The settings and queue entries are specified in the File Gerber Out dialog.

Defaults

drawPolygons: True

Used By

pcbDesign

globalAttrs

globalAttrs ::= '(' 'globalAttrs'
{ attr }
')'

Description

globalAttrs lists all design level attributes for a Schematic or PCB design. These attributes will have been created using Place Attributes to place an attribute on the sheet or board, or by adding an attribute through the Design Info Attributes dialog.

Used By

pcbDesign, schematicDesign, netList

globalCopperPourCutoutBackoffFlag

globalCopperPourCutoutBackoffFlag ::= '(' 'globalCopperPourCutoutBackoffFlag'
booleanToken
')'

Description

globalCopperPourCutoutBackoffFlag specifies whether the copper pours in the design should back away from the embedded cutouts to exactly the edge of the cutouts. A True means back away to exactly the edge, False means use previous behavior, which was to overlap the cutout by half the thickness of the copper line width.

Used By

pcbDesignHeader

gluePoint (PCB)

gluePoint ::= '(' 'gluepoint'
location
[isFlipped]
[isVisible]

')'

Description

gluepoint represents a gluepoint object. The definition includes the gluepoint location.

Defaults

isFlipped: False
isVisible: True

Used By

pcbDrawObj

gluePointSize

gluePointSize ::= '(' 'gluePointSize'
dbNumber
')'

Description

gluePointSize describes the size of glue point.

Used By

pcbDesignHeader

gluePointSizePrint

gluePointSizePrint ::= '(' 'gluePointSizePrint'
dbNumber
')'

Description

gluePointSizePrint describes the print size of glue point.

Used By

pcbDesignHeader

grid

grid ::= '(' 'grid'
gridString
')'

Description

grid describes a single grid value defined for a design. The string includes the grid units. This value reflects the string listed for a grid in the Options Grids dialog.

Notes

A PCB design grid may have multiple values within a single grid string.

Used By

gridDfns

gridDfns

gridDfns ::= '(' 'gridDfns'
[relOrigin]
{ grid }
')'

Description

gridDfns lists the grids defined for a design. The grid definitions will include a relative origin if one has been defined. These grids and the relative origin are specified in the Options Grids dialog.

Defaults

relOrigin: (0,0)

Used By

pcbDesignHeader, schematicDesignHeader

gridState

gridState ::= '(' 'gridState'
currentAbsGrid
currentRelGrid
[currentViaGrid]
isAbsoluteGrid
isDottedGrid
isVisibleGrid
isPromptForRel
[viaGridVisibility]
[plowGrid]
[plowViaGrid]
')'

Description

gridState describes the state of the grid settings for a design. These values are set in the Options Grids dialog.

Notes

currentViaGrid is written for PCB only. *plowViaGrid* and *plowGrid* are obsolete and ignored.

Used By

programState

gridString

gridString ::= stringToken

Description

gridString describes a grid value. The string includes the grid spacing, and the grid units.

Notes

A PCB design grid may have multiple values within a single grid string.

Used By

currentAbsGrid, currentRelGrid, currentViaGrid, grid

headerString

headerString ::= '(' 'headerString'
stringToken
')'

Description

headerString is an optional string that may be added to a design file header. It is read and ignored by P-CAD programs during the file open operation.

Used By

asciiHeader

height

height ::= '(' 'height'
dbNumber
')'

Description
height describes the height of an object.

Used By
apertureDef, border, ieeeSymbol

hexToken (PCB)

hexToken

Description
hexToken represents a 4-byte (32-bit) number in hexadecimal format. This number is designated by a '0x' prefix followed by 8 hexadecimal digits, i.e. 0x012345AB.

Used By
binaryData

holeDiam (PCB)

holeDiam ::= '(' 'holeDiam'
dbNumber
')'

Description
holeDiam defines the hole diameter of an object.

Used By
apertureDef, drillSym, padStyleDef, toolAssn, viaStyleDef

holeOffset (PCB)

holeOffset ::= '(' 'holeOffset'
xPoint
yPoint
')'

Description
holeOffset describes the distances, in the x and y directions, a pad or via style hole is offset from the pad or via center.

Used By
padStyleDef, viaStyleDef

horizontalZones

horizontalZones ::= '(' 'horizontalZones'
integerToken
numDirection
numType
)'

Description
Describes the horizontal zoning information for a title sheet.

Used By
zones

hourNumber

hourNumber ::= integerToken

Description
hourNumber is an integer number ranging from 0-23 that represents a specific hour of a 24 hour day.

Used By
time

ieeeAlt

ieeeAlt ::= '(' 'ieeeAlt'
booleanToken
')'

Description
ieeeAlt indicates whether a component definition includes IEEE symbol representations. This value reflects the **Alternate Views IEEE** check box in the P-CAD Library Manager Edit Attach Symbols dialog.

Used By
alts

ieeeSymbol (SCH)

ieeeSymbol ::= '(' 'ieeeSymbol'
ieeeSymbolType
location
height
[rotation]
[isFlipped]
')'

Description
ieeeSymbol represents an IEEE symbol object.

Defaults
rotation: 0 degrees
isFlipped: False

Used By
schDrawObj

ieeeSymbolType (SCH)

ieeeSymbolType ::= ('Adder' | *'Amplifier'* | *'Astable'*| *'Complex'* | 'Generator' | 'Hysteresis' |
'Multiplier')

Description
ieeeSymbolType denotes the type of an IEEE symbol.

Used By
ieeeSymbol

infoPoint

infoPoint ::= '(' 'infoPoint'
location
infoPointViolationNumber
[severity]
[clearance]

desc
[infoPointRuleCategory]
[infoPointRuleType]
[infoPointViolationType]
')'

Description
infoPoint represents a DRC or ERC dot indicator.

Notes
clearance and severity were used upto V14 only in Pcb. Clearance& Severity values in an old design file are read but ignored.

Defaults
clearance: 0
severity: 0

Used By
pcbDrawObj, schDrawObj

infoPointRuleCategory

infoPointRuleCategory ::= '(' 'infoPointRuleCategory'
stringToken
')'

Description
infoPointRuleCategory represents the name of a rule category.

Notes
infoPointRuleCategory is new from V15. For the old design, this will be set to the default value

Defaults
Unknown Rule Categroy

Used By
infoPoint

infoPointRuleType

infoPointRuleType ::= '(' 'infoPointRuleType'
stringToken
')'

Description
infoPointRuleType represents the name of a rule type.

Notes
infoPointRuleType is new from V15. For the old design, this will be set to the default value

Defaults
Unknown Rule Type

Used By
infoPoint

infoPointSize

infoPointSize ::= '(' 'infoPointSize'

dbNumber
')'

Description
infoPointSize describes the size of info point.

Used By
pcbDesignHeader

infoPointSizePrint

infoPointSizePrint ::= '(' 'infoPointSizePrint'
dbNumber
')'

Description
infoPointSizePrint describes the print size of info point.

Used By
pcbDesignHeader

infoPointViolationNumber

infoPointViolatonnNumber ::= '(' 'number'
integerToken
')'

Description
infoPointViolatonnNumber designates a unique identifier for an info point (DRC/ERC dot).

Used By
InfoPoint

infoPointViolationType

infoPointViolationType::= '(' infoPointViolationType 'errorViolation'|'warningViolation' |'ignoredViolation' |'acceptedViolation' ')'

Description
infoPointViolationType indicates the Violation Type of the infoPoint

Notes
infoPointViolationType is new from V15. For the old design, this will be set to the default value

Defaults
errorViolation.

Used By
infoPoint

insideDiam (PCB)

insideDiam ::= '(' 'insideDiam'
dbNumber
')'

Description
insideDiam denotes the inside diameter of an object.

Used By

shapeHeight

insideEdgeStyle (SCH)

insideEdgeStyle ::= '(' 'insideEdgeStyle'
('None' | 'Clock')
')'

Description
insideEdgeStyle defines the display characteristic to be associated with the inside edge of a component pin. This value will reflect the **Inside Edge** combo box setting for the pin in the Place Pin or Modify Pin dialogs.

Used By
pin

insideStyle

insideStyle ::= '(' 'insideStyle'
('None' | 'Open' | 'OpenHigh' | 'OpenLow' | 'PassiveUp' |'PassiveDown' | 'ThreeState' | 'Amplifier' | 'Generator' |'Hysteresis' | 'Postponed' | 'Shift')
')'

Description
insideStyle defines the display characteristic to be associated with the inside of a component pin. This value will reflect the **Inside** combo box setting for the pin in the Place Pin or Modify Pin dialogs.

Used By
pin

integerToken

integerToken

Description
integerToken represents a whole number, that is, a non-fractional number that may be positive, negative, or zero.

Used By
many productions

isAbsoluteGrid

isAbsoluteGrid ::= '(' 'isAbsoluteGrid'
booleanToken
')'

Description
isAbsoluteGrid indicates if a grid is defined relative to the relative grid origin, or has its origin at the design (0,0) origin. A value of True indicates the grid is relative to the design origin; False indicates the grid is relative to **the Relative Grid Origin**. This value and the relative grid origin reflect the current settings in the Options Grid dialog.

Used By
gridState

isAutoSwapPatternGraphics

isAutoSwapPatternGraphics ::= '(' 'isAutoSwapPatternGraphics'
booleanToken
')'

Description

isAutoSwapPatternGraphics indicates whether a component pattern automatically modifies its graphics when its orientation is modified.

Used By

Pattern

See Also

patternDefExtended, patternOrientationsMap

isCopperTie (PCB)

isCopperTie ::= '(' 'isCopperTie'
booleanToken
')'

Description

isCopperTie indicates whether a PCB polygon design object is being used to tie two or more nets together. Nets are tied together when they are assigned the same *tieNetValue*.

Used By

poly

isDottedGrid

isDottedGrid ::= '(' 'isDottedGrid'
booleanToken
')'

Description

isDottedGrid specifies how a grid is to be displayed. If True, the grid is defined as a dotted grid; if False, the grid is displayed as a hatched grid. This value will reflect the current **Visible Grid Style** setting for this grid in the Options Grids dialog.

Used By

gridState

isDraft (PCB)

isDraft ::= '(' 'isDraft'
booleanToken
')'

Description

isDraft indicates if a print queue entry is to be printed in draft mode. This value reflects the status of the **Draft Mode** check box in the Print Options dialog.

Used By

printQueueEntry

isFixed

isFixed ::= '(' 'isFixed''
booleanToken
')'

Description

isFixed indicates whether a component's location is fixed. Fixed components cannot be moved, rotated, flipped, or moved.

Used By

pattern, room

isFlipped

isFlipped ::= '(' 'isFlipped'
booleanToken
')'

Description
isFlipped indicates whether an object has been flipped. P-CAD objects are flipped by rotating the object 180 degrees about the y-axis of a local coordinate system whose origin is the object origin or reference point and whose axes parallel the design x-y axes.

Used By
arc, attr, dimension, field, gluePoint, ieeeSymbol, line, pad, pattern, pickPoint, pin, poly, symbol, text, via

isHolePlated (PCB)

isHolePlated ::= '(' 'isHolePlated'
booleanToken
')'

Description
isHolePlated specifies whether a pad or via style's hole is plated or non-plated.

Used By
padStyleDef, viaStyleDef

islandOutline (PCB)

islandOutline ::= '(' 'islandOutline'
{ pt }
')'

Description
islandOutline lists the vertices of the outline of a *copperPourIsland.*

Used By
copperPour95

islandRemoval

islandRemoval ::= '(' 'islandRemoval'
('None' | area | 'Interior' | 'Unconnected')
')'

Description
islandRemoval specifies zero or more modes for performing automatic island removal by *copperPour95*. *'None'* means do no island removal, *area* menas remove islands smaller than the spcified number of square database units, *'Interior'* means remove islands that do not have at least one edge on the exterior of the *copperPour95*, and *'Unconnected'* means remove all islands that do not connect to any copper in the current net. Note: all removal options can be mixed together in any order.

Used By
copperPour95

isPlane

isPlane ::= '(' 'isPlane'
booleanToken
')'

Description

isPlane indicates if a net is a plane net.

Notes

isPlane is read but ignored if present in a Schematic design file.

Used By

net

isPromptForRel

isPromptForRel ::= '(' 'isPromptForRel'
booleanToken
')'

Description

isPromptForRel indicates if the **Prompt For Origin** check box in the Options Grids dialog has been checked.

Used By

gridState

isRightReading (PCB)

isRightReading ::= '(' isRightReading
booleanToken
')'

Description

isRightReading indicates if attribute is set to be smart rotation. If it's true then when the text is rotated 180 and 270 degree the text will always read right to left and bottom to top . This value reflects the status of the **Right Reading** check box in the attribute property and pattern property dialogs.

Used By

attribute, refdes, value and type of Pattern

isRotated

isRotated ::= '(' 'isRotated'
booleanToken
')'

Description

isRotated indicates if a print job is to be rotated 90 degrees in the clockwise direction. This value reflects the status of the **Rotate** check box in the Print Setup dialogs.

Used By

printQueueEntry, schematicPrintSettings

isSelected (PCB)

isSelected ::= '(' 'isSelected'
booleanToken
')'

Description

isSelected indicates if a print queue entry is selected for output.

Used By

outputItemd

isThinStrokeText (PCB)

isThinStrokeText ::= '(' 'isThinStrokeText'
booleanToken
')'

Description
isThinStrokeText indicates if a print queue entry is to use thin strokes to print text objects whose styles currently indicate stroke display mode. The *isThinStrokeText* setting supercedes the *isDraft* setting for text. This value reflects the status of the **Thin Stroked Text** check box in the Print Options dialog.

Used By
printQueueEntry

isVisible

isVisible ::= '(' 'isVisible'
booleanToken
')'

Description
isVisible indicates the display status of an object. If False, the item is not displayed.

Used By
attr, border, net, titleSheet, zones

isVisibleGrid

isVisibleGrid ::= '(' 'isVisibleGrid'
booleanToken
‘)'

Description
isVisibleGrid indicates if a grid is visible. This value reflects the status of the **Visible** check box in the Options Grid dialog.

Used By
gridState

isVisibleOnDrag

isVisibleOnDrag ::= '(' 'isVisibleOnDrag'
booleanToken
')'

Description
isVisibleOnDrag indicates the display status of an invisible object as its location or orientation is edited in the workspace. When the edit operation is completed, the object reverts back to being invisible.

Used By
net

itemMnemonic (PCB)

itemMnemonic ::= '(' 'itemMnemonic'
(stringToken |
stringToken stringToken |
stringToken stringToken stringToken |
stringToken stringToken stringToken stringToken)
')'

Description

itemMnemonic is a string representing the data necessary to represent an aperture assignment. These strings are the same strings presented in the Apertures Assignments dialog list box.

Used By

apertureAssn

junction (SCH)

junction ::= '(' 'junction'
location
netNameRef
')'

Description

junction represents a Schematic junction object.

Notes

Junctions are written for information only; these are calculated by P-CAD programs during File Open, and are ignored when read from the input file.

Used By

schDrawObj

junctionSize

junctionSize ::= '(' 'junctionSize'
dbNumber
')'

Description

junctionSize describes the size of junction.

Used By

schematicDesignHeader

junctionSizePrint

junctionSizePrint ::= '(' 'junctionSizePrint'
dbNumber
')'

Description

junctionSizePrint describes the print size of junction.

Used By

schematicDesignHeader

justify

justify ::= '(' 'justify'
('UpperLeft' | 'UpperCenter' | 'UpperRight' | 'Left' | 'Center' | 'Right' | 'LowerLeft' | 'LowerCenter' | 'LowerRight')
')'

Description

justify indicates the location of a text object reference point relative to the text object bounding box.

Used By

attr, field, text

layerAttrs

layerAttrs ::= '(' 'layerAttrs'
layerAttrNameDef
attrMgr
')'

Description

Used By
ClassToClass, net, netClass

layerAttrNameDef

layerAttrNameDef ::= '(' 'layerAttrNameDef'
nameDef
')'

Description

Used By
layerAttrs

layerBias (PCB)

layerBias ::= '(' 'layerBias'
('AutoBias' | 'HorizBias' | 'VertBias')
')'

Description
layerBias indicates the preferred router bias for a layer. The value specifies horizontal, vertical, or allows the bias to be chosen automatically by the router. *layerBias* is valid for signal layers only and will be ignored if present for other layer types.

Used By
layerDef

layerContents (PCB)

layerContents ::= '(' 'layerContents'
layerNumRef
{ pcbDrawObj }
')'

Description
layerContents lists the objects defined on a given layer. This is used in the PCB design section to list layer-specific items in the design, and within pattern definitions to describe pattern objects that are layer-specific.

