PARAMETRIC TEXT FORMAT (PTF) A

Parametric Text Format (PTF) is DENEBACAD's plain text language interface. PTF lets you save the contents of DENEBACAD documents in files that are compatible with external programs such as spreadsheets, word processors, and programming language text editors.

You also can create PTF files for import into DENEBACAD using such programs. The import feature lets you import objects as well as execute scripted commands.

Introduction

What is PTF?

PTF [P*arametric* T*ext* F*ormat*] ***adj.*** **1** a tabbed text file format that expresses the content of a DenebaCAD™ document *[This is a PTF file]* **2** a public interchange file format, native to DenebaCAD, that can be used to translate other CAD file formats to the DenebaCAD file format *[This is a PTF exchange file]* ***—n.*** A language with a syntax that lets you create DenebaCAD files from textual or numerical data *[This object is described in PTF]*

PTF files can be created in several ways:

Quick and easy You can use DENEBACAD's Save As command in the File menu to create a PTF file from a DENEBACAD document. The PTF file describes in plain text the objects in the DENEBACAD document, and can later be imported, with or without modifications, into another DENEBACAD document.

> **Pros:** Fast and easy. Ensures file transportability even in tough environments such as the Internet.
>
> **Cons:** Resulting file is flat. Modifications are difficult because objects are hard to locate within the file.

Slow and dirty You can use a text editor to write, line by line, the commands required to create a collection of objects.

> **Pros:** A text editor is probably the most common tool in any computer environment. Ensures file transportability even in tough environments such as the Internet.
>
> **Cons:** Resulting file is flat. Modifications are difficult because objects are hard to locate within the file. Error-prone. Files are hard to debug. This method, while viable, is not recommended.

Smart and savvy You can use a spreadsheet application to create every command required to reproduce a given collection of objects.

Pros: Besides text editors, a spreadsheet application is probably the most common tool in any computer environment. You can use your spreadsheet knowledge to create functions and macros to automate your object design process, effectively improving your professional productivity.

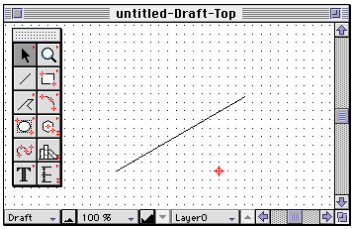
— Ensures file transportability even in tough environments such as the Internet.

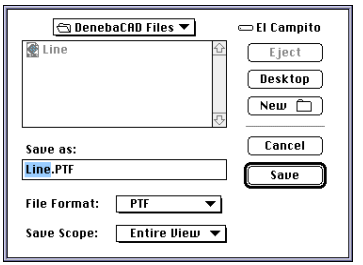
— Files are easy to debug because you have a better control at the source.

— Builds your self esteem and ego.

Cons: Knowledge of spreadsheet operations is required.

Creating and using PTF files

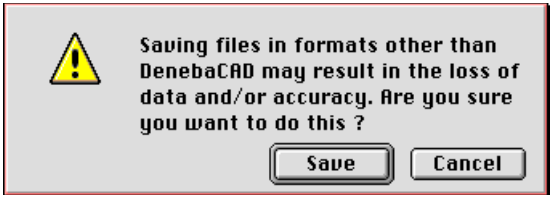


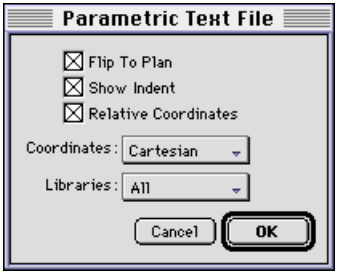


You create PTF files by using the Save As command in the File menu. This allows you to save an object, a group of objects, or a complete document using the PTF file format.

To create a simple PTF file in DENEBACAD

1. Draw a line in a new DENEBACAD document.
2. Choose Save As in the File menu.
3. Type a name for the file in the Save As text box.
4. In the File Format pop-up menu, select PTF. DENEBACAD adds ".PTF" to the file's name.
5. Choose a location to save the file, and then click Save.
6. A message appears, warning that saving documents in other formats can result in lost data. Click Save in the message box to continue.





7. The Parametric Text File dialog box appears.
8. Click OK to use the default settings and save the PTF file.

To view and modify the PTF file

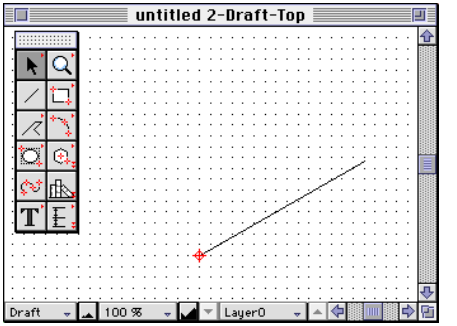
1. Launch your favorite spreadsheet program.
2. Choose Open in the File menu. In the dialog box that appears, select the PTF file that you saved and click Open. The information in the PTF file appears in the spreadsheet.

	A	B	C	D	E
1	Begin PTF				
2	Distance Units	DecInches			
3	Angular Units	DecDegrees			
4	Coordinates	Cartesian			
5	VectorCoordSys	XY	dXdY		
6	View	Top			
7	Mode	Draft			
8	Layer #	0			
9	Pen Color (R G B)	0	0	0	
10	Line Type (Thick Dash Eccen)	1	1	0	
11	Arrow Heads (Begin End)	0	0		
12	Fill Color (R G B)	0	0	0	
13	Fill Pattern	0			
14	Object	Line			
15		Name	Line 1		
16		# Points	2		
17		Pt# X Y	1	-108	48
18		Pt# dX dY	2	156	-72
19	End PTF				

3. Change the value of the X coordinate (cell D17 in the example) to the value you want. In this case, we'll change the value from -108 to 0. In your spreadsheet, you can use formulas to define this value.
4. Choose Save As in the File menu and save a copy of the file as a tab-delimited text file.