Used By
patternDef, pcbDesign

See Also
layerNum

layerDef (PCB)

layerDef ::= '(' 'layerDef'
layerNameDef
layerNum
layerType
[layerBias]

[planeNetNameRef]
{ attr }
[titleSheet]
[fieldSetRef]
[drillSymDisplayConfigRef]
')'

Description

layerDef defines a PCB layer. The definition includes the layer name, number and type, layer bias, and plane net, if it is a plane layer

Notes

The layer is not a plane layer if *planeNetNameRef* is not present. As of v14, the design rules for the layer, *padToPadClearance, padToLineClearance, etc. are under the {att} production as attributes*. The appearance of these values outside of attributes is obsolete.

Defaults

layerBias: AutoBias
attr padToPadClearance: 0 mils
attr padToLineClearance: 0 mils
attr lineToLineClearance: 0 mils
attr viaToPadClearance: uses *attr* padToPadClearance
attr viaToLineClearance: uses *attr* viaToLineClearance
attr viaToViaClearance: uses *attr* padToPadClearance

Used By

pcbDesign

layerDisabled (PCB)

layerDisabled ::= '(' 'layerDisabled'
layerNumRef
')'

Description

layerDisabled indicates that a PCB layer is disabled, by its layer number. This value reflects the layer status as displayed **Layers** list box in the Options Current Layer dialog.

Used By

layerState

layerList (PCB)

layerList ::= '(' 'layerList'
{ layerNumRef }
')'

Description

layerList lists the PCB layers that in a print, Gerber, or N/C Drill print job entry.

Used By

camQueueEntry, printQueueEntry, designView

layerNameDef (PCB)

layerNameDef ::= nameDef

Description

layerNameDef defines the name assigned to a PCB layer. This name reflects contents of the **Layer Name** edit box in the Options Current Layer dialog.

Used By
layerDef

layerNum (PCB)

layerNum ::= '(' 'layerNum'
integerToken
')'

Description
layerNum defines the number assigned to a PCB layer. This number reflects contents of the **Layer Number** edit box in the Options Current Layer dialog.

Notes
P-CAD PCB uses the concept of a "multi-layer": a layer where objects are defined that belong to all layers or are not specific to a particular layer. Pattern objects, for example, are defined as existing on the multi-layer. The multi-layer is assigned a layer number of zero (0). For this reason, users may not create a layer with a layer number of zero.

Used By
layerDef

layerNumRef (PCB)

layerNumRef ::= '(' 'layerNumRef'
integerToken
')'

Description
layerNumRef is a reference to a PCB layer by its layer number.

Used By
currentLayer, layerContents, layerDisabled, layerList, padShape, viaShape

layerPair

layerPair ::= '(' 'layerPair'
integerToken
integerToken
')'

Description
layerPair defines the layer pairing between two layers. The two integerTokens are the layer numbers of the two layers

Used By
layerPairs

layerSet

layerSet ::= '(' 'layerSet'
layerSetName
layerSetCurrentLayerNum
{ layerNameDef }
')'

Description
layerSet defines a single layer set in a design.

Notes

If the *layerSetCurrentLayerNum* setting represents a non-positive value, then the layer set is considered to be empty, regardless of any provided *layerNameDef*s.

Used By
layerSets

layerSetCurrentLayerNum

layerSetCurrentLayerNum ::= stringToken

Description
layerSetCurrentLayerNum is a text string that signifies a *layerSet*'s current layer number.

Used By
layerSet

layerSetName

layerSetName ::= nameDef

Description
layerSetName defines the name of a layer set.

Used By
layerSet

layerSets

layerSets ::= '(' 'layerSets'
{ layerSet }
')'

Description
layerSets defines the various *layerSet*s in a design.

Used By
pcbDesign

layersStackup (PCB)

layersStackup ::= '(' 'layersStackup'
{ layerStackupData }
')'

Description
layersStackup specify a list of layerStackupData.

Used By
layerStackupData

layerStackupData (PCB)

layerStackupData ::= '(' 'layerStackupData'
layerStackupName
layerStackupMaterial
layerStackupThickness
layerStackupDielectricConstant
')'

Description
layerStackupData is used to specify layer stackup information.

Used By
layersStackup

layerStackupDelectricConstant (PCB)

layerStackupDelectricConstant ::= '(' 'layerStackupDelectricConstant'
stringToken
')'

Description
layerStackupDielectricConstant is used to specify the dielectric constant of a substrate or non-conducting core. This is typically as floating point number such as 4.7.

Used By
layerStackupData

layerStackupInfo (PCB)

layerStackupInfo ::= '(' 'layerStackupInfo'
layerStackupStyle
')'

Description
layerStackupInfo is used to specify layer stackup diagram-specific information.

Used By
diagram

layerStackupMaterial (PCB)

layerStackupMaterial ::= '(' 'layerStackupMaterial'
stringToken
')'

Description
layerStackupMaterial is used to specify the type of material used in a substrate or conducting layer, such as FR-4 or copper respectively.

Used By
layerStackupData

layerStackupName (PCB)

layerStackupName ::= '(' 'layerStackupName'
stringToken
')'

Description
layerStackupName specifies the name of the substrate or conducting layer. These are autogenerated by the program based on the signal/plane layers and the index of the substrate between the conducting layers. Changing the names to something other than was generated will cause the data to be lost.

Used By
layerStackupData

layerStackupStyle (PCB)

layerStackupStyle ::= '(' 'layerStackupStyle'
integerToken
')'

Description

layerStackupStyle is used to specify the style of a layer stackup diagram. This integer is 1 for "Style #1" and 2 for "Style #2."

Used By

layerStackupInfo

layerStackupThickness (PCB)

layerStackupThickness ::= '(' 'layerStackupThickness'
stringToken
')'

Description

layerStackupThickness specify the width of the substrate or conducting layer.

Used By

layerStackupData

layerState

layerState ::= '(' 'layerState'
currentLayer
{ layerDisabled }
')'

Description

layerState indicates the current layer or sheet. For PCB designs, a list of all disabled layers is provided. This information reflects the status of the design layers or sheets as presented in the Options Current Layer and Options Current Sheet dialogs, and the Status Line **Current Layer** or **Current Sheet** combo box.

Notes

layerDisabled information, if included in a schematic design file, will be read but ignored.

Used By

programState

layerType (PCB)

layerType ::= '(' 'layerType'
('Signal' | 'Plane' | 'NonSignal')
')'

Description

layerType indicates whether a layer is a plane layer, signal layer, or non-signal layer. It is also used to define the default pad or via shape to be assigned to a layer type.

Used By

layerDef, padShape, viaShape

library

library ::= '(' 'library'
libraryNameDef
{ styleDef }
{ patternDef }
{ patternDefExtended }
{ patternAlias}
{ symbolDef }
{ symbolAlias}

{ compDef }
{ compAlias}
')'

Description

The *library* section provides definitions for styles, patterns, symbols, and components. These definitions may be referenced later by other sections of the file.

Notes

The current implementation supports only one *library* section per design file.

PCB design files may not include *symbolDef* information.

Schematic design files may not include *patternDef* information.

Used By

PCAD_ASCII

libraryNameDef

libraryNameDef ::= nameDef

Description

libraryNameDef provides a unique name for a *library* section.

Used By

library

line

line ::= '(' 'line'
oneEnd
[endStyle]
anotherEnd
[endStyle]
[width]
[style]
[isFlipped]
[netNameRef]
{ dimensionRef }
[isFixed]
')'

Description

line represents a Schematic or PCB line object. A line is defined by its endpoints, an end style for each end point, width and line style, may be flipped, and may belong to a net.

Notes

The *endStyle* of each end of a line can be *LeftLead, RightLead, Rounded*, or *TwoLeads*. PCB lines are always *Rounded.* If a line in Schematic is a wire and the wire is connected to a bus, the *endStyle* of end point connected to the bus is the bus connection style. Otherwise the *endStyle* is *Rounded.* If an end point's e*ndStyle* is missing, the *endStyle* is assumed to be *Rounded.*

The *width* entry for lines written by P-CAD applications are handled as follows: Schematic lines written always contain a *width* entry; Schematic wires do not contain a *width* entry if the wire is the default width; PCB lines and traces write *width* only if the wire width is different from the default.

PCB lines may only be defined with a Solid style. If a different style is specified, it is read and ignored.

The line does not belong to a net if *netNameRef* is not present.

Net information for lines in the *library* section is ignored.

Defaults

endStyle: Rounded
width: PCB: 10 mils; Schematic wires: Thin
style: solid
isFlipped: False
isFixed: False

Used By

pcbDrawObj, schDrawObj, wire

lineKeepOut (PCB)

lineKeepOut ::= '(' 'lineKeepOut'
oneEnd
anotherEnd
')'

Description

lineKeepOut represents a PCB line keepout object. A line keepout is defined by two endpoints.

Used By

pcbDrawObj

lineToLineClearance (PCB)

lineToLineClearance ::= '(' 'lineToLineClearance'
dbNumber
')'

Description

lineToLineClearance specifies the line to line clearance for DRC checking. This value is specific to a layer. The value reflects the clearance specified in the Design Rule Check Clearances dialog for the given layer.

Used By

layerDef

localSwell (PCB)

localSwell ::= '(' 'localSwell'
dbNumber
')'

Description

localSwell defines a plane swell value for a pad or via that overrides the global swell value for the plane. This value reflects the **Local Swell** specified in the Modify Pad Style dialog. Note that plane swell is not applicable to pads or vias that are thermally or directly connected to a plane.

Used By

padStyleDef, viaStyleDef

location

location ::= pt

Description

location specifies an x,y location on a PCB or Schematic design. The location will include a *dbUnit* specification if the value can not be represented exactly in the design *fileUnits*.

Used By
anotherEnd, arc, attr, field, gluePoint, ieeeSymbol, infoPoint, junction, metafile, oneEnd, pad, pattern, pickPoint, pin, refPoint, relOrigin, symbol, table, text, via

See Also
dbNumber, dbUnits, fileUnits

majorVersion

majorVersion ::= integerToken

Description
majorVersion indicates the major revision value for a version number.

Used By
asciiVersion

metafile (PCB)

metafile ::= '(' 'metafile'
location
[extent]
scaleFactor
binaryData
')'

Description
metafile is used to describe an object in P-CAD Picture format. These objects cannot easily be generated in textual format, so they are stored as a sequence of binary data.

Notes
P-CAD Picture objects can only be created by P-CAD Document Toolbox.

Used By
detail, pcbDrawObj

minorVersion

minorVersion ::= integerToken

Description
minorVersion indicates the minor revision value for a version number.

Used By
asciiVersion

minuteNumber

minuteNumber ::= integerToken

Description
minuteNumber specifies the minute portion of a time value. The value may range from 0-59.

Used By
time

monthNumber

monthNumber ::= integerToken

Description

monthNumber specifies the month portion of a date value. The value may range from 1-12.

Used By

date

multiLayer (PCB)

multiLayer ::= '(' 'multiLayer'
{ pcbDrawObj }
')'

Description

multiLayer represents a listing of PCB objects that are not defined as existing on a specific layer. *multiLayer* may be used to represent design-level objects such as free pads, vias, connections, and patterns.

Used By

patternDef, pcbDesign

nameDef

nameDef ::= stringToken

Description

nameDef is used to define the name of a particular object. This name can later be referenced using *nameRef.*

Used By

asciiNameDef, busNameDef, componentNameDef, componentPinDesDef, fieldNameDef, fieldSetNameDef, layerNameDef, libraryNameDef, netlistNameDef, netNameDef, originalName, patternNameDef, pcbDesignNameDef, refDesDef, schematicDesignNameDef, sheetNameDef, styleNameDef, symbolNameDef

nameRef

nameRef ::= stringToken

Description

nameRef provides a reference to a previously-defined *nameDef.*

Used By

busNameRef, componentNameRef, componentPinDesRef, fieldNameRef, fieldSetNameRef, netNameRef, padStyleRef, patternNameRef, refDesNameRef, symbolNameRef, textStyleRef, viaStyleRef

ncDrillMMFormat (PCB)

ncDrillMMFormat ::= '(' 'ncDrilMMFormat'
integerToken
integerToken
')'

Description

ncDrillMMFormat specifies the format used for N/C Drill output in *mm* units mode.

Notes

The two integers signify digits left of decimal and digits right of decimal, respectively. Legal formats are 3:2, 3:3, and 4:2.

Used By

ncDrillSettings

ncDrillSettings (PCB)

ncDrillSettings ::= '(' 'ncDrillSettings'

units
[ncDrillMMFormat]
codeFormat
zeroFormat
outputPath
viewLog
autoClear
{ toolDef }
{ toolAssn }
{ camQueueEntry }
')'

Description

ncDrillSettings specifies the options to be used for N/C Drill output. It contains all of the information specified in the dialogs accessed from the File N/C Drill dialog.

Notes

If *units = 'mm'*, then *ncDrillMMFormat* optionally specifies the format (3:2, 3:3, or 4:2). The default is 4:2.

Used By

pcbDesign

net

net ::= '(' 'net'
netNameDef
[isPlane]
[isVisible]
[isVisibleOnDrag]
[netPlaneColor]
[netColor]
{ node }
{ attr }
{ layerAttrs }
')'

Description

net defines a net in a design. It consists of a list of nodes and attrs that make up the net.

Notes

In PCB designs, *isVisible* controls the visibility of the connections in a net. If *isVisibile* is set *False*, then *isVisibleOnDrag* controls the visibility of the connections during graphical editing operations. These settings reflect the connection visibility that is set in the Edit Nets dialog with the **Show** , **Show Only On Drag**, and **Hide** buttons.

isPlane is no longer used by P-CAD PCB.

isPlane, *isVisible*, *isVisibleOnDrag,* and *netPlaneColor* are not used by P-CAD Schematic.

Defaults

isPlane: False
isVisible: True

Used By

netlist

netClass

netClass ::= '(' 'netClass'
netClassNameDef
{ netNameRef }

{ attr }
{ LayerAttrs }
')'

Description
netClass defines a net class in a design. It consists of a list of *netNameRefs* and *attrs* that make up the net class.

Used By
netlist

netClassNameDef

netClassNameDef ::= nameDef

Description
*netClassNameD*ef defines the name of a net class in a design.

Used By
netClass

netClassNameRef

netClassNameRef ::= '(' 'classNameRef'
nameRef
')'

Description
netClassNameRef is used to provide a reference to an existing net class. The net class name must have been previously defined in a netlist section.

Used By
classToClass

netColor

netColor ::= '(' netColor'
stringToken
')'

Description
netColor is used to define a color for nets. The color is used to uniquely identify a net and all its objects.

Used By
net

netlist

netlist ::= '(' 'netlist'
netlistNameDef
[globalAttrs]
{ compInst }
{ net }
{ netClass }
{ classToClass }
{variant}
')'

Description
netlist lists the component, net, net class, and net class to net class information for a design.

Used By

PCAD_ASCII

netlistNameDef

netlistNameDef ::= nameDef

Description
netlistNameDef provides a unique name for each *netlist* section in a design.

Used By
netlist

netNameDef

netNameDef ::= nameDef

Description
netNameDef defines the name of a net in a design.

Used By
net

netNameRef

netNameRef ::= '(' 'netNameRef'
nameRef
')'

Description
netNameRef is used to provide a reference to an existing net. The net name must have been previously defined in a *netlist* section.

Used By
arc, copperPour, fromTo, junction, line, pad, planeNetRef, poly, port, ruleSPECCTRARouteExposePin, ruleSPECCTRARouteLoadPin, ruleSPECCTRARouteReorder, ruleSPECCTRARouteSourcePin, ruleSPECCTRARouteTerminatorPin, via

netPlaneColor

netPlaneColor ::= '(' 'planeColor'
stringToken
')'

Description
netPlaneColor is used to define a color for plane nets. The color is used for drawing plane objects and the thermal connection indicators on pads and vias..

Used By
net

noCopperPourConnect

noCopperPourConnect ::= '(' 'noCopperPourConnect'

BooleanToken
')'

Description
noCopperPourConnect defines whether a *padShape* or v*iaShape* on a specific layer is forcibly prohibited from thermally connecting to copper pours on that layer. Instead, a routed trace is required to connect the pad or via to the net.

Used By
padShape

node

node ::= '(' 'node'
refDesNameRef
componentPinDesRef
')'

Description
node specifies a node in a net. It references a specific component pin on a specific component.

Used By
net

noteAnnotation

noteAnnotation ::= '(' 'noteAnnotation'
('box' | 'circle' | 'triangle' | 'none')
')'

Description
noteAnnotation describes the graphical annotation for a note. The annotation can be a box, a circle, a triangle, or none.

Used By
noteDef

noteDef

noteDef ::= '(' 'note'
noteNum
noteValue
[noteAnnotation]
')'

Description
noteDef specifies a specific numbered note.