To insert the edited PTF file into DENEBACAD

1. Launch DENEBACAD.
2. Choose Import in the File menu. The PTF Place Options dialog box appears. Click OK to implement the default settings and import the file. Notice that the line you've drawn appears in a new position on screen.

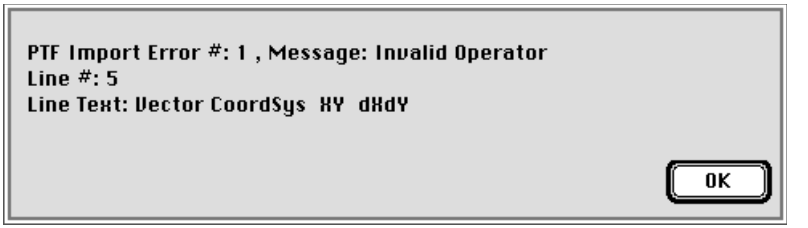


When you import the PTF file, the changes you made in the spreadsheet program appear in the DenebaCAD document.

In addition to changing position data, you can change color, dash settings, line thickness, and any other parameter in the description of an object by editing a PTF file.

Error messages If you change something in the PTF file so that DENEBACAD can't interpret the file correctly, an error message appears. If this happens, note that the message lists the line in the file where the error occurs. Following the label "Line Text," the message box displays the text of the line containing the error.

Click OK in the message box to continue. Then, check the line in the PTF file that was listed in the error message.



Options for saving PTF files

You have several options you can choose when you save files in the PTF file format:

Flip to Plan Uses the Cartesian coordinate system in Top view. Positive coordinates are up and right, negative coordinates are down and left. If this option is not selected, DENEBACAD uses the current view.

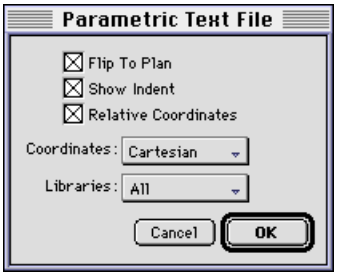
Show Indent Saves the text tabs in the file to show the hierarchy of objects. If this option is not selected commands and object descriptions will not be indented.

Relative Coordinates Choose this option to define points by their relationship to the first point in the object or command. If you clear this check box, all coordinates will be saved as absolute and refer to a common fixed origin point

Coordinates pop-up menu Choose the coordinate system you want DENEBACAD to use when saving the PTF file.

Libraries pop-up menu Choose which Library Objects and Symbols you want to include in the PTF file. You can include all Library Objects and Symbols listed in the Library palette, Library Objects and Symbols used in the document only, or no Objects and Symbols.

Options for saving PTF files appear in the Parametric Text File dialog box



PTF Place Options

When importing a PTF format file, you have two options:

Use Top Coordinates System Uses the Cartesian coordinate system in Top view. Positive coordinates are up and right, negative coordinates are down and left.

Show Only Placed Layers Displays objects in the drawing that belong to imported layers only.

Options for importing PTF files appear in the PTF Place Options dialog box

Creating PTF files with a text editor

Just fire up your text editor and type away! Keep the following issues in mind:

PTF is not case-sensitive "Fill Pattern," "FILL PATTERN" and "fILL pATTERN" all represent the same information.

The field separator is the Tab character Always use the Tab key to separate commands from their parameters, and the parameters from each other.

Space to the left is irrelevant The following two samples are the same, even though the space to the left is different.

Although space to the left is not relevant, it helps organize your information. Notice that the second sample is easier to read and follow than the first one.

Begin PTF			
Distance Units	DecInches		
Angular Units	DecDegrees		
View	Top		
Mode	Draft		
Layer #	0		
Pen Color (R G B)	89	0	0
Line Type (Thick Dash Eccen)	1	1	0
Arrow Heads (Begin, End)	0	0	
Fill Color (R G B)	0	0	0
Fill Pattern	0		
Object	DiagonalRect		
Name	DiagonalRect 1		
# Points	2		
Pt# X Y	1	-228.000	144.000
Pt# DX DY	2	192.000	-96.000
End PTF			

```
Begin PTF
        Distance Units                          DecInches
        Angular Units                           DecDegrees
        View                                    Top
        Mode                                    Draft
        Layer #                                 0
        Pen Color (R G B)                       89          0           0
        Line Type (Thick Dash Eccen)            1           1           0
        Arrow Heads (Begin, End)                0           0
        Fill Color (R G B)                      0           0           0
        Fill Pattern                            0
        Object                                  DiagonalRect
                Name                            DiagonalRect 1
                # Points                        2
                Pt# X Y                         1           -228.000    144.000
                Pt# DX DY                       2           192.000     -96.000
End PTF
```

Space above "Begin PTF" is irrelevant To annotate or describe your work, you can use as many lines as you want above the "Begin PTF" command.

Space below "End PTF" is irrelevant To annotate or describe your work, you can use as many lines as you want below the "End PTF" command.

Empty lines are irrelevant You can insert blank lines as needed. In general, this makes PTF files easier to read and follow.

Creating PTF files in a spreadsheet program

Using a spreadsheet program is by far the smartest and easiest way to generate PTF text files.

Because the final file structure is the same, the same issues apply when you use a spreadsheet to create a PTF file as when you create a PTF file with a text editor.

You can use all of the available spreadsheet functions to generate values and strings which can then be used to create and modify DENEBACAD objects.

When you finish creating or modifying a PTF file, remember to save the file in tab-delimited text format.