Notes
Notes can only be specified by the user with the P-CAD Document Toolbox option.

Defaults
noteAnnotation: none

Used By
fieldSet

noteNum

noteNum ::= integerToken

Description
noteNum specifies the number of a specific note or revision note.

Used By
noteDef, noteRef, revisionNoteRef

noteRef

noteRef ::= '(' 'note' noteNum ')'

Description
noteRef allows a field to refer to a previously-defined note by its number.

Used By
fieldType

noteTableInfo

noteTableInfo ::= '(' 'noteTableInfo'
numDirection
columnWidth
')'

Description
noteTableInfo specifies the information specific to a note table or a revision note table.

Used By
table

noteValue

noteValue ::= stringToken

Description
noteValue defines the string value of a note.

Used By
noteDef

number

number ::= '(' 'number'
integerToken
')'

Description
number designates a unique identifier for an info point (DRC/ERC dot).

Used By
infoPoint

numberToken

numberToken

Description
numberToken is used to represent a number which may or may not include a decimal point.

Used By
many productions

numBytes (PCB)

numBytes ::= '(' 'numBytes'
integerToken
')'

Description
numBytes is used to specify the number of bytes that follow.

Used By
binaryData

numDirection

numDirection ::= '(' 'numDirection'
('ascending' | 'descending')
)'

Description
numDirection indicates an ascending (1, 2, 3…) or descending (…3, 2, 1) numbering style.

Used By
horizontalZones, noteTableInfo, verticalZones

numFormat (PCB)

numFormat ::= '(' 'numFormat'
('gbr53' | 'gbr44' | 'float')
')'

Description
numFormat specifies the resolution of the numbers in a Gerber Output file. *gbr53* indicates that there are five digits to the left of the decimal point and three digits to the right. *gbr44* indicates that there are four digits to the left of the decimal point and four digits to the right. *float* means that the decimal point is explicitly supplied; this setting is not currently supported by P-CAD PCB. *numFormat* reflects the **Numeric Format** radio buttons in the Gerber Format dialog.

Used By
gerberSettings

numMajorColumns

numMajorColumns ::= '(' 'numMajorColumns'
integerToken
')'

Description
numMajorColumns designates the number of columns a Net Index Table is displayed with. This only applies to a Net Index Table type.

Used By
table

numPads

numPads ::= '(' 'numPads'
integerToken
')'

Description
numPads specifies the number of pads in a component.

Used By
attachedPattern

numParts

numParts ::= '(' 'numParts'
integerToken

')'

Description

numParts specifies the number of parts in a component. This value reflects the number in the **Number of Parts** edit box in the Edit Attach Symbols dialog in P-CAD Library Manager.

Used By

compHeader

numPins

numPins ::= '(' 'numPins'
integerToken
')'

Description

numPins specifies the number of pins in a component. This value reflects the number of rows in a component spreadsheet in P-CAD Library Manager.

Used By

compHeader

numType

numType ::= '(' 'numType'
('Alphabetic' | 'Numeric')
)'

Description

numType indicates an alphabetic or numeric numbering style.

Used By

compHeader, horizontalZones, verticalZones

odbLayerContext

odbLayerContext ::= '(' 'odbLayerContext'
('odbBoard' | 'odbMisc')
')'

Description

odbLayerContext specifies the ODB++ layer context..

Used By

odbLayerDef:

odbLayerDef

odbLayerDef ::= '(' 'odbLayerDef'
odbLayerName
odbLayerContext
odbLayerLayerType
odbLayerPolarity
odbLayerStartLayer
odbLayerEndLayer
odbLayerIsSelected
odbLayerMirror
odbLayerPads
odbLayerVias
odbLayerRefdes

odbLayerType
odbLayerValue
odbLayerTitle
odbLayerNoMtHoleCu
odbLayerPlatedHoles
odbLayerNonPlatedHoles
{odbLayerIncludedLayer}
')'

Description
odbLayerDef contains the settings for outputting a specific ODB++ layer.

Used By
odbSettings:

odbLayerEndLayer

odbLayerEndLayer ::= '(' 'odbLayerEndLayer'
stringToken
')'

Description
odbLayerEndLayer is the name of the ending board layer for a ODB++ drill layer.

Used By
odbLayerDef:

odbLayerIncludedLayer

odbLayerIncludedLayer ::= '(' 'odbLayerIncludedLayer'
stringToken
')'

Description
odbLayerIncludedLayer indicates a secondary layer to be included with the design layer referenced by odbLayerName during output. There can be zero or more included layers per each ODB++ layer/design layer.

Used By
odbLayerDef:

odbLayerIsSelected

odbLayerIsSelected ::= '(' 'odbLayerIsSelected'
booleanToken
')'

Description
odbLayerIsSelected indicates whether this ODB++ layer is selected for output..

Used By
odbLayerDef:

odbLayerLayerType

odbLayerLayerType ::= '(' 'odbLayerLayerType'
('odbSignal' | 'odbPowerGround' | 'odbMixed' | 'odbSolderMask' | 'odbSolderPaste' | 'odbSilkScreen' | 'odbDrill' | 'odbRout' | 'odbDocument' | 'odbComponent')
')'

Description
odbLayerLayerType specifies the ODB++ layer context..

Used By
odbLayerDef:

odbLayerMirror

odbLayerMirror ::= '(' 'odbLayerMirror'
booleanToken
')'

Description
odbLayerMirror indicates whether this ODB++ layer is mirrored for output..

Used By
odbLayerDef:

odbLayerName

odbLayerName ::= '(' 'odbLayerName'
stringToken
')'

Description
odbLayerName is the name of the ODB++ output layer. It is also the name of the design layer who's data is being output. For an ODB++ drill layer it is a composition of the start and end layers.

Used By
odbLayerDef:

odbLayerNoMtHoleCu

odbLayerNoMtHoleCu ::= '(' 'odbLayerNoMtHoleCu'
booleanToken
')'

Description
odbLayerNoMtHoleCu indicates whether this ODB++ layer will include mounting hole pad copper with output..

Used By
odbLayerDef:

odbLayerNonPlatedHoles

odbLayerNonPlatedHoles ::= '(' 'odbLayerNonPlatedHoles'
booleanToken
')'

Description
odbLayerNonPlatedHoles indicates whether this ODB++ drill layer includes non-plated holes with output..

Used By
odbLayerDef:

odbLayerPads

odbLayerPads ::= '(' 'odbLayerPads'
booleanToken
')'

Description
odbLayerPads indicates whether this ODB++ layer includes pads with output..

Used By
odbLayerDef:

odbLayerPlatedHoles

odbLayerPlatedHoles ::= '(' 'odbLayerPlatedHoles'
booleanToken
')'

Description
odbLayerPlatedHoles indicates whether this ODB++ drill layer includes plated-holes with output..

Used By
odbLayerDef:

odbLayerRefdes

odbLayerRefdes ::= '(' 'odbLayerRefdes'
booleanToken
')'

Description
odbLayerRefdes indicates whether this ODB++ layer includes component reference designators with output..

Used By
odbLayerDef:

odbLayerStartLayer

odbLayerStartLayer ::= '(' 'odbLayerStartLayer'
stringToken
')'

Description
odbLayerStartLayer is the name of the starting board layer for a ODB++ drill layer.

Used By
odbLayerDef:

odbLayerType

odbLayerTupe ::= '(' 'odbLayerType'
booleanToken
')'

Description
odbLayerType indicates whether this ODB++ layer includes component type attribute with output..

Used By
odbLayerDef:

odbLayerValue

odbLayerValue ::= '(' 'odbLayerValue'
booleanToken
')'

Description
odbLayerValue indicates whether this ODB++ layer includes component value attribute with output..

Used By

odbLayerDef:

odbLayerVias

odbLayerVias ::= '(' 'odbLayerVias'
booleanToken
')'

Description
odbLayerVias indicates whether this ODB++ layer includes vias with output..

Used By
odbLayerDef:

odbSettings

odbSettings ::= '(' 'odbSettings'
outputPath
[viewLog]
{odbLayerDef}
')'

Description
odbSettings is an *outputPath*, a *viewLog* indicator, and a list of *odbLayerDef*. Together this describes the settings for output of an ODB++ file.

Defaults
viewLog: False

Used By
pcbDesign:

offset

offset ::= '(' 'offset'
xPoint
yPoint
')'

Description
offset describes the distance that an object is offset from the origin.

Used By
apertureDef, border, outputItem, schematicPrintSettings, titleSheet

oneEnd

oneEnd ::= location

Description
oneEnd represents the first of two locations defining an item.

Used By
bus, fromTo, line, lineKeepOut

onlineDrcClearanceEnabled

onlineDrcClearanceEnabled ::= '(' 'onlineDrcClearanceEnabled'
booleanToken
')'

Description

onlineDrcClearanceEnabled indicates the state of Online DRC Clearance rules checking. It reflects the state of the **Clearance** check box in the Options Configure dialog.

Used By

onlineDrcState

onlineDrcEnabled

onlineDrcEnabled ::= '(' 'onlineDrcEnabled'
booleanToken
')'

Description

onlineDrcEnabled indicates the state of Online DRC checking. It reflects the state of the **Enable Online DRC** check box in the Options Configure dialog. It is not used by P-CAD Schematic.

Used By

onlineDrcState

onlineDrcComponentEnabled

onlineDrcComponentEnabled ::= '(' 'onlineDrcComponentEnabled'
booleanToken
')'

Description

onlineDrcComponentEnabled indicates the state of Online DRC Component rules checking. It reflects the state of the **Component** check box in the Options Configure dialog.

Used By

onlineDrcState

onlineDrcNetlistEnabled

onlineDrcNetlistEnabled ::= '(' 'onlineDrcNetlistEnabled'
booleanToken
')'

Description

onlineDrcNetlistEnabled indicates the state of Online DRC Netlist rules checking. It reflects the state of the **Netist** check box in the Options Configure dialog.

Used By

onlineDrcState

onlineDrcReport

onlineDrcReport ::= '(' 'onlineDrcReport'
booleanToken
')'

Description

onlineDrcReport indicates whether reports should be automatically generated during Online DRC checking. It reflects the state of the **View Report** check box in the Options Configure dialog. It is not used by P-CAD Schematic.

Used By

onlineDrcState

onlineDrcSameCompPadsEnabled

onlineDrcSameCompPadsEnabled ::= '(' 'onlineDrcSameCompPadsEnabled'
booleanToken
')'

Description
onlineDrcSameCompPadsEnabled indicates the state of Online DRC Pad-To-Pad Clearance rules checking for pads that have the same parent component. It reflects the state of the **Same-component Pads** check box in the Options Configure dialog.

Used By
onlineDrcState

onlineDrcSilkEnabled

onlineDrcSilkEnabled ::= '(' 'onlineDrcSilkEnabled'
booleanToken
')'

Description
onlineDrcSilkEnabled indicates the state of Online DRC Clearance rules checking for silkscreen layers. It reflects the state of the **Silk** check box in the Options Configure dialog.

Used By
onlineDrcState

onlineDrcState

onlineDrcState ::= '(' 'onlineDrcState'
onlineDrcEnabled
onlineDrcReport
[onlineDrcClearanceEnabled]
[onlineDrcTextEnabled]
[onlineDrcSameCompPadsEnabled]
[onlineDrcComponentEnabled]
[onlineDrcSilkEnabled]
[onlineDrcNetlistEnabled]
[onlineDrcWidthEnabled]
')'

Description
onlineDrcState describes the state of the Online DRC settings for a design. These values are set in the **Online DRC** group box in the Options Configure dialog. They are not used by P-CAD Schematic.

Defaults
All optional fields default to *True* unless specified as *False*.

Used By
programState

onlineDrcTextEnabled

onlineDrcTextEnabled ::= '(' 'onlineDrcTextEnabled'
booleanToken
')'

Description
onlineDrcTextEnabled indicates the state of Online DRC Clearance rules checking for text objects. It reflects the state of the **Text** check box in the Options Configure dialog.

Used By

onlineDrcState

onlineDrcWidthEnabled

onlineDrcWidthEnabled ::= '(' 'onlineDrcWidthEnabled'
booleanToken
')'

Description

onlineDrcWidthEnabled indicates the state of Online DRC Width rules checking. It reflects the state of the **Width** check box in the Options Configure dialog.

Used By

onlineDrcState

orderedLayerList (PCB)

orderedLayerList ::= '(' 'orderedLayerList'
{ layerNumRef }
')'

Description

orderedLayerList lists the PCB layers to be printed for print job entry, in the order of printing.

Used By

printQueueEntry

orient (SCH)

orient ::= '(' 'orient'
('Up' | 'Left' | 'Down' | 'Right')
')'

Description

orient specifies the orientation of a bus entry. A bus entry will be oriented in the direction that the wire is traveling to meet the bus.

Used By

busEntry

originalName

originalName ::= '(' 'originalName'
nameDef
')'

Description

originalName identifies the actual name of a component, pattern, or symbol, regardless of the name assigned to it in the P-CAD ASCII file.

Notes

There may be more than one *compDef, patternDef,* or *symbolDef* for a given P-CAD name; additional definitions will be created if the two items of the same name are not equivalent. The *compNameDef, patternNameDef,* or *symbolNameDef* uniquely identifies each template, and *originalName* specifies the actual P-CAD name.

The *originalName* for a component indicates the component name, which appears in the **Type** edit box in the Modify Component dialog in PCB, or the Modify Part dialog in Schematic. The *originalName* for a pattern identifies the pattern name, which appears in the **Pattern** edit box in the Modify Component dialog in PCB. The

originalName for a symbol designates the symbol name, which appears in the **Symbol** edit box in the Modify Part dialog in Schematic.

Used By

compDef, compInst, patternDef, symbolDef

outputConnect (PCB)

outputConnect ::= '(' 'outputConnect'
booleanToken
')'

Description

outputConnect indicates whether connections are to be included in printer output. This value reflects the status of the **Connections** check box in the Setup Print Jobs dialog.

Used By

outputItem

outputCutout (PCB)

outputCutout ::= '(' 'outputCutout'
booleanToken
')'

Description

outputCutout indicates whether cutouts are to be included in printer output. This value reflects the status of the **Cutouts** check box in the Setup Print Jobs dialog.

Used By

printQueueEntry

outputDrillSym (PCB)

outputDrillSym ::= '(' 'outputDrillSym'
booleanToken
')'

Description

outputDrillSym indicates whether drill symbols are to be included in printer or CAM output. This value reflects the status of the **Drill Sym** check box in the Setup Print Jobs and the Setup Output Files dialogs.

Used By

outputItem

outputDrillSymNonplated (PCB)

outputDrillSymNonplated ::= '(' 'outputDrillSymNonplated'
booleanToken
')'

Description

outputDrillSymNonplated indicates whether drill symbols/holes for nonplated holes are to be included in printer or CAM output. This value reflects the status of the **Plated Holes** radio buttons in the Setup Print Jobs and the Setup Output Files dialogs.

Used By

outputItem

outputDrillSymPlated (PCB)

outputDrillSymPlated ::= '(' 'outputDrillSymPlated'
booleanToken
')'

Description
outputDrillSymPlated indicates whether drill symbols/holes for plated holes are to be included in printer or CAM output. This value reflects the status of the **Plated Holes** radio buttons in the Setup Print Jobs and the Setup Output Files dialogs.

Used By
outputItem

outputGluedot (PCB)

outputGluedot ::= '(' 'outputGluedot'
booleanToken
')'

Description
outputGluedot indicates whether glue dots are to be included in printer output. This value reflects the status of the **Glue Dot** check box in the Setup Print Jobs dialog.

Used By
printQueueEntry

outputHoles (PCB)

outputHoles ::= '(' 'outputHoles'
booleanToken
')'

Description
outputHoles indicates whether pad and via holes are to be included in printer or CAM output. This value reflects the status of the **Pad/Via Holes** check box in the Setup Print Jobs and the Setup Output Files dialogs.

Used By
outputItem

outputItem (PCB)

outputItem ::= isSelected
[offset]
[drillSymSize]
[outputDrillSymPlated]
[outputDrillSymNonplated]
[outputMirror]
[outputPads]
[outputVias]
[outputHoles]
[outputRefDes]
[outputType]
[outputValue]
[outputDrillSym]
[outputKeepout]
[outputConnect]
[outputNoMountingHole]
[outputTitle]
[outputTestPoint]
[variantName]

Description

outputItem contains several settings for printer and CAM output.

Defaults

offset: none
drillSymSize: 0
outputDrillSymPlated: True
outputDrillSymNonplated: True
outputMirror: False
outputPads: False
outputVias: False
outputHoles: False
outputRefDes: False
outputType: False
outputValue: False
outputDrillSym: False
outputKeepout: False
outputConnect: False
outputNoMountingHole: False
outputTitle: False
variantName: None

Used By

camQueueEntry, printQueueEntry

outputKeepout (PCB)

outputKeepout ::= '(' 'outputKeepout'
booleanToken
')'

Description

outputKeepout indicates whether keepouts are to be included in printer output. This value reflects the status of the **Keepout** check box in the Setup Print Jobs dialog.

Used By

outputItem

outputMirror (PCB)

outputMirror ::= '(' 'outputMirror'
booleanToken
')'

Description

outputMirror indicates whether printer or CAM output should be mirrored (reversed) about the Y-axis. This value reflects the status of the **Mirror** check box in the Setup Print Jobs and the Setup Output Files dialogs.

Used By

outputItem

outputNoMountingHole (PCB)

outputNoMountingHole ::= '(' 'outputNoMountingHole'
booleanToken
')'

Description

outputNoMountingHole directs various output utilities to suppress output of mounting hole copper. This value reflects the status of the **No Mt Hole Cu** check box in the Setup Print Jobs and the Setup Output Files dialogs.

Used By
outputItem

outputPads (PCB)

outputPads ::= '(' 'outputPads'
booleanToken
')'

Description
outputPads indicates whether pads are to be included in printer or CAM output. This value reflects the status of the **Pads** check box in the Setup Print Jobs and the Setup Output Files dialogs.

Used By
outputItem

outputPath (PCB)

outputPath ::= '(' 'outputPath'
stringToken
')'

Description
outputPath specifies the directory path to be used for generation of CAM files. This setting reflects the contents of the **Output Path** edit box in the Setup Output Files dialogs.

Used By
gerberSettings, ncDrillSettings

outputPickPlace (PCB)

outputPickPlace ::= '(' 'outputPickPlace'
booleanToken
')'

Description
outputPickPlace indicates whether pick and place dots are to be included in printer output. This value reflects the status of the **Pick** and **Place** check box in the Setup Print Jobs dialog.

Used By
printQueueEntry

outputRefDes (PCB)

outputRefDes ::= '(' 'outputRefDes'
booleanToken
')'

Description
outputRefDes indicates whether *RefDes* attributes are to be included in printer or CAM output. This value reflects the status of the **RefDes** checkbox in the Setup Print Jobs and the Setup Output Files dialogs.

Used By
outputItem

outputTestPoint (PCB)

outputTestPoint ::= '(' 'outputTestPoint'
booleanToken
')'

Description

outputType indicates whether testpoint attributes are to be included in printer or CAM output. This value reflects the status of the **TestPoint** checkbox in the Setup Print Jobs and the Setup Output Files dialogs.

Used By

outputItem

outputTitle (PCB)

outputTitle ::= '(' 'outputTitle'
booleanToken
')'

Description

outputType indicates whether title attributes are to be included in printer or CAM output. This value reflects the status of the **Title** checkbox in the Setup Print Jobs and the Setup Output Files dialogs.

Used By

outputItem

outputType (PCB)

outputType ::= '(' 'outputType'
booleanToken
')'

Description

outputType indicates whether type attributes are to be included in printer or CAM output. This value reflects the status of the **Type** checkbox in the Setup Print Jobs and the Setup Output Files dialogs.

Used By

outputItem

outputValue (PCB)

outputValue ::= '(' 'outputValue'
booleanToken
')'

Description

outputValue indicates whether value attributes are to be included in printer or CAM output. This value reflects the status of the **Value** check box in the Setup Print Jobs and the Setup Output Files dialogs.

Used By

outputItem

outputVias (PCB)

outputVias ::= '(' 'outputVias'
booleanToken
')'

Description

outputVias indicates whether vias are to be included in printer or CAM output. This value reflects the status of the **Vias** check box in the Setup Print Jobs and the Setup Output Files dialogs.

Used By

outputItem

outsideDiam (PCB)

outsideDiam ::= '(' 'outsideDiam'
dbNumber
')'

Description
outsideDiam represents the outside diameter of a thermal pad or via shape.

Used By
shapeWidth

outsideEdgeStyle (SCH)

outsideEdgeStyle ::= '(' 'outsideEdgeStyle'
('None' | 'Dot' | 'PolarityIn' | 'PolarityOut')
')'

Description
outsideEdgeStyle defines the outside edge display characteristics for a symbol pin. This value reflects the status of the **Outside Edge** combo box in the Place Pin dialog.

Used By
pin

outsideStyle (SCH)

outsideStyle ::= '(' 'outsideStyle'
('None' | 'FlowIn' | 'FlowOut' | 'FlowBi' | 'Analog' | 'Digital' | 'NonLogic')
')'

Description
outsideStyle defines the outside display characteristics for a symbol pin. This valuereflects the status of the **Outside** combo box in the Place Pin dialog.

Used By
pin

pad (PCB)

pad ::= '(' 'pad'
padNum
padStyleRef
location
[rotation]
[isFlipped]
[netNameRef]
{ dimensionRef }
[isFixed]
[testPointId]
')'

Description
pad represents a pad in a PCB design. It references a pad style that must have been previously defined in a *library* section. If the pad is a free pad with net connections, it will have a net name reference.

Defaults
rotation: 0 degrees
isFlipped: False
isFixed: False

Used By
pcbDrawObj

padEnd (PCB)

padEnd ::= pt

Description

padEnd specifies the coordinates of the end of a thermal touching a pad in a copper pour.

Used By

thermal

padNum

padNum ::= '(' 'padNum'
integerToken
')'

Description

padNum specifies the number of a pad. This value corresponds to the value of the **Pad Number** edit box in the Modify Pad dialog.

Used By

pad, padPinMapEntry

padPinMap

padPinMap ::= '(' 'padPinMap'
{ padPinMapEntry }
')'

Description

padPinMap maps pads in a pattern to pins in a component. The values in *padPinMap* correspond to the **Pin Designator** and **Pad #** columns in a P-CAD Library Manager spreadsheet.

Used By

attachedPattern

padPinMapEntry

padPinMapEntry ::= padNum
compPinRef

Description

padPinMapEntry provides a single mapping between a particular pad number and a particular component pin.

Used By

padPinMap

padShape (PCB)

padShape ::= '(' 'padShape'
(layerNumRef | layerType)
padShapeDfn
[spokeWidth]
[noCopperPourConnect]
')'

Description

padShape defines the size and shape of a pad on a particular layer or layer type. Each *padShape* corresponds to a single **Pad Definition** entry in the Modify Pad Style dialog.

Defaults

spokeWidth: 0
noCopperPourConnect: False

Used By
padStyleDef

padShapeDfn (PCB)

padShapeDfn ::= padShapeType
(shapeWidth shapeHeight |
[shapeSidesDfn rotation outsideDiam] shapeOutline)

Description
padShapeDfn defines the shape and size of a pad.

Notes
shapeOutline is required when *padShapeType* is *Polygon.*
[shapeSidesDfn rotation outsideDiam] are required for regular polygon shapes.

Used By
padShape

padShapeType (PCB)

padShapeType ::= '(' 'padShapeType'
(padViaShapeType | 'Target' | 'MtHole')
')'

Description
padShapeType defines the shape of a pad. This may be any of the shapes that are legal for vias, a target, or a mounting hole.

Used By
padShapeDfn

padStyleDef (PCB)

padStyleDef ::= '(' 'padStyleDef'
styleNameDef
holeDiam
[isHolePlated]
[holeOffset]
[useGlobalSwell]
[localSwell]
[startRange]
[endRange]
{ padShape }
')'

Description
padStyleDef defines a pad style. Each *padStyle* reflects the pad style information in the Modify Pad Style dialog and the Modify Hole Range dialog.

Each pad that is created references a defined pad style using *padStyleRef.* The pad style is defined by specifying a list of *padShapes,* each of which corresponds to a single **Pad Definition** entry in the Modify Pad Style dialog.

Defaults
isHolePlated: True
holeOffset: none
useGlobalSwell: True

localSwell: 0

Used By
styleDef

padStyleRef (PCB)

padStyleRef ::= '(' 'padStyleRef'
nameRef
')'

Description
padStyleRef provides a reference to a pad style which was previously defined in a *library* using *padStyleDef.*

Used By
pad

padToLineClearance (PCB)

padToLineClearance ::= '(' 'padToLineClearance'
dbNumber
')'

Description
padToLineClearance defines the pad-to-line clearance value for a particular layer. This value reflects the pad-to-line clearance value for a layer in the Design Rule Check Clearances dialog.

Used By
layerDef

padToPadClearance (PCB)

padToPadClearance ::= '(' 'padToPadClearance'
dbNumber
')'

Description
padToPadClearance defines the pad-to-pad clearance value for a particular layer. This value reflects the pad-to-pad clearance value for a layer in the Design Rule Check Clearances dialog.

Used By
layerDef

padViaShapeType (PCB)

padViaShapeType ::= ('Ellipse' | *'Oval'* | *'Rect'* | *'RndRect'* | *'Thrm2'* | 'Thrm2_90' | 'Thrm4' |
'Thrm4_45' | *'Direct'* | *'NoConnect'* | *'Polygon')*

Description
padViaShapeType indicates shapes that can be used for both pad and via definitions.

Notes
Several shapes (*Thrm2, Thrm2_90, Thrm4, Thrm4_45, Direct, and NoConnect)* are applicable as assigned to plane layers. The *NoConnect* shape acts to prohibit connections between a pad/via and plane layers, similar to the behavior of *noCopperPourConnect*.

Used By
padShapeType, viaShapeType

pageSize (SCH)

pageSize ::= ('size_A' | 'size_B' | 'size_C' | 'size_D' | 'size_E' | 'size_A0' | 'size_A1' | 'size_A2' |
'size_A3' | 'size_A4' | 'user' | 'scaleToFitPage')

Description

pageSize represents the size the print output should be scaled to. Valid sizes are imperial A through E, metric A4 through A0, user defined, or scale print to fit on a single page.

Used By

sheet

partNum

partNum ::= '(' 'partNum'
integerToken
')'

Description

partNum is used to specify a part number within a component.

Used By

attachedSymbol, compPin, symbol

pasteSwell (PCB)

pasteSwell ::= '(' 'pasteSwell'
dbNumber
')'

Description

pasteSwell specifies the global paste mask shrink value for a PCB design. This value reflects the value in the **Paste Mask Shrink** edit box in the Options Configure dialog.

Used By

pcbDesignHeader

pattern (PCB)

pattern ::= '(' 'pattern'
patternRef
refDesRef
location
[rotation]
[isFlipped]
[isFixed]
[patternGraphicsNameRef]
[isAutoSwapPatternGraphics]
{ patternGraphicsRef }
{ attr }
{ dimensionRef }
')'

Description

pattern represents a PCB pattern object created from a previously defined *patternDef* or *patternDefExtended* template. The pattern is created for a particular RefDes and references a specific pattern template.

Notes

patternGraphicsNameRef identifies the current pattern graphics. It is the name of an existing pattern graphics in the *patternGraphicsRef* list.

pattern attributes are handled as follows: If the position, textstyle, or visibility flag of the RefDes or Value attribute is different from the *patternDef*, it is written in the *pattern* with an empty value string; the value string is written in the *compInst*. All other pattern attributes are written in *pattern.*

Defaults

rotation: 0 degrees
isFlipped: False
isFixed: False
isAutoSwapPatternGraphics: False

Used By

pcbDrawObj

See Also

compInst, patternDef, patternDefExtended

patternAlias (PCB)

patternAlias ::= '(' 'patternAlias'
patternNameDef
patternNameRef
')'

Description

patternAlias defines a name alias *patternNameDef* for the pattern name *patternNameDef.*

Notes

There may be more than one *patternAlias* for a given P-CAD pattern name; additional *patternAliases* will be created for each pattern name alias.

Used By

library

patternDef (PCB)

patternDef ::= '(' 'patternDef'
patternNameDef
originalName
[multiLayer]
{ layerContents }
')'

Description

patternDef defines a template used to create *patterns.* It contains a *multiLayer* and a list of *layerContents* which contain the primitive objects that make up the pattern.

Notes

There may be more than one *patternDef* for a given P-CAD pattern name; additional *patternDefs* will be created if two patterns of the same name are not equivalent. The *patternNameDef* uniquely identifies each template, and *originalName* specifies the actual name of the pattern.

The *multiLayer* contains the objects within a pattern that exist on multiple layers, including pads. Each *layerContents* contains the objects that exist on a single layer; pads are not included here. *layerContents* productions are only generated for those layers that contain pattern information.

Patterns must have RefDes and Type attributes, and cannot contain connections, copper pours, DRC dots, fields, cutouts, or other patterns.

patternDef attributes are handled as follows: RefDes and Type attributes are always written; however, they are written to indicate their graphical information and placement only, and their value string is always written empty. Their textstyle and visibility are written if they are different from the Defaults. Value attributes are written if they

exist, and the value string is written empty. All visible attributes are written here as well; non-visible attributes are written in the *pattern.*

Used By

library

See Also

compInst, pattern, patternDefExtended

patternDefExtended (PCB)

patternDefExtended ::= '(' 'patternDefExtended'
patternNameDef
originalName
patternGraphicsNameRef
{ patternGraphicsDef }
[patternOrientationsMap]
')'

Description

patternDefExtended defines a template used to create *patterns.* It can optionally contain multiple *patternGraphicsDef*s, which can be mapped to specific orientations using a *patternOrientationsMap*. This allows the pattern graphics to automatically change in concert with selected pattern orientations.

Notes

patternGraphicsNameRef identifies the default pattern graphics. It is the name of an existing pattern graphics in the *patternGraphicsDef* list.

Used By

library

See Also

pattern

patternGraphicsDef (PCB)

patternGraphicsDef ::= '(' 'patternGraphicsDef'
patternGraphicsNameDef
[multiLayer]
{ layerContents }
')'

Description

patternGraphicsDef defines a template used for *pattern graphics.* Its *multiLayer* definition contains the pattern pads and its *layerContents* definitions contain the primitive objects and attributes that comprise the pattern.

Used By

patternDefExtended

patternGraphicsNameDef (PCB)

patternGraphicsNameDef ::= '(' 'patternGraphicsNameDef'
nameDef
')'

Description

patternGraphicsNameDef defines the name of a pattern graphics template.

Used By

patternGraphicsDef

patternGraphicsNameRef (PCB)

patternGraphidsNameRef ::= '(' 'patternGraphicsNameRef'
nameRef
')'

Description
patternGraphicsNameRef is used to refer to a previously defined *patternGraphicsNameDef.*

Used By
defaultPatternGraphicsName, patternGraphicsRef, patternOrientationAssignment

patternGraphicsRef (PCB)

patternGraphicsRef ::= '(' 'patternGraphicsRef'
patternGraphicsNameRef
{ attr }
')'

Description
patternGraphicsRef defines a pattern reference's pattern graphics. Optional *attr* definitions signify how the attributes of the pattern graphics are different from the *patternGraphicsDef* template.

Used By
pattern

See Also
patternGraphicsDef

patternName

patternName ::= '(' 'patternName'
stringToken
')'

Description
patternName indicates the name of the pattern attached to a component. This pattern name reflects the pattern name listed in the Edit Attach Pattern dialog in P-CAD Library Manager. *patternName* is also used to indicate the name of the pattern entered into the **Pattern** edit box in the Modify Part dialog in Schematic.

Used By
attachedPattern, compInst

patternNameDef (PCB)

patternNameDef ::= nameDef

Description
patternNameDef defines the name of a pattern template.

Used By
patternDef

patternNameRef (PCB)

patternNameRef ::= nameRef

Description
patternNameRef is used to refer to a previously-defined *patternNameDef.*

Used By
patternRef

patternNum

patternNum ::= '(' 'patternNum'
integerToken
')'

Description
patternNum specifies a unique number for each attached pattern.

Used By
attachedPattern

patternOrientation (PCB)

patternOrientation ::= '(' 'patternOrientation'
('pattern_orient_top_0' | 'pattern_orient_top_90' | 'pattern_orient_top_180' |
'pattern_orient_top_270' | 'pattern_orient_bottom_0' | 'pattern_orient_bottom_90' |
'pattern_orient_bottom_180' | 'pattern_orient_bottom_270')
')'

Description
PatternOrientation identifies a specific orientation of a pattern.

Used By
patternOrientationAssignment

See Also
PatternOrientationsMap, patternDefExtended

patternOrientationAssignment (PCB)

patternOrientation ::= '(' 'patternOrientationAssignment'
patternGraphicsNameRef
patternOrientation
')'

Description
PatternOrientationAssignment correlates a pattern orientation with its intended pattern graphics. This informs the system which pattern graphics to use for a specific orientation.

Used By
patternOrientationsMap

See Also
patternDefExtended

patternOrientationsMap (PCB)

patternOrientationsMap ::= '(' 'patternOrientationsMap'
{ patternOrientationAssignment }
')'

Description
patternOrientationsMap defines a list of *patternOrientationAssignment* definitions to inform the system which pattern graphics to use for various orientations.