PTF Reference

This section describes the commands and object descriptions that are implemented in the PTF file format.

The PTF file structure

When you save a file in PTF format, it appears in spreadsheet applications as shown under “Sample PTF file structure,” next. The exact appearance depends on the spreadsheet you use.

The PTF information is arranged in lines, with keywords and parameters separated from each other by tab characters.

There are three general areas in a PTF file: a Header, a Body, and a Footer.

The PTF Header The PTF Header contains commands that identify the beginning of the PTF file. It sets values that will affect the entire DENEBACAD document. The PTF Header can be set one time only.

The PTF Body The PTF Body contains all of the other commands that relate to object creation and manipulation, and to the DENEBACAD work environment settings.

The PTF Footer The PTF Footer contains a single line that identifies the end of the PTF file:

```
End PTF
```

Sample PTF file structure

Begin PTF			
Distance Units	DecInches		
Angular Units	DecDegrees		
View	Top		
Mode	Draft		
Layer #	0		
Pen Color (R G B)	89	0	0
Line Type (Thick Dash Eccen)	1	1	0
Arrow Heads (Begin, End)	0	0	
Fill Color (R G B)	0	0	0
Fill Pattern	0		
Object	DiagonalRect		
Name	DiagonalRect 1		
# Points	2		
Pt# X Y	1	-228.000	144.000
Pt# DX DY	2	192.000	-96.000
Pen Color (R G B)	0	40	100
Object	DiagonalRect		
Name	DiagonalRect 2		
# Points	2		
Pt# X Y	1	-84.000	84.000
Pt# DX DY	2	180.000	-84.000
Pen Color (R G B)	0	0	0
Object	DiagonalRect		
Name	DiagonalRect 3		
# Points	2		
Pt# X Y	1	60.000	12.000
Pt# DX DY	2	204.000	-84.000
End PTF			

Typographic conventions

Several typographic conventions are used in the PTF Reference section to distinguish commands, parameters, and related information.

Commands and keywords These words must be typed as shown (although they are not case-sensitive). Commands and keywords tell DENEBACAD to create objects, use specific settings, and apply commands.

Commands and keywords are shown in `monospace` type:

```
Line Type (Thick Dash Eccen)
```

Parameters Parameters are the values or data used by commands. In some cases, parameters are optional. In other cases, as noted in the text, a missing parameter is assumed to be zero.

Most parameters shown in this section are merely "metasymbols" in the PTF descriptions. In other words, the parameters shown are placeholders to be replaced by actual data or values.

Parameter metasymbols are shown in *italic* type:

Num_Points_Integer

Data Data used in this section represent actual numbers or other values that you specify. For example, when coordinates appear in an object description, they represent actual coordinates that you specify in a PTF object description. Examples of actual values that can be used are shown in roman type.

General syntax for object descriptions

The following syntax is used for object descriptions in PTF format. Keywords are separated from parameters, and parameters are separated from each other, by a tab character.

Keywords	Parameters		
`Object`	*Tool_Name*		
`Name`	*Default_Name* | *User_Name*		
`# Points`	*Num_Points_Integer*		
`Pt# X Y`	*Ordinal_Integer*	*X_Real*	*Y_Real*
`Pt# DX DY`	*Ordinal_Integer*	*dX_Real*	*dY_Real*
`Pt# D A`	*Ordinal_Integer*	*Distance_Real*	*Angle_Real*
`Pt# DD DA`	*Ordinal_Integer*	*dDistance_Real*	*dAngle_Real*

Parameter definitions

The parameters used in object descriptions and other commands are represented by the following metasymbols (metasymbols are terms that will be replaced by actual parameter values in a PTF file):

Tool_Name	The name of the tool used to generate the object
Default_Name	The object name assigned by DENEBACAD
User_Name	An object name assigned by the user
Num_Points_Integer	The number of creation points in the object
Ordinal_Integer	An integer indicating the position of the point within the object definition list
X_Real, Y_Real	Coordinate real values (absolute coordinates)
Distance_Real	Distance real value
Angle_Real	Angle real value
dDistance_Real	Change in distance real value (relative to previous point)
dAngle_Real	Change in angle real value (relative to previous point)
n	An integer value
a b c ...	Coordinate values

Notes

When creating PTF documents using text editors or spreadsheet applications you can use the *User_Name* parameter when naming objects. The *Default_Name* parameters are used by DENEBACAD when you save a DENEBACAD document in PTF format.

In PTF keywords and parameters, point coordinates can be expressed as absolute coordinates (*X Y*), relative coordinates (*dX dY*), absolute distance and angle (*D A*)or relative distance and angle (*dD dA*).

Note: The letter “D” or “d” is used in place of the delta symbol to indicate a relative distance or angle. For example, *dX_Real* is a relative X coordinate value.

In a PTF file, an object's definition ends when the indicated number of points (*Num_Points_Integer*) is read, or when another object description is initiated, or when a general or particular attribute command is encountered.

If the number of points read is different than the number of points specified by the *Num_Points_Integer* parameter, the object is not drawn and an error message is generated.

Simple Object descriptions

Simple objects are objects you would create in DENEBACAD using a single tool, without additional object operations.

Simple Object descriptions lets you create these same objects in PTF files. A Simple Object description starts with the `Object` keyword, followed by keywords for the object's name, number of points, and the point coordinates.

Simple Object descriptions

3PointsArc

Describes an arc drawn by the Arc 3 Points tool, defined by three points: two endpoints and a point on the arc's periphery.