Used By
patternDefExtended

patternRef (PCB)

patternRef ::= '(' 'patternRef'
patternNameRef
')'

Description
patternRef provides a reference to a pattern previously defined in a library using *patternDef*.

Used By
pattern

pcbDesign (PCB)

pcbDesign ::= '(' 'pcbDesign'
pcbDesignNameDef
pcbDesignHeader
[globalAttrs]
{ layerDef }
multiLayer
{ layerContents }
[pcbPrintSettings]
[drillSymSettings]
[gerberSettings]
[ncDrillSettings]
[programState]
[layerSets]
[layerPairs]
[reportSettings]
[odbSettings]
[layersStackup]
')'

Description
pcbDesign provides design-specific and placement data for a PCB design, such as global attributes, layer definitions, layer contents, layer sets, report settings, and program state information. Component and net information is located in the *netlist* and *library* sections.

Used By
PCAD_ASCII

See Also
library, netlist

pcbDesignHeader (PCB)

pcbDesignHeader ::= '(' 'pcbDesignHeader'
workspaceSize
gridDfns
designInfo
solderSwell
pasteSwell
planeSwell
[refPointSize]
[infoPointSize]
[gluePointSize]
[pickPointSize]
[testPointSize]
[refPointSizePrint]
[infoPointSizePrint]
[gluePointSizePrint]

[pickPointSizePrint]
[testPointSizePrint]
[pourOrder]
[solderFlowDirection]
[autoPlowCopperPours]
[globalCopperPourCutoutBackoffFlag]
')'

Description

pcbDesignHeader provides general information about a PCB design, including the workspace size, the grid definitions, the information in the File Design Info dialog, global swell values and pour order.

Defaults

solderFlowDirection: SolderFlowTopToBottom

autoPlowCopperPours: True
globalCopperPourCutoutBackoffFlag: False

Used By

pcbDesign

pcbDesignNameDef (PCB)

pcbDesignNameDef ::= nameDef

Description

pcbDesignNameDef uniquely identifies a *pcbDesign* within a file.

Used By

pcbDesign

pcbDrawObj (PCB)

pcbDrawObj ::= (arc | attr | copperPour | detail | diagram | field | fromTo | gluePoint | infoPoint | line | lineKeepOut | metafile | pad | pattern | pickPoint | plane | poly | polyCutOut | polyKeepOut | refPoint | room | table | testPoint | text | triplePointArc | via)

Description

A *pcbDrawObj* is a PCB primitive object. These objects might appear in a design or in a pattern; restrictions on which objects may appear in each are detailed in the appropriate sections.

Used By

layerContents, multiLayer

pcbPoly (PCB)

pcbPoly ::= '(' 'pcbPoly'
{ pt }
[isFlipped]
[netNameRef]
[isCopperTie]
[tieNet]
[fillets]
')'

Description

pcbPoly is the new definition for all polygonal objects in PCB-based designs. It obsoletes *poly* in these applications. *pcbPoly* can include both tie and fillet information.

Used By

copperPour95

plane
polyCutOut
polyKeepOut

pcbPrintSettings (PCB)

pcbPrintSettings ::= '(' 'pcbPrintSettings'
{ printQueueEntry }
')'

Description
pcbPrintSettings lists the print jobs defined in a PCB design. It contains all of the information specified in the dialogs accessed from the File Print dialog.

Used By
pcbDesign

pickPoint (PCB)

pickPoint ::= '(' 'pickpoint'
location
[isFlipped]
[isVisible]
')'

Description
pickPoint represents a pick-and-place point in a PCB design.

Defaults
isFlipped: False
isVisible: True

Used By
pcbDrawObj

pickPointSize (PCB)

pickPointSize ::= '(' 'pickPointSize'
dbNumber
')'

Description
pickPointSize describes the size of pick point.

Used By
pcbDesignHeader

pickPointSizePrint (PCB)

pickPointSizePrint ::= '(' 'pickPointSizePrint'
dbNumber
')'

Description
pickPointSizePrint describes the print size of pick point.

Used By
pcbDesignHeader

pin (SCH)

pin ::= '(' 'pin'
pinNum
location
[rotation]
[isFlipped]
[pinLength]
[outsideStyle]
[outsideEdgeStyle]
[insideEdgeStyle]
[insideStyle]
[pinDisplay]
[pinDesignatorProperty]
[pinNameProperty]
')'

Description
pin represents a symbol pin object in a schematic design.

Defaults
rotation: 0 degrees
isFlipped: False
pinLength: 300 mils (normal length)
outsideStyle: none
outsideEdgeStyle: none
insideEdgeStyle: none
insideStyle: none

Used By
schDrawObj

pinDesignatorProperty (SCH)

pinDesignatorProperty ::= '(' 'pinDes'
tangoText
')'

Description
pinDesignatorProperty specifies the display properties of a pin designator on a symbol pin.

Used By
pin

pinDisplay (SCH)

pinDisplay ::= '(' 'pinDisplay'
[dispPinDes]
[dispPinName]
')'

Description
pinDisplay specifies the display status of the pin designator and pin name for a symbol pin. These values represent the status of the **Display** box in the Modify Pin dialog.

Defaults
dispPinDes: True
dispPinName: False

Used By
pin

pinEq

pinEq ::= '(' 'pinEq'
integerToken
')'

Description
pinEq specifies the pin equivalence value of a component pin. This value corresponds to the value of a cell in the **Pin Eq** column in a P-CAD Library Manager spreadsheet.

Used By
compPin

pinLength (SCH)

pinLength ::= '(' 'pinLength'
dbNumber
')'

Description
pinLength specifies the length of a symbol pin. This value corresponds to the status of the **Length** box in the Modify Pin dialog.

Used By
pin

pinName

pinName ::= '(' 'pinName'
stringToken
')'

Description
pinName specifies the name of a component pin. This value corresponds to the value of a cell in the **Pin Name** column in a P-CAD Library Manager spreadsheet.

Used By
compPin

pinNameProperty (SCH)

pinNameProperty ::= '(' 'pinDes'
tangoText
')'

Description
pinNameProperty specifies the display properties of a pin name on a symbol pin.

Used By
pin

pinNum (SCH)

pinNum ::= '(' 'pinNum'
integerToken
')'

Description
pinNum specifies the number of a symbol pin. This value corresponds to the value of the **Pin Number** edit box in the Modify Pin dialog.

Used By

pin

pinType

pinType ::= '(' 'pinType'
('Unknown' | 'Passive' | 'Input' | 'Output' | 'Bidirectional' |'OpenH' | 'OpenL' | 'PassiveH' |
'PassiveL' | 'ThreeState' |'Power')
')'

Description

pinType specifies the type of a component pin. This value corresponds to the value of a cell in the **Elec Type** column in a P-CAD Library Manager spreadsheet.

Used By

compPin

plane (PCB)

plane ::= '(' 'planeObj'
width
planeOutline
[netNameRef]
')'

Description

plane represents a plane object in PCB design.

Notes

width defines the width of the lines that form the polygonal outline of the plane.

Used By

pcbDrawObj

planeNetNameRef (PCB)

planeNetNameRef ::= netNameRef

Description

planeNetNameRef specifies the net name of a plane layer. It must refer to a net name that was previously defined in the *netlist* section. The net name referred to here corresponds to the net name entered in the **Net Name** combo box in the Plane Layer Net Name dialog.

Used By

layerDef

planeOutline (PCB)

planeOutline ::= '(' 'planeOutline'
{ pt }
')'

Description

planeOutline lists the vertices of the outline of a plane object.

Used By

plane

planeSwell (PCB)

planeSwell ::= '(' 'planeSwell'
dbNumber
')'

Description
planeSwell specifies the global plane swell value for a PCB design. This value reflects the value in the **Plane Swell** edit box in the Options Configure dialog.

Used By
pcbDesignHeader

plowGrid (PCB)

plowGrid ::= '(' ' plowGrid '
gridString
')'

Description
plowGrid is a string that describes the design file plow grid setting, as listed in the Options Grids dialog. The string includes the grid units.

Used By
gridState

plowViaGrid (PCB)

plowViaGrid ::= '(' ' plowViaGrid '
gridString
')'

Description
plowViaGrid is a string that describes the design file plow via grid setting, as listed in the Options Grids dialog. The string includes the grid units.

Used By
gridState

poly

poly ::= '(' 'poly'
{ pt }
[isFlipped]
[netNameRef]
[isCopperTie]
[tieNetValue]
')'

Description
poly defines a polygonal shape. It is used both to represent polygon objects in a design, and to define the vertices for objects such as *fillPoly, polyCutOut,* and *polyKeepOut.*

Notes
A *poly* in a *library* section should not have a *netNameRef* or *isCopperTie;* nor should a *poly* in a Schematic design.

Defaults
isFlipped: False
isCopperTie: False

Used By
fillPoly, pcbDrawObj, polyCutOut, polyKeepOut, room, schDrawObj

polyCutOut (PCB)

polyCutOut ::= '(' 'polyCutOut'
poly
')'

Description
polyCutOut represents a polygonal cutout object in a PCB design.

Used By
pcbDrawObj

polyDescriptor (PCB)

polyDescriptor ::= '(' 'polyDescriptor'
shapeSidesDfn
rotation
outsideDiam
')'

Description
polyDescriptor represents a general description of a regular polygon shape.

Used By
polyShape

polyKeepOut (PCB)

polyKeepOut ::= '(' 'polyKeepOut'
poly
')'

Description
polyKeepOut represents a polygon keepout object in a PCB design.

Used By
pcbDrawObj

polyPoint (PCB)

polyPoint ::= '(' 'polyPoint'
xPoint
yPoint
angle
')'

Description
polyPoint represents a vertex of an enhancedPolygon. *xPoint* and *yPoint* specify the location, *angle* specifies if the following edge is straight (angle of 0), or curved (angle of 1 to 359 in tenth degree values).

Used By
enhancedPolygon

polyShape (PCB)

polyShape ::= '(' 'polyShape'
[polyDescriptor]
shapeOutline
')'

Description

polyShape represents a polygon shape.

Used By

apertureDef

port (SCH)

port ::= '(' 'port'
location
portType
[portPinLength]
[netNameRef]
[rotation]
[isFlipped]
')'

Description

port represents a port in a schematic design. The port has a specific shape defined in *portType*. If the port is part of a net, it will have a net name reference.

Defaults

portPinLength: PortPinLong
rotation: 0 degrees
isFlipped: False
netNameRef: the net name text will reference the text style "(PortStyle)", if this style is defined. Otherwise, it will reference the default text style "(WireStyle)".

Used By

schDrawObj

portPinLength (SCH)

portPinLength ::= '(' 'portPinLength'
('PortPinLong' | 'PortPinShort')
')'

Description

portPinLength defines the length of a pin on a port object in a schematic design.

Used By

port

portType (SCH)

portType ::= '(' 'portType'
('NoAngle_Sgl_Horz' | 'NoAngle_Sgl_Vert' |'NoAngle_Dbl_Horz' | 'NoAngle_Dbl_Vert' |
'LeftAngle_Sgl_Horz' | 'LeftAngle_Sgl_Vert' |'LeftAngle_Dbl_Horz' | 'LeftAngle_Dbl_Vert' |
'RightAngle_Sgl_Horz' | 'RightAngle_Sgl_Vert' |'RightAngle_Dbl_Horz' | 'RightAngle_Dbl_Vert' |
'BothAngle_Sgl_Horz' | 'BothAngle_Sgl_Vert' | 'BothAngle_Dbl_Horz' | 'BothAngle_Dbl_Vert'
|'VertLine_Sgl_Horz' | 'VertLine_Sgl_Vert' |
'VertLine_Dbl_Horz' | 'VertLine_Dbl_Vert' |'NoOutline_Sgl_Horz' | 'NoOutline_Sgl_Vert' |
'NoOutline_Dbl_Horz' | 'NoOutline_Dbl_Vert')
')'

Description

portType defines the shape, the number of pins, and the pin orientation, for a port object in a schematic design.

Notes

The shapes for the various types are defined as :

portType	shape
NoAngle_xxxxxx	rectangular
LeftAngle_xxxxx	rectangular, but left side angles out
RightAngle_xxxxx	rectangular, but right side angles out
BothAngle_xxxxx	rectangular, but left side angles out
VertLine_xxxxx	vertical lines drawn on the left and right side
NoOutline_xxxxx	no shape is drawn

The port will have a single pin if *portType* contains the letters *'Sgl',* and it will have two pins if *portType* contains the letters *'Dbl'*.

The pin or pins attached to the port will be oriented horizontally if *portType* contains the letters *'Horz',* and will be oriented vertically if *portType* contains the letters *'Vert'.*

Used By

port

pourBackoff (PCB)

pourBackoff ::= '(' 'pourBackoff'
dbNumber
')'

Description

pourBackoff defines the backoff value between a copper pour and any objects inside the copper pour outline. This value corresponds to the value in the **Pour Backoff** edit box in the Modify Copper Pour dialog.

Used By

copperPour

pourEnd (PCB)

pourEnd ::= pt

Description

pourEnd specifies the coordinates of the end of a thermal touching a copper pour.

Used By

thermal

pourOrder (PCB)

pourOrder ::= '(' pourOrder
booleanToken
')'

Description

This parameter is specifies whether *the specific from smallest to the largest pour order has been selected. New in V16.*

Used By

pcbDesignHeader

See Also

pcbDesignHeader

pourOutline (PCB)

pourOutline ::= '(' 'pourOutline'
{ pt }

')'

Description

pourOutline lists the vertices of the outline of a copper pour.

Used By

copperPour

See Also

copperPour, fillPoly

pourSmoothness (PCB)

pourSmoothness ::= '(' 'pourSmoothness'
numberToken
')'

Description

pourSmoothness defines how many edges to use when appoximating the backoff polygon of objects under a *copperPour95*. Current values are 1 (low, 8 sides), 2 (medium, 12 sides, and 3 (high, 16 sides).

Used By

copperPour95

pourSpacing (PCB)

pourSpacing ::= '(' 'pourSpacing'
dbNumber
')'

Description

pourSpacing defines the separation between fill or hatch lines in a copper pour. This value corresponds to the value in the **Line Spacing** edit box in the Modify Copper Pour dialog.

Used By

copperPour

pourType (PCB)

pourType ::= '(' 'pourType'
('SolidPour' | 'HorizPour' | 'VertPour' | 'Hatch45Pour' | 'Hatch90Pour')
')'

Description

pourType specifies the pour pattern for a copper pour. This value corresponds to the status of the **Pour Pattern** radio buttons in the Modify Copper Pour dialog.

Used By

copperPour

powerTableInfo (SCH)

powerTableInfo ::= '(' 'powerTableInfo'
allPins
allComponents
')'

Description

powerTableInfo is used to specify information specific to a power table.

Used By

tableInfo

printQueueEntry (PCB)

printQueueEntry ::= '(' 'printQueueEntry'
entryName
[layerList]
[orderedLayerList]
[isRotated]
[isDraft]
[isThinStrokeText]
[scaling]
[outputItem]
[outputCutout]
[outputGluedot]
[outputPickPlace]
[scaleToFitPage]
[entireDesign]
[printRegion]
')'

Description

printQueueEntry defines a single PCB print job. The values specified in *printQueueEntry* correspond to the settings in the **Print Job Selections** box in the Setup Print Jobs dialog.

Notes

layerList specifies layers to be printed in default order (layer number order). *orderedLayerList* specifies layers to be printed in specific order. If both *layerList* and *orderedLayerList* are specified, then *orderedLayerList* takes precedence. If neither are specified, then the *printQueueEntry* contains no layers.

Defaults

layerList: None
orderedLayerList: None
isRotated: False
isDraft: False
scaling: 1.0
outputCutout: False
outputGluedot: False
outputPickPlace: False
scaleToFitPage: False
entireDesign: True
printRegion: A rectangle of zero width and height, located at (0,0).

Used By

pcbPrintSettings

printRegion

printRegion ::= '(' 'printRegion'
pt
pt
')'

Description

printRegion describes a specified area to be printed. The first point describes the lower left corner of a rectangle, the second point describes the opposing corner.

Used By

printQueueEntry, sheet

program

program ::= '(' 'program'
programName
programVersion
')'

Description
program identifies the application that created an P-CAD ASCII file.

Used By
written

programName

programName ::= stringToken

Description
programName identifies the name of the application that created an P-CAD ASCII file.

Used By
program

programState

programState ::= '(' 'programState'
[layerState]
gridState
ecoState
[onlineDrcState]
[currentTextStyle]
[currentPadStyle]
[currentViaStyle]
')'

Description
programState defines certain states of the application when the P-CAD ASCII file was generated, including the current layer, the grid settings, the status of the ECO recorder, the Online DRC state, the current text style, the current pad style, and the current via style.