`Object`	3PointsArc		
`Name`	3PointsArc *n*		
`# Points`	3		
`Pt# X Y`	1	*a*	*b*
`Pt# DX DY`	2	*c*	*d*
`Pt# DX DY`	3	*e*	*f*

3PointsCircle

Describes a circle drawn by the Circle 3 Points tool, defined by three points on its periphery.

`Object`	3PointsCircle		
`Name`	3PointsCircle *n*		
`# Points`	3		
`Pt# X Y`	1	*a*	*b*
`Pt# DX DY`	2	*c*	*d*
`Pt# DX DY`	3	*e*	*f*

Simple Object descriptions

CenterEllipse

Describes an ellipse or circle drawn by the Ellipse Center to Corner tool, defined by its center point and one corner of its bounding rectangle.

`Object`	CenterEllipse		
`Name`	CenterEllipse *n*		
`# Points`	2		
`Pt# X Y`	1	*a*	*b*
`Pt# DX DY`	2	*c*	*d*

CenterRadiusArc

Describes an arc drawn by the Arc Radius tool, defined by three points: its center, a starting point on the periphery and an ending point on its periphery

`Object`	CenterRadiusArc		
`Name`	CenterEllipse *n*		
`# Points`	3		
`Pt# X Y`	1	*a*	*b*
`Pt# DX DY`	2	*c*	*d*
`Pt# DX DY`	2	*e*	*f*

CenterRadiusCircle

Describes an ellipse or circle drawn by the Circle Radius tool, defined by two points: its center and a point on its periphery.

`Object`	CenterRadiusArc		
`Name`	CenterEllipse *n*		
`# Points`	2		
`Pt# X Y`	1	*a*	*b*
`Pt# DX DY`	2	*c*	*d*

CenterRect

Describes a rectangle drawn by the Rectangle Center to Corner tool, defined by two points: the center point and one corner point.

`Object`	CenterRect		
`Name`	CenterRect *n*		
`# Points`	2		
`Pt# X Y`	1	*a*	*b*
`Pt# DX DY`	2	*c*	*d*

Simple Object descriptions

CenterRoundRect

Describes a rounded rectangle drawn by the Rounded Rectangle tool, defined by two points: its center and one of its corners.

`Object`	CenterRect		
`Name`	CenterRect *n*		
`# Points`	2		
`Pt# X Y`	1	*a*	*b*
`Pt# DX DY`	2	*c*	*d*

DiagonalEllipse

Describes an ellipse or circle drawn by the Ellipse Diagonal tool, defined by two opposite corners of its bounding rectangle.

`Object`	DiagonalEllipse		
`Name`	DiagonalEllipse *n*		
`# Points`	2		
`Pt# X Y`	1	*a*	*b*
`Pt# DX DY`	2	*c*	*d*

DiagonalRect

Describes a rectangle drawn by the Rectangle Diagonal tool, defined by two opposite corner points.

`Object`	DiagonalRect		
`Name`	DiagonalRect *n*		
`# Points`	2		
`Pt# X Y`	1	*a*	*b*
`Pt# DX DY`	2	*c*	*d*

DiagonalRoundRect

Describes a rounded rectangle drawn by the Rounded Rectangle Diagonal tool, defined by two opposite corner points.

`Object`	DiagonalRoundRect		
`Name`	DiagonalRoundRect *n*		
`# Points`	2		
`Pt# X Y`	1	*a*	*b*
`Pt# DX DY`	2	*c*	*d*

Simple Object descriptions

DirectLightSource

Describes a directional light source drawn by the Directional Light tool, defined by its position and a point located on its illumination axis. The Power setting defaults to 100. You can not define the Power (brightness) setting.

`Object`	DirectLightSource		
`Name`	DirectLightSource *n*		
`# Points`	2		
`Pt# X Y`	1	*a*	*b*
`Pt# DX DY`	2	*c*	*d*

FreeRect

Describes a rectangle drawn by the Rectangle 3 Points tool, defined by the end points of one side and an end point of the opposite side.

`Object`	FreeRect		
`Name`	FreeRect *n*		
`# Points`	3		
`Pt# X Y`	1	*a*	*b*
`Pt# DX DY`	2	*c*	*d*
`Pt# DX DY`	3	*e*	*f*

LightSource

Describes a light source drawn by the Omnidirectional Light tool, defined by its position. The Power setting defaults to 100. You can not define the Power (brightness) setting.

`Object`	LightSource		
`Name`	LightSource *n*		
`# Points`	1		
`Pt# X Y`	1	*a*	*b*

Line

Describes a line drawn by the Line tool, defined by two endpoints.

`Object`	Line		
`Name`	Line *n*		
`# Points`	2		
`Pt# X Y`	1	*a*	*b*
`Pt# DX DY`	2	*c*	*d*

Simple Object descriptions

PolygonVertex

Describes a polygon drawn by the Ploygon Vertex tool, defined by two endpoints.

`Object`	PolygonVertex		
`Name`	PolygonVertex *n*		
`#Sides`	6		
`# Points`	2		
`Pt# X Y`	1	*a*	*b*
`Pt# DX DY`	2	*c*	*d*

PolygonMidPoint

Describes a polygon drawn by the Polygon Midpoint tool, with six sides.

`Object`	PolygonMidpoint		
`Name`	PolygonMidpoint *n*		
`#Sides`	6		
`# Points`	2		
`Pt# X Y`	1	*a*	*b*
`Pt# DX DY`	2	*c*	*d*

QB_Spline

Describes a curve drawn by the Curve tool, defined by 5 points. The first and last points are the endpoints and the other points define the curvature of the curve.