Notes
layerState is not written for single-sheet schematic designs. *onlineDrcState is* not used by P-CAD Schematic. c*urrentPadStyle* and *currentViaStyle* are not used by P-CAD Schematic.

Used By
pcbDesign, schematicDesign

programVersion

programVersion ::= stringToken

Description
programVersion identifies the version of the application that created an P-CAD ASCII file.

Used By
program

pt

pt ::= '(' 'pt'
xPoint

yPoint
')'

Description
pt specifies a specific coordinate in an P-CAD design.

Used By
busPoint, dimension, dimensionOffsets, location, padEnd, poly, pourEnd, pourOutline, printRegion, vertex

radius

radius ::= '(' 'radius'
dbNumber
')'

Description
radius specifies the radius of an arc.

Used By
arc

refDesDef

refDesDef ::= nameDef

Description
refDesDef defines the name of a reference designator.

Used By
compInst

refDesNameRef

refDesNameRef ::= nameRef

Description
refDesNameRef is used to refer to a previously-defined *refDesDef.*

Used By
node, refDesRef

refDesPrefix

refDesPrefix ::= '(' 'refDesPrefix'
stringToken
')'

Description
refDesPrefix specifies the RefDes prefix for a component. It corresponds to the value in the **RefDes Prefix** edit box in the Component Save As dialog in P-CAD Library Manager.

Used By
allComponents, compHeader

refDesRef

refDesRef ::= '(' 'refDesRef'
refDesNameRef
')'

Description

refDesRef is used to refer to a reference designator previously defined in a netlist section.

Used By

pattern, room, symbol

refPoint

refPoint ::= '(' 'refPoint'
location
')'

Description

refPoint represents a reference point object.

Notes

refPoint is only used to represent reference point objects that are not part of a symbol or pattern. Since the location of all objects in a symbol or pattern are specified relative to the reference point, a reference point is implicitly assumed to be part of each symbol and attern at location (0, 0) and is not explicitly defined.

Used By

pcbDrawObj, schDrawObj

refPointSize

refPointSize ::= '(' 'refPointSize'
dbNumber
')'

Description

refPointSize describes the size of ref point.

Used By

pcbDesignHeader, schematicDesignHeader

refPointSizePrint

refPointSizePrint ::= '(' 'refPointSizePrint'
dbNumber
')'

Description

refPointSizePrint describes the print size of ref point.

Used By

pcbDesignHeader, schematicDesignHeader

relOrigin

relOrigin ::= '(' 'relOrigin'
location
')'

Description

relOrigin is used to specify the origin of the relative grid. This setting corresponds to the values in the **Relative Grid Origin** box in the Options Grids dialog.

Used By

gridDfns

reportColumnWidth

reportColumnWidth ::= '(' reportColumnWidth'
integerToken
')'

Description
reportColumnWidth specifies the number of characters across each page of a report.

Used By
reportDefinition

reportDataFileName

reportDataFileName ::= '(' 'reportDataFileName'
stringToken
')'

Description
reportExtension is the file name of the external data file used when import MRP data into the bill of materials report.

Used By
reportDefinition

reportDefinition

reportDefinition ::= '(' reportDefinition'
reportName
reportExtension
[reportShowFlag]
reportType
[reportUserDefined]
reportLinesPerPage
reportColumnWidth
[reportUseHeader]
reportHeader
[reportUseFooter]
reportFooter
[reportUseDesignInfo]
[reportShowDate]
[reportPaginate]
[reportDataFileName]
[reportMapFileName]
[reportVariantName]
{reportFieldsSections}
')'

Description
reportDefinition defines the settings for a specific report created through the File Reports dialog.

Used By
reportDefinitions

reportDefinitions

reportDefinitions ::= '(' 'reportDefinitions'
{reportDefinition}
')'

Description
reportDefinitions is a variable length list of *reportDefinition*.

Used By
reportSettings

reportDestination

reportDestination ::= '(' 'reportDestination'
(DESTINATIONSCREEN |
DESTINATIONPRINTER |
DESTINATIONFILE)
')'

Description
ReportDestination is the output destination of the reports.

Used By
reportSettings

reportExtension

reportExtension ::= '(' 'reportExtension'
stringToken
')'

Description
reportExtension is the file extension for a specific report.

Used By
reportDefinition

reportField

reportField ::= '(' 'reportField'
reportFieldName
reportFieldType
reportFieldSortOrder
reportFieldSortType
reportFieldShowFlag
reportFieldColumnWidth
reportFieldConditions
')'

Description
reportField defines a report field's specific data and its conditions.

Used By
reportFieldsSections

reportFieldColumnWidth

reportFieldColumnWidth ::= '(' 'reportFieldColumnWidth'
integerToken
')'

Description
reportFieldColumnWidth defines the number of characters for displaying a report field's data.

Used By
reportField

reportFieldCondition

reportFieldCondition ::= '(' 'reportFieldCondition'
stringToken
')'

Description
reportFieldCondition defines the selection criteria for displaying a report field's data.

Used By
reportFieldConditions

reportFieldConditions

reportFieldConditions ::= '(' 'reportFieldConditions'
{reportFieldCondition}
')'

Description
reportFieldConditions is a variable sized list of *reportFieldCondition*.

Used By
reportField

reportFieldName

reportFieldName ::= '(' 'reportFieldName'
stringToken
')'

Description
reportFieldName is the name of a specific *reportField*.

Used By
reportField

reportFields

reportFields ::= '(' 'reportFields'
{reportField}
')'

Description
reportFields is a variable sized list of *reportField*.

Used By
reportFieldsSections

reportFieldShowFlag

reportFieldShowFlag ::= '(' 'reportFieldShowFlag'
[booleanToken]
')'

Description
ReportFieldShowFlag defines the flag whether to show the *reportField* in question during output.

Defaults
reportFieldShowFlag: FALSE

Used By
reportField

reportFieldsSections

reportFieldsSections ::= '(' 'reportFieldsSections'
{reportFields}
')'

Description
reportFieldsSections is a variable sized list of *reportFields*.

Used By
reportDefinition

reportFieldSortOrder

reportFieldSortOrder ::= '(' 'reportFieldSortOrder'
{integerToken}
')'

Description
ReportFieldSortOrder defines the sorting order for a *reportField* with respect to other *reportField*.

Used By
reportField

reportFieldSortType

reportFieldSortType ::= '(' 'reportFieldSortType'
(NONE |
ASCENDING |
DESCENDING)
')'

Description
ReportFieldSortType defines how a *reportField* should be sorted.

Used By
reportField

reportFieldType

reportFieldType ::= '(' 'reportFieldType'
(*PropertyTypeUser* |
PropertyTypeAttribute |
PropertyTypeLocation |
PropertyTypeBoundRect |
PropertyTypeComponentName |
PropertyTypeComponentLibrary |
PropertyTypeComponentType |
PropertyTypeCurrentFootprint |
PropertyTypeNumberOfPads |
PropertyTypeNumberOfPins |
PropertyTypeNumberOfParts |
PropertyTypeHomogeneous |
PropertyTypeAlphaNumeric |
PropertyTypeHasIEEE |
PropertyTypeHasDemorgan |
PropertyTypeRefDesPrefix |
PropertyTypePatternName |
PropertyTypeAlias |
PropertyTypeCreateDate |

PropertyTypeModifyDate |
PropertyTypeVerifyDate |
PropertyTypeDCode |
PropertyTypeShape |
PropertyTypeDimensionX |
PropertyTypeDimensionY |
PropertyTypeDiameter |
PropertyTypeType |
PropertyTypeAngle |
PropertyTypeOffsetX |
PropertyTypeOffsetY |
PropertyTypeRefDes |
PropertyTypeLayer |
PropertyTypeLocationX |
PropertyTypeLocationY |
PropertyTypeRotation |
PropertyTypeFixed |
PropertyTypeErrorNumber |
PropertyTypeError |
PropertyTypeCompValue |
PropertyTypeRoom |
PropertyTypeBoardSide |
PropertyTypeComponents |
PropertyTypeAttrKeyword |
PropertyTypeAttrValue |
PropertyTypeNetName |
PropertyTypeCount |
PropertyTypeSheetNumber |
PropertyTypeUnusedParts)
')'

Description

ReportFieldSortType defines how a *reportField* should be sorted.

Used By

reportField

reportFooter

reportFooter ::= '(' 'reportFooter'
stringToken
')'

Description

reportFooter defines the footer string for a *reportDefinition*.

Used By

reportDefinition

reportHeader

reportHeader ::= '(' 'reportHeader'
stringToken
')'

Description

reportFooter defines the header string for a *reportDefinition*.

Used By

reportDefinition

reportLinesPerPage

reportLinesPerPage ::= '(' 'reportLinesPerPage'
integerToken
')'

Description
reportLinesPerPage defines the number of lines each page will contain for a *reportDefinition*.

Used By
reportDefinition

reportMapFileName

reportMapFileName ::= '(' 'reportMapFileName'
stringToken
')'

Description
reportMapFileName is the name of the external map file when importing data into the bill of materials report.

Used By
reportDefinition

reportName

reportName ::= '(' 'reportName'
stringToken
')'

Description
reportName is the name of the *reportDefinition*.

Used By
reportDefinition

reportPaginate

reporPaginate ::= '(' 'reportPaginateFlag'
[booleanToken]
')'

Description
reportPaginate defines the flag for using pagination in a report.

Defaults
ReportPaginate: FALSE

Used By
reportDefinition

reportSettings

reportSettings ::= '(' 'reportSettings'
reportStyle
reportDestination
appPoint
reportDefinitions
')'

Description

reportSettings is used to specify settings for the custom reports created through the File Reports dialog.

Used By
PcbDesign/schematicDesign

reportShowDate

reportShowDate ::= '(' 'reportShowDate'
[booleanToken]
')'

Description
reportShowDate defines whether the date should be shown in a report.

Defaults
reportShowDate: FALSE

Used By
reportDefinition

reportShowFlag

reportShowFlag ::= '(' 'reportShowFlag'
[booleanToken]
')'

Description
reportShowFlag defines whether the report should be output.

Defaults
reportShowFlag: FALSE

Used By
reportDefinition

reportStyle

reportStyle ::= '(' 'reportStyle'
(reportStyleComma |
reportStyleAccel |
reportStyleWord)
')'

Description
reportStyle defines which style of report should be output.

Used By
reportDefinition

reportType

reportType ::= '(' 'reportType'
(reportTypeApertures |
reportTypeAttributes |
reportTypeBillOfMaterials |
reportTypeComponentsLocations |
reportTypeDRCErrors |
reportTypeGlobalNets |
reportTypeGlueDots |
reportTypeLastRefdes |
reportTypeLibrary |

reportTypePartsLocations |
reportTypePartsUsage |
reportTypePickAndPlace |
reportTypeRooms |
reportTypeStatistics |
reportTypeTestPoint|
reportTypeVariant)
')'

Description
reportType defines which type of report the *reportDefinition* is.

Used By
reportDefinition

reportUserDefined

reportUserDefined ::= '(' 'reportUserDefined'
[booleanToken]
')'

Description
reportStyle defines whether the *reportDefinition* is a custom user-defined.

Defaults
reportUserDefined: FALSE

Used By
reportDefinition

reportUseDesignInfo

reportUseDesignInfo ::= '(' 'reportUseDesignInfo'
[booleanToken]
')'

Description
reportUseDesignInfo defines whether the report should include design information on output.

Defaults
reportUseDesignInfo: FALSE

Used By
reportDefinition

reportVariantName

reportVariantName ::= '(' 'reportVariantName'
stringToken
')'

Description
reportVariantName is the name of the variant chosen to generate the report.

Used By
reportDefinition

reportUseFooter

reportUseFooter ::= '(' 'reportUseFooter'
[booleanToken]

')'

Description
reportUseFooter defines whether the report should include the report footer on output.

Defaults
'reportUseFooter': FALSE

Used By
reportDefinition

reportUseHeader

reportUseHeader ::= '(' 'reportUseHeader'
[booleanToken]
')'

Description
reportUseHeader defines whether the report should include the report header on output.

Defaults
'reportUseHeader': FALSE

Used By
reportDefinition

revisionNoteDef

revisionNoteDef ::= '(' 'revisionNote'
noteNum
noteValue
')'

Description
revisionNoteDef specifies a specific numbered revision note.

Notes
Revision notes can only be specified by the user with the P-CAD Document Toolbox option.

Used By
fieldSet

revisionNoteRef

revisionNoteRef ::= '(' 'revisionNote' noteNum ')'

Description
revisionNoteRef allows a field to refer to a previously-defined revision note by its number.

Used By
fieldType

room

room ::= '(' nameDef
[roomFillPattern]
[roomPlacementSide]
[isFixed]
[roomInclusionList]
[roomAttrMgr]
poly

')'

Description

room represents a PCB room object

Defaults

roomFillPattern: Clear
roomPlacementSide: roomTopOrBottom
isFixed: False

Used By

pcbDrawObj

roomAttrMgr

roomAttrMgr ::= '(' 'roomAttrMgr'
{attr}
')'

Description

roomAttrMgr indicates the attributes list attached to the room. This can be empty.

Used By

room

roomFillPattern

roomFillPattern::= '(' roomFillPattern 'clear'|'solid'|'hatched' ')'

Description

roomFillPattern indicates the fillPattern of the room. Default value for the *roomFillPattern* is Clear.

Used By

room

roomInclusionList

roomInclusionList ::= '(' roomInclusionList
{refDesRef}
')'

Description

roomInclusionList designates the *refDefRef* of the componets attached to the rooms. This can be empty, and is the list of components *refDefRef*.

Used By

room

roomPlacementSide

roomPlacementSide ::= '(' 'roomPlacementSide'
('roomTop' | 'roomBottom' | 'roomTopOrBottom')
')'

Description

roomPlacementSide indicates the PlacementSide of the room.

Used By

room

rotation

rotation ::= '(' 'rotation'
numberToken
')'

Description
rotation specifies the amount of rotation of an object. Rotations are specified in degrees, with a resolution of 0.1 degree, and are always measured in a counterclockwise direction relative to the positive X-axis.

Used By
apertureDef, attr, dimension, field, fillPoly, ieeeSymbol, pad, pattern, pin, symbol, table, text, via

row

row ::= '(' 'row'
stringToken
')'

Description
row defines the contents of a row in a table.

Used By
column

scaleFactor

scaleFactor ::= '(' 'scaleFactor'
numberToken
')'

Description
scaleFactor designates a scaling factor. In Schematic, *scaleFactor* corresponds to the **User Scale Factor** edit box in the Page Setup dialog for printed output.

Used By
diagram, metafile, sheet

scaling

scaling ::= '(' 'scaling'
(sheetSize | 'user' numberToken)
')'

Description
scaling designates the scaling factor to be used for printed output. In Schematic, the scaling factor can be a specific sheet size or a user-defined value. In PCB, the scaling factor is always a user-defined value. The scaling value corresponds to the **Image Scale** radio buttons in the Page Setup dialog in P-CAD Schematic, and the **Scale** edit box in the Setup Print Jobs dialog in P-CAD PCB.

NOTE: scaling is no longer used in Schematic but is maintained for backward compatability.

Used By
printQueueEntry, schematicPrintSettings

scaleToFitPage

scaleToFitPage ::= '(' 'scaleToFitPage' booleanToken ')'

Description

scaleToFitPage designates the printed output should be scaled to fit on a single page. This applies on a individual sheet, or print job basis.

Used By

printQueueEntry, sheet

schDrawObj (SCH)

schDrawObj ::= (arc | attr | bus | busEntry | field | ieeeSymbol | infoPoint | junction | line | pin | poly | port | refPoint | symbol | table | text | triplePointArc| wire | xRef)

Description

A *schDrawObj* is a schematic primitive object. These objects might appear on a sheet, title sheet, or in a symbol; restrictions on which objects may appear on each are detailed in the appropriate sections.

Used By

sheet, symbolDef, titleSheet

sheetOnlyNets (SCH)

sheetOnlyNets ::= '(' 'SheetOnlyNets'
booleanToken
')'

Description

sheetOnlyNets specifies that a net index table should include only those nets that reside on the sheet on which the table is placed.

Used By

table

schematicDesign (SCH)

schematicDesign ::= '(' 'schematicDesign'
schematicDesignNameDef
schematicDesignHeader
[globalAttrs]
[titleSheet]
{ sheet }
[schPrintSettings]
[programState]
[reportSettings]
{variant}
')'

Description

schematicDesign provides design-specific and placement data for a schematic design, such as global attributes, sheet contents, and program state information. Component and net information is located in the *netlist* and *library* section.