`Object`	QB_Spline		
`Name`	QB_Spline *n*		
`# Points`	*n*		
`Pt# X Y`	1	*a*	*b*
`Pt# DX DY`	2	*c*	*d*
`Pt# DX DY`	*n*	*e*	*f*

OffsetPoly

Describes an offset polyline drawn by the Double Polyline tool, defined by “n” points. The Poly Width parameter defines the distance between the parallel lines. The Offset parameter defines the distance from the center of the parallel lines to the line described by the set points. The Poly Ends parameter must be set to Close Close.

`Object`	OffsetPoly				
`Name`	OffsetPoly *n*				
`Poly Ends`	Close Close				
`Poly Width / Offset`	4.00	0.00			
`# Points`	5				
`Pt# X Y`	1	*a*	*b*		
`Pt# DX DY`	2	*c*	*d*		
`Pt# DX DY`	*n*	*e*	*f*		
`#Lines Eccen List`	*n*	*a*	*b*	*c*	*d*

Simple Object descriptions

Polyline

Describes a polyline drawn by the Single Polyline tool,defined by its "n" endpoints.

Object	Polyline		
Name	Line *n*		
# Points	n		
Pt# X Y	1	*a*	*b*
Pt# DX DY	2	*c*	*d*
Pt# DX DY	*n*	*e*	*f*

3DElement

Describes a three-dimensional object defined by "n" points. DENEBACAD must be set to Sculpt mode before you define any 3D object.

Object	3DElement			
Name	3DElement *n*			
# Points	4			
Pt# X Y Z	1	*a*	*b*	*c*
Pt# DX DY DZ	2	*d*	*e*	*f*
Pt# DX DY DZ	3	*g*	*h*	*i*
Pt# DX DY DZ	*n*	*j*	*k*	*l*
#Face #Pts List	*n*	*n1*	*n2*	*n3*

2DLibrary

Describes a 2D Library Object.

Object	Group		
Name	Object Name		
ObjectLibrary			
HiddenBoolean			
Insertion Point	*X*	*Y*	
# Objects	2		
Pen Color (R G B)	*r*	*g*	*b*
Line Type (Thick Dash Eccen)	1	1	0
Arrow Heads (Begin End)	0	0	
Fill Color (R G B)	0	0	0
Fill Pattern	0		
Object	DiagonalRect		
Name	DiagonalRect *n*		
# Points	2		
Pt# X Y	1	*a*	*b*
Pt# DX DY	2	*c*	*d*
Object	DiagonalEllipse		
Name	DiagonalEllipse *n*		
# Points	2		
Pt# X Y	1	*a*	*b*
Pt# DX DY	2	*c*	*d*

Simple Object descriptions

3DLibrary

Describes a 3D Library Object.

`Object`	3DElement			
`# Points`	*n*			
`Pt# X Y Z`	1	*a*	*b*	*c*
`Pt# DX DY DZ`	2	*d*	*e*	*f*
`Pt# DX DY DZ`	3	*d*	*e*	*f*
`Pt# DX DY DZ`	*n*	*d*	*e*	*f*
`#Face #Pts List`	1	*1*	*3*	*2*
`#Face #Pts List`	2	*1*	*4*	*3*
`#Face #Pts List`	3	*1*	*2*	*3*
`#Face #Pts List`	4	*2*	*1*	*4*
`#Face #Pts List`	5	*2*	*3*	*4*
`#Face #Pts List`	*n*	*3*	*4*	*2*

Boolean Object descriptions

Boolean Object descriptions specify composite objects derived from the application of a Boolean operation to several objects.

Boolean object descriptions

Boolean

`Object`	Boolean
`Name`	*Boolean_Element_Name*
`Boolean Mode`	*Boolean_Mode_String**
`# Objects`	*n*
`Pen Color (R G B)`	*R G B*
`Line Type (Thick Dash Eccen)`	*T D E*
`Arrow Heads (Begin End)`	*B E*
`Fill Color (R G B)`	*R G B*
`Fill Pattern`	*FP*
`Object`	*Object_Definition_1*
`Object`	*Object_Definition_2*
`Object`	*Object_Definition_n*

* *Boolean_Mode_String* can be replaced with one of five values: None, Add, Substract, Intersect, or Hide

Grouped Object descriptions

Grouped Object descriptions specify composite objects derived from the grouping of simple objects. A grouped object can be one of the following types:

- Simple group
- Section group
- Library Object group
- Library Symbol group

Library Objects and Library Symbols contained in PTF files can be loaded into standard DENEBACAD documents and used in exactly the same way as Library Objects and Library Symbols that are contained in standard DENEBACAD documents.

PTF Libraries

When you add a Library Object or Library Symbol to a DENEBACAD document, the directory dialog box includes a pop-up menu you can use to select the type of library host file to use.

The directory dialog box also includes a “Show All Files” checkbox.

Grouped Object descriptions

Group

`Object`	Group
`Name`	*Group_Name*
`# Objects`	*n*
`Object`	*Object_Definition_1*
`Object`	*Object_Definition_2*
`Object`	*Object_Definition_n*

Grouped Object descriptions

Group: Library

```
Object                          Group
Name                            Library 1
Object                          Library
Insertion Point                 -4.20 2.70
# Objects                       2
Pen Color (R G B)               0       0       0
Line Type (Thick Dash Eccen)    1       1       0
Arrow Heads (Begin End)         0       0
Fill Color (R G B)              0       0       0
Fill Pattern                    0
Object                          DiagonalRect
Name                            DiagonalRect 1
# Points                       2
Pt# X Y                         1       -5.40   4.00
Pt# DX DY                       2       1.80    -2.00
Object                          DiagonalEllipse
Name                            DiagonalEllipse 1
# Points                       2
Pt# X Y                         1       -4.40   2.60
Pt# DX DY                       2       1.40    -1.20
```

Group: Symbol

```
Object                          Group
Name                            Symbol 1
SymbolLibrary
Insertion Point                 3.10    2.30
# Objects                       3
Object                          DiagonalEllipse
Name                            DiagonalEllipse 2
# Points                       2
Pt# X Y                         1       2.20    3.20
Pt# DX DY                       2       1.80    -1.80
Object                          Line
Name                            Line 1
# Points                       2
Pt# X Y                         1       3.10    4.10
Pt# DX DY                       2       0.00    -3.60
Object                          Line
Name                            Line 1
# Points                       2
Pt# X Y                         1       1.30    2.30
Pt# DX DY                       2       3.60    0.00
```

Grouped Object descriptions

Group: Section

Object	Group
Name	Section 3
Section	
Insertion Point	-4.30 -2.50
# Objects	3
Object	DiagonalEllipse
Name	DiagonalEllipse 2
# Points	2
Pt# X Y	1 -5.20 -1.60
Pt# DX DY	2 1.80 -1.80
Object	Line
Name	Line 1
# Points	2
Pt# X Y	1 -4.30 -0.70
Pt# DX DY	2 0.00 -3.60
Object	Line
Name	Line 1
# Points	2
Pt# X Y	1 -6.10 -2.50
Pt# DX DY	2 3.60 0.00

Container descriptions

A Container is a non-printing object used to group otherwise unrelated objects for reporting purposes. Any object can be used as a Container. In the example below, a DiagonalRect is used as a Container.

Container

Object	DiagonalRect
Name	*Object_Name_String*
Container	
# Points	2
Pt# X Y	1 *X* *Y*
Pt# DX DY	2 *Dx* *Dy*

Dimension Object descriptions

Dimension Object descriptions specify dimension objects that can be drawn with the Dimension tools in DENEBACAD.

Dimension object descriptions

ChainDimm Describes a dimension drawn by the Chain Dimension tool.

```
Object                          ChainDimm
Name                            ChainDimm 1
Dimm Arrowhead                          8
Witness (Position Length)               1      2
Dimm Text (Font Size Face Just)         3      9      0      1
Dimm Distances (A B C D E)            1.6    3.2    1.6    1.6    1.6
Dimm Precision (Dec Places +Tol -Tol)   2  0.000  0.000
# Points                                4
Pt# X Y                                 1  -7.80   3.20
Pt# DX DY                               2   2.40   0.00
Pt# DX DY                               3   2.00   0.00
Pt# DX DY                               4   2.20   0.00
Dimm Line                           -7.80   4.20   6.60   0.00
Dimm Label# Dh Dv Text                  1   0.83   0.16   2.40
Dimm Label# Dh Dv Text                  2   0.63   0.16   2.00
Dimm Label# Dh Dv Text                  3   0.73   0.16   2.20
```

BaseLineDimm Describes a dimension drawn by the Basline Dimension tool.

```
Object                          BaseLineDimm
Name                            BaseLineDimm 1
Dimm Arrowhead                          8
Witness (Position Length)               1      2
Dimm Text (Font Size Face Just)         3      9      0      1
Dimm Distances (A B C D E)            1.6    3.2    1.6    1.6    1.6
Dimm Precision (Dec Places +Tol -Tol)   2  0.000  0.000
# Points                                4
Pt# X Y                                 1  -7.80   0.00
Pt# DX DY                               2   2.40   0.00
Pt# DX DY                               3   2.00   0.00
Pt# DX DY                               4   2.20   0.00
Dimm Line                           -7.80   0.60   6.60   0.00
Dimm Label# Dh Dv Text                  1   0.83   0.16   2.40
Dimm Label# Dh Dv Text                  2   1.83   0.16   4.40
Dimm Label# Dh Dv Text                  3   2.93   0.16   6.60
```

Dimension object descriptions

LeaderDimm

Describes a dimension drawn by the Leader Dimension tool.

```
Object                          LeaderDimm
Name                        LeaderDimm 1
Dimm Arrowhead                        8
Dimm Text (Font Size Face Just)       3      9      0      1
Dimm Distances (A B C D E)          1.6    0.0    0.0    0.0    0.0
Dimm Precision (Dec Places +Tol -Tol) 2  0.000  0.000
# Points                              2
Pt# X Y                               1  -7.80  -1.20
Pt# DX DY                             2   6.60   0.00
Dimm Label# Dh Dv Text                1   2.93   0.16   6.60
```

DoubleLeader Dimm

Describes a dimension drawn by the Constrained Dimension tool.

```
Object                    DoubleLeaderDimm
Name                  DoubleLeaderDimm 1
Dimm Arrowhead                        8
Dimm Text (Font Size Face Just)       3      9             0      1
Dimm Distances (A B C D E)          1.6    0.0    0.0    0.0    0.0
Dimm Precision (Dec Places +Tol -Tol) 2  0.000  0.000
# Points                              2
Pt# X Y                               1  -7.80  -2.20
Pt# DX DY                             2   6.60   0.00
Dimm Label# Dh Dv Text                1   2.93   0.16   6.60
```

AngleDimm

Describes a dimension drawn by the Angle Dimension tool. The angle measured is between two lines. The first two points define the first line. The second set of points define the second line.

```
Object                          AngleDimm
Name                        AngleDimm 2
# Points                            4
Pt# X Y                             1     1.80    0.00
Pt# DX DY                           2     4.80    0.00
Pt# DX DY                           3    -2.00    2.80
Pt# DX DY                           4     1.63   -0.96
```

PTF commands

The PTF commands described in this section are presented using the following scheme:

CommandName

Command description

Location	Where in the PTF file the command is allowed.
Switches	Command modifiers, if applicable.
Parameters	None (if the command requires no parameters), or *ParameterName_Type*
Values	Strings or data that can be entered as parameters (shown separated by commas).
Syntax	One or more examples of the command syntax in the following format: `CommandName` *Param_1 Param_2 Param_n*
Note	When applicable, any special behavior or additional information needed to correctly use the command.
Warning	When applicable, any special information that should be taken into account or that could lead to problems or unexpected results.