Used By

PCAD_ASCII

See Also

library, netlist

schematicDesignHeader (SCH)

schematicDesignHeader ::= '(' 'schDesignHeader'
workspaceSize

gridDfns
designInfo
[refPointSize]
[erc point size]
[junction size]
[refPointPrintSize]
[ercPointPrintSize]
[junctionPointPrintSize]
')'

Description

schDesignHeader provides general information about a schematic design, including the workspace size, the grid definitions, and the information in the File Design Info dialog.

Used By

schematicDesign

schematicDesignNameDef (SCH)

schematicDesignNameDef ::= nameDef

Description

schematicDesignNameDef uniquely identifies a *schematicDesign* within a file.

Used By

schematicDesign

schematicPrintSettings (SCH)

schematicPrintSettings ::= '(' 'schematicPrintSettings'
sheetList
[scaling]
[offset]
[isRotated]
[drawBorder]
')'

Description

schematicPrintSettings describes the state of the print settings in a schematic design. These settings are specified in the File Print and Page Setup dialogs. The *sheetList* indicates the schematic sheets that are to be printed.
NOTE: scaling, offset, isRotated, and drawBorder are optional and are maintained only for backward compability. Their values are ignored when read. They have been moved into *sheet*.

Used By

schematicDesign

secondNumber

secondNumber ::= integerToken

Description

secondNumber denotes the seconds portion of a time value, from 0 to 59.

Used By

time

severity

severity ::= '(' 'severity'
integerToken

')'

Description
This parameter is reserved for future use and is not currently used.

Used By
infoPoint

shapeHeight (PCB)

shapeHeight ::= (shapeHeightDfn | insideDiam)

Description
shapeHeight defines the height of a pad or via shape. For thermals, this represents the inside diameter of the thermal; for other shapes, this is the actual height of the shape.

Used By
padShapeDfn, viaShapeDfn

shapeHeightDfn (PCB)

shapeHeightDfn ::= '(' 'shapeHeight'
dbNumber
')'

Description
shapeHeightDfn defines the height of a non-thermal pad or via shape.

Used By
shapeHeight

shapeOutline

shapeOutline ::= '(' 'shapeOutline'
{ pt }

')'

Description
shapeOutline defines a polygon shape.

Used By
polyShape

shapeSidesDfn

shapeSidesDfn ::= '(' 'sides'
integerToken
')'

Description
shapeSidesDfn indicates a number of sides.

Used By
polyDescriptor

shapeWidth (PCB)

shapeWidth ::= (shapeWidthDfn | outsideDiam)

Description

shapeWidth defines the width of a pad or via shape. For thermals, this represents the outside diameter of the thermal; for other shapes, this is the actual width of the shape.

Used By

padShapeDfn, viaShapeDfn

shapeWidthDfn (PCB)

shapeWidthDfn ::= '(' 'shapeWidth'
dbNumber
')'

Description

shapeWidthDfn defines the width of a non-thermal pad or via shape.

Used By

shapeWidth

sheet (SCH)

sheet ::= '(' 'sheet'
sheetNameDef
sheetNum
[titleSheet]
[fieldSetRef]
{ schDrawObj }
[drawBorder]
[entireDesign]
[isRotated]
[pageSize]
[scaleFactor]
[offset]
[printRegion]
[sheetOrderNum]
')'

Description

sheet defines a schematic sheet. The sheet is defined by name and number, then contains a list of *schDrawObjs* that appear on that sheet. *TitleSheet, fieldSetRef, schDrawObj*, *drawBorder*, *entireDesign*, *isRotated*, *pageSize*, *scaleFactor*, *offset*, and *printRegion* are all optional.

Notes

TitleSheet overrides the *schematicDesign*'s *titleSheet*. If *titleSheet* is absent, the *sheet* receives the *schematicDesign*'s titleSheet.

Used By

schematicDesign

sheetList (SCH)

sheetList ::= '(' 'sheetList'
{ sheetRef }
')'

Description

sheetList is a list of sheets in a schematic design.

Used By

schematicPrintSettings

sheetNameDef (SCH)

sheetNameDef ::= nameDef

Description

sheetNameDef provides the sheet name of a schematic sheet. This reflects the contents of the **Sheet Name** edit box in the Options Current Sheet dialog.

Used By

sheet

sheetNum (SCH)

sheetNum ::= '(' 'sheetNum'
integerToken
')'

Description

sheetNum provides the sheet number of a schematic sheet. This reflects the sheet number in the **Sheets** list box in the Options Current Sheet dialog.

Used By

sheet

sheetOrderNum (SCH)

sheetOrderNum ::= '(' 'sheetOrderNum'
integerToken
')'

Description

sheetOrderNum.

Used By

sheet

sheetRef (SCH)

sheetRef ::= '(' 'sheetRef'
integerToken
')'

Description

sheetRef provides a reference to a sheet in a schematic sheet. The referenced value is the number of the sheet.

Used By

sheetList

sheetSize (SCH)

sheetSize ::= ('size_A' | 'size_B' | 'size_C' | 'size_D' | 'size_E' | 'size_A0' | 'size_A1' | 'size_A2' |
'size_A3' | 'size_A4')

Description

sheetSize represents the size of a schematic sheet. Valid sheet sizes are imperial A through E, or metric A4 through A0.

Used By

scaling

solderFlowDirection (PCB)

solderFlowDirection ::= '(' 'solderFlowDirection'
('solderFlowTopToBottom' | 'solderFlowLeftToRight' | 'solderFlowRightToLeft' | 'solderFlowBottomToTop')
')'

Description

solderFlowDirection specifies the direction of wave solder flow during the manufacturing process. When pattern graphics are automatically swapped, the proper orientation is determined relative to solder flow direction.

Used By

pcbDesignHeader

solderSwell (PCB)

solderSwell ::= '(' 'solderSwell'
dbNumber
')'

Description

solderSwell specifies the global solder mask swell for a PCB design. This value reflects the value in the **Solder Mask Swell** edit box in the Options Configure dialog.

Used By

pcbDesignHeader

sourceLibrary

sourceLibrary ::= '(' 'sourceLibrary'
stringToken
')'

Description

sourceLibrary indicates the library from which a component was placed. It is currently included in schematic designs, but not in PCB designs; the PCB file reader will ignore it if it is present.

Used By

compHeader

spokeWidth (PCB)

spokeWidth ::= '(' 'spokeWidth'
dbNumber
')'

Description

spokeWidth specifies the width of thermal spokes.

Used By

padShape, viaShape

startAngle

startAngle ::= '(' 'startAngle'
numberToken
')'

Description

startAngle specifies the angle at which an arc begins. Angles begin at 0 degrees on the positive X axis and increase in a counterclockwise fashion.

Used By
arc

startPoint

startPoint ::= location

Description
startPoint is a starting point locatoin.

Used By
triplePointArc

startRange (PCB)

startRange::= integerToken

Description
startRange represents the layer a pad/via's hole range begins on. This is specified in the Modify Hole Range dialog..

Default
LAYER_TOP_SIGNAL

Used By
padStyleDef, viaStyleDef

stringToken

stringToken ::= QUOTEDIDENTIFIER

Description
stringToken is a string of characters surrounded by double quotation marks.

Notes
Within a *stringToken,* certain characters are represented by special character sequences since they are normally cannot be printed in a meaningful fashion or they conflict with the characters used to delimit the string. These sequences look like two characters, but represent only one. They include:

\f	formfeed
\n	newline
\r	carriage return
\t	horizontal tab
\"	double quote
\\	backslash

Used By
many productions

strokeWidth

strokeWidth ::= '(' 'strokeWidth'
dbNumber
')'

Description
strokeWidth indicates the width of a stroke in a font. This value reflects the value in the **Thickness** edit box in the Modify Text Style dialog.

Used By
font

style

style ::= '(' 'style'
('SolidLine' | 'DottedLine' | 'DashedLine')
')'

Description
style indicates the style of a line: it can be solid, dotted, or dashed. This value reflects the status of the **Style** radio buttons in the Modify Line dialog in Schematic.

Used By
line

styleDef

styleDef ::= (padStyleDef | viaStyleDef | textStyleDef)

Description
A *styleDef* is either a *padStyleDef,* a *viaStyleDef,* or a *textStyleDef.* This name reflects the contents of the **Pad Name** edit box in the Add Pad Style dialog, the **Via Name** edit box in the Add Via Style dialog, or the **Text Name** edit box in the Add Text Style dialog.

Used By
library

styleNameDef

styleNameDef ::= nameDef

Description
styleNameDef indicates the name of a pad style, via style, or text style.

Used By
padStyleDef, textStyleDef, viaStyleDef

subtitle (PCB)

subtitle ::= '(' 'subtitle'
stringToken
')'

Description
subtitle represents the subtitle of a detail or diagram object.

Used By
detail, diagram

sweepAngle

sweepAngle ::= '(' 'sweepAngle'
numberToken
')'

Description
sweepAngle specifies the angle through which an arc extends. The ending angle of an arc is the *startAngle* plus the *sweepAngle.*

Used By
arc

symbol (SCH)

symbol ::= '(' 'symbol'
symbolRef
refDesRef
partNum
[altType]
location
[rotation]
[isFlipped]
{ attr }
')'

Description
symbol represents a Schematic symbol object created from a previously-defined *symbolDef* template. The symbol is created for a particular RefDes and part number, and references a specific symbol template.

Notes
symbol attributes are handled as follows: if the position, textstyle, or visibility flag of the RefDes or Value attribute is different from the *symbolDef,* it is written in the *symbol* with an empty value string; the value string is written in *compInst.* All user symbol attributes are written in symbol.

Defaults
altType: Normal
rotation: 0 degrees
isFlipped: False

Used By
schDrawObj

See Also
compInst, symbolDef

symbolAlias (SCH)

symbolAlias ::= '(' 'symbolAlias'
symbolNameDef
symbolNameRef
')'

Description
symbolAlias defines a name alias *symbolNameDef* for the symbol name *symbolNameDef.*

Notes
There may be more than one *symbolAlias* for a given P-CAD symbol name; additional *symbolAliases* will be created for each symbol name alias.

Used By
library

symbolDef (SCH)

symbolDef ::= '(' 'symbolDef'
symbolNameDef
originalName
{ schDrawObj }
')'

Description
symbolDef defines a template used to create symbols. It contains a list of *schDrawObjs* which represent the primitive objects that make up the symbol.

Notes

There may be more than one *symbolDef* for a given P-CAD symbol name; additional *symbolDefs* will be created if two symbols of the same name are not equivalent. The *symbolNameDef* uniquely identifies each template, and *originalName* specifies the actual name of the symbol.

Symbols must have RefDes and Type attributes, and cannot contain wires, busses, or other symbols.

symbolDef attributes are handled as follows: RefDes and Type attributes are always written; however, they are written to indicate their graphical information and location only, and their value string is always empty. Their textstyle and visibility are written if they are different from the defaults. Value attributes are written if they exist, and the value string is written empty. All visible attributes are written here as well; non-visible attributes are written in the *symbol.*

Used By

library

See Also

compInst, symbol

symbolName

symbolName ::= '(' 'symbolName'
stringToken
')'

Description

symbolName indicates the name of the symbol attached to a component for a particular part number and alternate combination. This symbol name reflects the symbol name listed in a particular cell in the Edit Attach Symbols dialog in P-CAD Library Manager.

Used By

attachedSymbol

symbolNameDef (SCH)

symbolNameDef ::= nameDef

Description

symbolNameDef defines the name of a symbol template.

Used By

symbolDef

symbolNameRef (SCH)

symbolNameRef ::= nameRef

Description

symbolNameRef is used to refer to a previously-defined *symbolNameDef.*

Used By

symbolRef

symbolRef (SCH)

symbolRef ::= '(' 'symbolRef'
symbolNameRef
')'

Description

symbolRef provides a reference to a symbol previously defined in a library using *symbolDef.*

Used By
symbol

symPinNum

symPinNum ::= '(' 'symPinNum'
integerToken
')'

Description
symPinNum specifies the symbol pin number for a component pin.

Used By
compPin

table

table ::= '(' 'table'
tableType
location
[width]
[extent]
[rotation]
[isFlipped]
title
textStyleRef
[tableInfo]
[numMajorColumns]
[sheetOnlyNets]
')'

Description
table represents a table object in an P-CAD design.

Notes
The *table* production is used to represent all of the P-CAD table types; the actual type of the table is designated by the *tableType* production. Other productions are included through the *tableInfo* production to maintain table-specific information, such as user-defined columns in a drill table. These productions do not store the data maintained in each cell, but rather just enough data so that they can be regenerated when read.

numMajorColumns and *sheetOnlyNets* options apply only to tables of type *netIndexTable*.

Tables can be created and viewed only by P-CAD Document Toolbox; they are displayed as placeholders in P-CAD without the Document Toolbox option.

Defaults
width: 10.0 mils
rotation: 0 degrees
isFlipped: False
numMajorColumns: 1
sheetOnlyNets: False

Used By
pcbDrawObj, schDrawObj

tableInfo

tableInfo ::= (drillTableInfo | powerTableInfo | noteTableInfo)

Description
tableInfo is used to add table-specific information to a *table* production.

Used By
table

tableType

tableType ::= ('noteTable' | 'revisionNoteTable' | 'drillTable' | 'spareGateTable' | 'powerTable' | 'netIndexTable')

Description
tableType designates the type of a table object.

Used By
table

testPoint (PCB)

testPoint::= '(' 'testPoint'
pt
testPointSide
[testPointSnapToCenter]
[testPointAssociation]
[isFixed]
')'

Description
testPoint contains the data that describes a testPoint object, including its location, association, etc.

Notes
No testPointAssociation is used when a testPoint is a member of a component pattern. It is used to associate *free* test points to vias and pattern pads.

Defaults
snapToCenter: False
testPointAssociation: -1
isFixed: False
isFlipped: False
isVisible: True

Used By
pcbDrawObj

testPointAssociation

testPointAssociation::= '(' 'testPointAssociation'
[testPointID | node]
')'

Description
testPointAssociation is used to associate test points with vias and pattern pads. If *testPointID* is used, the test point can be associated to a via with an identical *testPointID* designation. If *node* is used, the test point is associated to a component pattern pad.

Notes
No testPointAssociation is used when a testPoint is a member of a component pattern. It is used to associate *free* test points to vias and pattern pads.

Used By
testPoint

testPointID (PCB)

testPointID::= '(' 'testPointID'
integerToken
')'

Description
testPointID is used to designate a unique identifier for a *testPoint* object. This identifier is used to maintain associative test point information.

Used By
testPointAssociation

testPointSide (PCB)

testPointSide ::= '(' 'testPointSide'
('TestPointSideTop' | 'TestPointSideBottom')
')'

Description
testPointSide specifies the side of the board from which the test point is accessed.

Used By
testPoint

testPointSize

testPointSize ::= '(' 'testPointSize'
dbNumber
')'

Description
testPointSize describes the size of test point.

Used By
pcbDesignHeader, schematicDesignHeader

testPointSizePrint

testPointSizePrint ::= '(' 'testPointSizePrint'
dbNumber
')'

Description
testPointSizePrint describes the print size of test point.

Used By
pcbDesignHeader, schematicDesignHeader

testPointSnapToCenter (PCB)

testPointSnapToCenter ::= '(' 'testPointSnapToCenter'
booleanToken
')'

Description
testPointSnapToCenter specifies that a test point will snap to the center of pads/vias on which it placed/moved.

Used By
testPoint

text

text ::= '(' 'text'
location
stringToken
textStyleRef
[rotation]
[isFlipped]
[justify]
[extent]
')'

Description
text represents a text object. It references a text style that must have been previously defined in a *library* section.

Notes
A single *text* object can contain up to 2000 characters. *extent*, which indicates the bounding box of the text, is always written by P-CAD PCB and P-CAD Schematic, but is ignored during File Open.

Defaults
rotation: 0 degrees
isFlipped: False
justify: Lower-left

Used By
bus, pcbDrawObj, schDrawObj, wireName, xRef

textStyleAllowTType

textStyleAllowTType ::= '(' 'textStyleAllowTType'
booleanToken
')'

Description
textStyleAllowTType indicates whether a text style has dual stroke/TrueType font behavior, allowing text objects that use the style to be displayed in either their stroke or TrueType forms. *textStyleAllowTType* reflects the **Allow TrueType** check box in the Text Style Properties dialog, and affects whether the display of text objects that use the style can be set to TrueType mode by the **Display TrueType** push button in the Options Text Style dialog

Used By
textStyleDef

textStyleDef

textStyleDef ::= '(' 'textStyleDef'
styleNameDef
font
[font]
[textStyleAllowTType]
[textStyleDisplayTType]
')'

Description
textStyleDef defines a text style. The *textStyleDef* reflects the text style information in the Modify Text Style dialog.