One-time commands

These commands should appear only once in any PTF file. If a One-time command is repeated, DENEBACAD exectutes only the first instance of the command in the PTF file.

Begin PTF

Indicates the beginning of the PTF file information.

Location	PTF Header
Parameters	None
Syntax	`Begin PTF`
Note	Any information physically above it in the file is ignored

End PTF

Indicates the end of the PTF file information.

Location	PTF Footer
Parameters	None
Syntax	`End PTF`
Note	Any information physically below it in the file is ignored

Distance Units

Specifies linear units used to dimension objects.

Location	PTF Header
Parameters	*Distance_Unit_String*
Values	DecInches, Meters
Syntax	`Distance Units` *Distance_Unit_String*
Note	All distance units will be inserted as signed decimals. U.S. System units are always decimal inches.

Angular Units

Specifies angular units used to measure angles. DecDegrees is the only value allowed for the *Angular_Unit_String* parameter.

Location	PTF Header
Parameters	*Angular_Unit_String*
Values	DecDegrees
Syntax	`Angular Units` *Angular_Unit_String*
Note	All Angular units will be inserted as signed decimal degrees.

Environmental commands

The Environmental commands can be inserted anywhere within the Body of the PTF file (see "The PTF Body" on page 473 for more information). An Environmental command affects the operation of all commands and tools initiated after its appearance.

Mode

Directs the following operations to either the Draft or Sculpt modes in DenebaCAD's drawing environments. See the User Guide for more information on working in DENEBACAD modes. Draft is the default mode.

Parameters	*Mode_String*
Values	Draft, Sculpt
Syntax	`Mode` *Mode_String*

Notes — If omitted, PTF assumes that all operations will be executed within the current active drawing mode.

In Sculpt mode all 2D object descriptions are extruded using the currently active extrusion method.

In Draft mode DENEBACAD displays an error message if you define a 3D object.

View

Changes the environment to the view indicated in the parameter. Top is the default view.

Parameters — *View_String*

Values — Top, Bottom, Front, Back, Left, Right

Syntax — `View` *View_String*

Notes — If omitted, PTF assumes that all operations will be executed within the current active view.

General Attributes commands

The General Attributes commands can be inserted anywhere within the body of the PTF file (see "The PTF Body" on page 473 for more information). The requested setting affects the operation of all commands and tools initiated after its appearance.

Arrow Heads (Begin End)

Establishes the beginning and ending arrowheads to be applied to arrow lines.

Parameters — *Begin_Integer End_Integer*

Values — Positive non-zero integers

Syntax — `Arrow Heads` (Begin End) *Begin_Integer End_Integer*

Notes — *Begin_Integer* and *End_Integer* point to the relative position of the defined begin and end arrowheads in the arrowhead palette; 1 points to the first arrowhead at the top of the palette, 2 points to the second arrowhead in the palette, and so on.

VectorCoordSys

Establishes the coordinate definition standard for the DenebaCAD Drawing Vector. Vectors you define in other commands must be in the VectorCoordSys format you specified last.

Parameters	*Start_Point_String End_Point_String*
Values	*Start_Point_String* *XY:* Cartesian absolute coordinates *DA:* Polar absolute coordinates *End_Point_String* *XY:* Cartesian absolute coordinates *DA:* Polar absolute coordinates *dXdY:* Cartesian relative coordinates *dDdA:* Polar relative coordinates
Syntax	`VectorCoordSys` *Start_Point_String End_Point_String*
Notes	The command affects all coordinate input that occurs after its appearance.
Warning	Double-check input coordinate values. PTF will interpret an intended 25', 45 input as 25', 45' if the *Start_Point_String* is set to 'XY' and as 25' at 45º if set to 'DA'.

Layer

Specifies the current layer, which is the layer that will contain the next object created.

Parameters	*Layer_Number_Integer*
Values	Positive non-zero integer
Syntax	`Layer #` *Layer_Number_Integer*
Notes	If *Layer_Number_Integer* indicates a non-existent layer, a layer with that number is created.

Pen Color (R G B)

Defines a pen color based on RGB (Red, Green, and Blue) color values.

Parameters	*Red_Integer, Green_Integer, Blue_Integer*
Values	Positive integers from 0 to 100
Syntax	`Pen Color` (R G B) *Red_Integer Green_Integer Blue_Integer*
Notes	Any missing value is interpreted as zero

Line Type (Thick Dash Eccen)

Specifies the line type, using a integer values indexed to the tabs in the Pen palette.

Parameters	*Thick_Integer, Dash_Integer, Eccentricity_Integer*
Values	Positive non-zero integers
Syntax	`Line Type` (Thick Dash Eccen) *Thick_Integer Dash_Integer Eccentricity_Integer*
Notes	*Thick_Integer* is indexed to the Pen weight tab; *Dash_Integer* is indexed to the Pen style tab; *Eccentricity_Integer* is indexed to the eccentricity options on the Pen weight tab. If a value exceeds the largest index in any list, an error is produced and the last active value is reinstated.

Fill Color (R G B)

Defines a fill color using RGB (Red, Green, Blue) color values.

Parameters	*Red_Integer, Green_Integer, Blue_Integer*
Values	Positive integers from 0 to 100
Syntax	`Fill Color` (R G B) *Red_Integer Green_Integer Blue_Integer*
Notes	A missing value is interpreted as zero

Fill Pattern

Specifies a Fill pattern, using values indexed to the Fill Pattern tab.