Each text object that is created references a defined text style using *textStyleRef*.

Notes
If two fonts are provided, they normally should be of different *fontType*'s (Stroke/TrueType). If they are of the same type, then the second font defintion takes precedence.

Defaults

If a single TrueType font is provided ---
textStyleAllowTType: True
textStyleDisplayTType: True

Otherwise ---
textStyleAllowTType: False
textStyleDisplayTType: False

Used By

styleDef

textStyleDisplayTType

textStyleDisplayTType ::= '(' 'textStyleDisplayTType'
booleanToken
')'

Description

textStyleDisplayTType indicates whether a text style that has a TrueType font mode is displayed in its TrueType font. *textStyleDisplayTType* reflects the **Display** radio button setting in the Text Style Properties dialog.

Used By

textStyleDef

textStyleRef

textStyleRef ::= '(' 'textStyleRef'
nameRef
)'

Description

textStyleRef provides a reference to a text style previously defined in a *library* using *textStyleDef*.

Used By

attr, field, table, text, zones

thermal (PCB)

thermal ::= '(' 'thermal'
padEnd
pourEnd
[thermalWidth]
')'

Description

thermal represents a thermal in a copper pour. Coordinates are specified for the end of the thermal touching the pad and the end of the thermal touching the copper pour.

Defaults

thermalWidth: thermalWidth specified in copperPour95.

Used By

copperPour

thermalSpokes (PCB)

thermalSpokes ::= '(' 'thermalSpokes'
numberToken
')'

Description

thermalSpokes is unsupported and fixed at a value of 4.

Used By

copperPour95

thermalType (PCB)

thermalType ::= '(' 'thermalType'
('NoTherm' | 'Therm45' | 'Therm90')
')'

Description

thermalType specifies the type of a thermal. *Therm45* and *Therm90* represent 45-degree and 90-degree thermals respectively, while *NoTherm* represents the absence of a thermal.

Used By

copperPour

thermalWidth (PCB)

thermalWidth ::= '(' 'thermalWidth'
dbNumber
')'

Description

thermalWidth specifies the width of a thermal.

Used By

copperPour

tieNetValue (PCB)

tieNetValue ::= '(' 'tieNetValue'
stringToken
')'

Description

tieNetValue specifies a unique name for associating nets that are tied together using a polygon tie.

Used By

poly

time

time ::= hourNumber
minuteNumber
secondNumber

Description

time represents a time of day by listing hour, minute, and second values.

Used By

timeStamp

timeStamp

timeStamp ::= '(' 'timeStamp'
date
time

')'

Description

timeStamp lists a *date* and *time* combination. It is intended to designate when an P-CAD ASCII file was generated.

Used By

written

title

title ::= '(' 'title'
stringToken | text
')'

Description

title designates the title of an object. Only the title for a designView will use a text production, all others will a stringToken

Used By

column, detail, designView, diagram, table

titleSheet

titleSheet ::= '(' 'titleSheet'
[titleSheetNameDef]
[titleSheetScale]
[isVisible]
[offset]
[border]
[zones]
{ schDrawObj } | { pcbDrawObj }
)'

Description

titleSheet lists the objects that appear on the title sheet of a design. The *isVisible* flag corresponds to the **Display Title Sheet** check box in the Options Configure dialog.

Notes

Only lines, arcs, polygons, text, attributes, fields, tables, and metafiles can appear in a *titleSheet*. The *offset*, *border*, and *zones* productions are generated only by P-CAD Document Toolbox. In Schematic, the *isVisible* option of the *schematicDesign*'s *titleSheet* controls visibility of all *titleSheet*s in the design.

Defaults

isVisible: False

Used By

layerDef, schematicDesign, sheet

titleSheetNameDef (SCH)

titleSheetNameDef ::= nameDef

Description

titleSheetNameDef denotes the title sheet file name.

Used By

titleSheet

titleSheetScale (SCH)

titleSheetScale ::= numberToken

Description
titleSheetScale denotes the title sheet scale factor.

Notes
TitleSheetScale currently has no effect (it is reserved for future use).

Used By
titleSheet

toolAssn (PCB)

toolAssn ::= '(' 'toolAssn'
holeDiam
[isHolePlated]
toolRef
')'

Description
drillSym associates holes in a PCB design with a drill symbol shape or a single alphabetic character label. A hole is identified by diameter and plating characteristic. These symbols are used for Printed, DXF, and Gerber Output, and reflect the assignments in the Drill Symbol Assignments dialog.

Description
toolAssn assigns holes in a PCB design to a particular tool. A hole is identified by diameter and plating characteristic. This assignment is the same as the assignment shown in the Tool Assignments dialog.

Defaults
isHolePlated: True

Used By
ncDrillSettings

toolCode (PCB)

toolCode ::= '(' 'toolCode'
integerToken
')'

Description
toolCode specifies a tool code for an N/C Drill tool.

Used By
toolDef, toolRef

toolDef (PCB)

toolDef ::= '(' 'toolDef'
toolCode
toolDiam
')'

Description
toolDef maps an N/C Drill tool code to a tool of a particular diameter. This mapping reflects the mapping defined in the Describe/Assign Tools dialog.

Used By
ncDrillSettings

toolDiam (PCB)

toolDiam ::= '(' 'toolDiam'

dbNumber
')'

Description
toolDiam defines the diameter of an N/C Drill tool.

Used By
toolDef

toolRef (PCB)

toolRef ::= '(' 'toolRef'
toolCode
')'

Description
toolRef is used to refer to a previously-defined N/C Drill tool, by its tool code.

Used By
toolAssn

triplePointArc

arc ::= '(' 'triplePointArc'
centerPoint
startPoint
endPoint
width
[netNameRef]
{ dimensionRef }
[isFixed]
')'

Description
triplePointArc represents a circular arc object for instances where the starting and terminal points of the arc are critical. The arc is defined by its center point, beginning at its start point, and sweeping to its terminal end point in a counter-clockwise direction.

Defaults
isFixed: False

Used By
dimensionGraphic, pcbDrawObj, schDrawObj

units (PCB)

units ::= '(' 'units'
('in' | 'mm')
')'

Description
units designates the units to be used in Gerber and N/C Drill output files, and in drill tables. For CAM, this setting reflects the status of the **Output Units** radio buttons in the Gerber Format and N/C Drill Format dialogs; for drill tables, this setting reflects the status of the **Units** radio button in the Place Table dialog.

Notes
Drill tables can only be placed in P-CAD Document Toolbox.

Used By
drillTableInfo, gerberSettings, ncDrillSettings

useApertureHoles (PCB)

useApertureHoles ::= '(' 'useApertureHoles'
booleanToken
')'

Description
useApertureHoles indicates whether P-CAD PCB should create Gerber apertures with holes during automatic aperture assignment. It reflects the status of the **Pad/Via Holes** check box in the Aperture Assignments dialog.

Used By
gerberSettings

useArcs (PCB)

useArcs ::= '(' 'useArcs'
booleanToken
')'

Description
This parameter is reserved for future use and is not currently used.

Used By
gerberSettings

useDesignRules (PCB)

useDesignRules ::= '(' 'useDesignRules'
booleanToken
')'

Description
This parameter is specifies whether *copperPour95* should use the system clearance rules or a fixed amount to calculate backoff distance.

Used By
copperPour95

useGlobalSwell (PCB)

useGlobalSwell ::= '(' 'useGlobalSwell'
booleanToken
')'

Description
useGlobalSwell specifies whether pad and via styles should use global or local swell values. If this value is True, the style will use the global swell value; otherwise, it will use the local swell value.

Used By
padStyleDef, viaStyleDef

variant

variant ::= '(' 'variant'
variantName
[variantDescription]
{varintExcludedComponentName}
{varintComponentName}
')'

Description

Variant represents a variant object.

Used By

netlist, schematicDesign

variantComponent

variantComponent ::= '(' 'variantComponent'
variantComponentName
{attr}
')'

Description

variantComponent specifies included component which attributes has been modified for a variant. It is optional property of variant.

Used By

variant

variantComponentName

variantComponentName ::= '(' 'variantComponentName'
stringToken
')'

Description

variantComponentName specifies the component name which attribute has been modified for a variant.

Used By

variantComponent

variantDescription

variantDescription ::= '(' 'variantDescription'
stringToken
')'

Description

variantDescription specifies the description of a variant. It is optional property of variant.

Used By

variant

variantName

variantName ::= '(' 'variantName'
stringToken
')'

Description

variantName specifies the name of a variant. It could signify the name of a variant for a variant definition, or a reference to an existing variant, depending on its context.

Used By

outputItem

variantExcludedComponentName

variantExcludedComponentName ::= '(' 'variantExcludedComponentName'
stringToken
')'

Description
variantExcludedComponentName specifies the excluded component name for a variant.

Used By
variant

vertex (PCB)

vertex ::= '(' 'vertex'
depth
[first]
pt
')'

Description
vertex designates a vertex within a copper pour.

Defaults
first: False

Used By
copperPour

See Also
copperPour, depth

verticalZones

verticalZones ::= '(' verticalZones'
integerToken
numDirection
numType
')'

Description
Describes the vertical zoning information for a title sheet.

Used By
zones

via (PCB)

via ::= '(' 'via'
viaStyleRef
location
[rotation]
[isFlipped]
[netNameRef]
{ dimensionRef }
[isFixed]
[testPointAssociation]
')'

Description
via represents a via in a PCB design. It references a via style that must have been previously defined in a *library* section. If the via is part of a net, it will have a net name reference.

Defaults

rotation: 0 degrees
isFlipped: False
isFixed: False

Used By

PcbDrawObj

viaGridVisibility (PCB)

viaGridVisibility ::= '(' ' viaGridVisibility '
('Show' | ' Show_at_plowing ' | 'Hide')
')'

Description

viaGridVisibility indicates if plow via grid is visible. This value reflects the status of the **Plow Via Grid Visibility** group box in the Options Grid dialog.

Defaults

viaGridVisibility : Show_at_plowing

Used By

gridState

viaShape (PCB)

viaShape ::= '(' 'viaShape'
(layerNumRef | layerType)
viaShapeDfn
[spokeWidth]
[noCopperPourConnect]
')'

Description

viaShape defines the size and shape of a via on a particular layer or layer type. Each *viaShape* corresponds to a single **Via Definition** entry in the Modify Via Style dialog.

Defaults

spokeWidth: 0
noCopperPourConnect: False

Used By

viaStyleDef

viaShapeDfn (PCB)

viaShapeDfn ::= viaShapeType
(shapeWidth shapeHeight | [shapeSidesDfn rotation outsideDiam] shapeOutline)

Description

viaShapeDfn defines the shape and size of a via.

Notes

shapeOutline is required when *padShapeType* is *Polygon.*
[shapeSidesDfn rotation outsideDiam] are required for regular polygon shapes.

Used By

viaShape

viaShapeType (PCB)

viaShapeType ::= '(' 'viaShapeType'
padViaShapeType
')'

Description
viaShapeType defines the shape of a via.

Used By
viaShapeDfn

viaStyleDef (PCB)

viaStyleDef ::= '(' 'viaStyleDef'
styleNameDef
holeDiam
[isHolePlated]
[holeOffset]
[useGlobalSwell]
[localSwell]
[startRange]
[endRange]
{ viaShape }
')'

Description
viaStyleDef defines a via style. *viaStyleDef* reflects the via style information in the Modify Via Style dialog and the Modify Hole Range dialog.

Each via that is created references a defined via style using *viaStyleRef.* The via style is defined by specifying a list of *viaShapes,* each of which corresponds to a single **Via Definition** entry in the Modify Via Style dialog.

Defaults
isHolePlated: True
holeOffset: none
useGlobalSwell: True
localSwell: 0

Used By
styleDef

viaStyleRef (PCB)

viaStyleRef ::= '(' 'viaStyleRef'
nameRef
')'

Description
viaStyleRef provides a reference to a via style previously defined in a *library* using *viaStyleDef.*

Used By
via

viaThermalSpokes (PCB)

viaThermalSpokes ::= '(' viaTthermalSpokes'
numberToken
')'

Description
viaThermalSpokes is unsupported and fixed at a value of 4.

Used By
copperPour95

viaThermalType (PCB)

viaThermalType ::= '(' viaTthermalType'
('NoTherm' | 'Therm45' | 'Therm90')
')'

Description
viaThermalType specifies the type of a thermal specifically for vias. *Therm45* and *Therm90* represent 45-degree and 90-degree thermals respectively, while *NoTherm* represents the absence of a thermal.

Used By
copperPour95

viaThermalWidth (PCB)

viaThermalWidth ::= '(' viaThermalWidth'
dbNumber
')'

Description
viaThermalWidth specifies the width of a thermal specifically for vias.

Used By
copperPour95

viaToLineClearance (PCB)

viaToLineClearance ::= '(' 'viaToLineClearance'
dbNumber
')'

Description
viaToLineClearance defines the via-to-line clearance value for a particular layer. This value reflects the via-to-line clearance value for a layer in the Design Rule Check Clearances dialog.

Used By
layerDef

viaToPadClearance (PCB)

viaToPadClearance ::= '(' 'viaToPadClearance'
dbNumber
')'

Description
viaToPadClearance defines the via-to-pad clearance value for a particular layer. This value reflects the via-to-pad clearance value for a layer in the Design Rule Check Clearances dialog.

Used By
layerDef

viaToViaClearance (PCB)

viaToViaClearance ::= '(' 'viaToViaClearance'
dbNumber
')'

Description

viaToViaClearance defines the via-to-via clearance value for a particular layer. This value reflects the via-to-via clearance value for a layer in the Design Rule Check Clearances dialog.

Used By

layerDef

viewLog (PCB)

viewLog ::= '(' 'viewLog'
booleanToken
')'

Description

viewLog specifies whether the log file should be automatically displayed after generation of Gerber or N/C Drill files. This setting reflects the status of the **View log upon completion** check box in the Setup Output Files dialogs.

Used By

gerberSettings, ncDrillSettings

width

width ::= '(' 'width'
dbNumber
')'

Description

width specifies the width of an item.

Used By

apertureDef, arc, border, copperPour, line, plane, table

wire (SCH)

wire ::= '(' 'wire'
line
[dispName]
[wireName]
')'

Description

wire represents a wire in a schematic design. The *line* designates the coordinates of the wire, which is extended for bus entries. The *netname* in the *line* specifies the netname of the wire.

Defaults

dispName: False
wireName: The default text style and wire name location are used.

Used By

schDrawObj

wireName (SCH)

wireName ::= text

Description

wireName specifies the location of the wirename text associated with a wire, using text. The quoted string is ignored (the name of the wire is always equal to the name of the net to which the wire belongs), as are the text style (which must always be "(WireStyle)") and layer number.

Used By

wire

workspaceSize

workspaceSize ::= '(' 'workspaceSize'
xPoint
yPoint
')'

Description
workspaceSize specifies the workspace size of a PCB or Schematic design.

Used By
pcbDesignHeader, schematicDesignHeader

written

written ::= timeStamp
{ fileAuthor | copyright | program }

Description
written provides general information about an P-CAD ASCII file, including the time the file was generated , the author of the file, the name of the program that created the file, and any copyright information for that program.

Used By
asciiHeader

xPoint

xPoint ::= dbNumber

Description
xPoint specifies an X-coordinate. The coordinate value may be followed by an optional unit designator; if no unit designator is specified, the units specified by the *fileUnits* keyword are used.

Used By
holeOffset, offset, pt, workspaceSize

xRef (SCH)

xRef ::= '(' 'xRef'
text
')'

Description
xRef describes a cross reference annotation on a Schematic sheet connector.

Notes
Sheet connector cross references are only displayed in P-CAD Document Toolbox, and only if the **Show Cross Referencing** checkbox in Options Display is checked.

Used By
schDrawObj

yearNumber

yearNumber ::= integerToken

Description
yearNumber denotes the number of the year in a date value.

Used By
date

yPoint

yPoint ::= dbNumber

Description
yPoint specifies a Y-coordinate. The coordinate value may be followed by an optional unit designator; if no unit designator is specified, the units specified by the *fileUnits* keyword are used.

Used By
holeOffset, offset, pt, workspaceSize

zeroFormat (PCB)

zeroFormat ::= '(' 'zeroFormat'
('leading' | 'trailing' | 'none')
')'

Description
zeroFormat specifies the zero suppression format for N/C Drill file generation. *leading* specifies that leading zeroes should be suppressed from the output file, *trailing* specifies that trailing zeroes should be suppressed, and *none* specifies that no zeros should be suppressed.

Used By
ncDrillSettings

zones

zones ::= '(' 'zones'
isVisible
textStyleRef
horizontalZones
verticalZones
')'

Description
Describes the zoning information specified for a title sheet. This information is accessible in P-CAD Document Toolbox.

Used By
titleSheet