Parameters	*Fill_Integer*
Values	Positive Integer
Syntax	`Fill Pattern` *Fill_Integer*
Notes	A missing value is interpreted as zero.

Fill Hatch

Specifies a Fill Hatch, using values indexed to the Fill Hatch tab.

Parameters	*Fill_Integer*
Values	Positive Integer
Syntax	`Fill Hatch` *Fill_Integer*
Notes	A missing value is interpreted as zero.

Fill Material

Specifies a Surface Material, using values indexed to the Surface Material tab.

Parameters	*Fill_Integer*
Values	Positive Integer
Syntax	`Fill Material` *Fill_Integer*
Notes	A missing value is interpreted as zero.

Object Selection commands

These commands are the same as the commands in the Edit menu of DENEBACAD. Careful use of selection commands in PTF is important. The selection status of objects affects almost every other PTF command.

If you define a new object, all selected objects become deselected, and DENEBACAD selects the new object. There is no PTF command to deselect objects.

Select

Selects objects to later operate on them.

Switches	Last (selects the most recent objects created; the number of objects selected is user defined)
	All (selects all objects created up to the point where this command appears in the file)
	ByName (selects specifically named objects
Parameters	*Number_of_Objects_integer*
Values	Positive non-zero integers
Syntax	`Select` Last *Number_of_Objects_integer*
	`Select` All
	`Select` ByName *Name_String*

AddToSelection

Selects objects and adds them to a previous selection.

Switches	All (selects all objects created up to the point where this command appears in the file)
	ByName (selects specifically named objects)
Parameters	*Name_String*
Syntax	`AddToSelection` All

Object Position commands

The Object Position commands are the same as the commands in the Position submenu in the DENEBACAD Object menu.

Note: A PTF file created from the DENEBACAD environment does not contain these commands, but contains the final description of objects created or modified by the commands.

Mirror

Flips the selected object(s) symmetrically with reference to a user-defined mirror line.

Parameters	*MirrorLine_Vector*
Values	Refer to the VectorCoordSys command
Syntax	`Mirror` *MirrorLine_Vector*

MirrorACopy

Flips a copy of the selected object(s) symmetrically with reference to a user-defined mirror line.

Parameters	*MirrorLine_Vector*
Values	Refer to the VectorCoordSys command
Syntax	`MirrorACopy` *MirrorLine_Vector*

Move

Moves the selected object(s) to a new position defined by a user-defined vector.

Parameters	*Move_Vector*
Values	Refer to the VectorCoordSys command
Syntax	`Move` *Move_Vector*

MoveACopy

Moves a copy of the selected object(s) to a new position defined by a user-defined vector.

Parameters	*Move_Vector*
Values	Refer to the VectorCoordSys command
Syntax	`MoveACopy` *Move_Vector*

InclusiveLArray

Takes the selected object(s) and distributes a user-defined number of copies along a user-defined vector. An off-plane distance may also be specified if Mode is set to Sculpt.

Parameters *NCopies_Integer*
Direction_Vector
OffPLane_Real (optional; zero if ommited)

Values *NCopies_Integer:* Positive non-zero integer
Direction_Vector: Refer to the VectorCoord Sys command
OffPLane_Real: Signed real number

Syntax `InclusiveLArray` *NCopies_Integer Direction_Vector* [*OffPLane_Real*]

OffsetLArray

Takes the selected object(s) and places a user-defined number of copies along a user-defined vector separated by a distance taken from the length of that vector. An off-plane distance may also be specified if Mode is set to Sculpt.

Parameters *NCopies_Integer*
Direction_Vector
OffPLane_Real (optional; zero if omitted)

Values *NCopies_Integer:* Positive non-zero integer
Direction_Vector: Refer to the VectorCoordSys command
OffPLane_Real: Signed real number

Syntax `OffsetLArray` *NCopies_Integer Direction_Vector OffPLane_Real*

Rotate

Rotates the selected object(s) a user-defined number of degrees around a user-defined centerpoint.

Parameters *Center_Point_A_Real* (first coordinate of the centerpoint)
Center_Point_B_Real (second coordinate of the centerpoint)
Angle_Angle

Syntax `Rotate` *Center_Point_A_Real Center_Point_B_Real Angle_Angle*

RotateACopy

Rotates a copy of the selected object(s) a user-defined number of degrees around a user-defined centerpoint.

Parameters *Center_Point_A_Real* (first coordinate of the centerpoint)
Center_Point_B_Real (second coordinate of the centerpoint)
Angle_Angle

Syntax `RotateACopy` *Center_Point_A_Real Center_Point_B_Real Angle_Angle*

PolarArray

Takes the selected object(s) and places a user-defined number of copies around a user-defined center point separated by an user-defined angle. An off-plane distance may also be specified if Mode is set to Sculpt.

Parameters *NCopies_Integer*
Direction_Vector
Center_Point
Angle_Angle
OffPLane_Real (optional)

Values *NCopies_Integer:* Positive Non-zero Integer
Direction_Vector: Refer to the VectorCoordSys command
Center_Point: Coordinate pair
OffPLane_Real: Signed real number

Syntax `OffsetLArray` *NCopies_Integer Direction_Vector Center_Point Angle_Angle OffPLane_Real*

Scale

Scales the selected object(s) by a user-defined horizontal and vertical proportion.

Parameters *Scale_Center:* Coordinate pair
HScale_Real: Positive real
VScale_Real: Positive real

Syntax `Scale` *Scale_Center HScale_Real VScale_Real*

Object Combine commands

The Object Combine commands in PTF perform Boolean operations on selected objects. The commands are the same as those in the Combine submenu in the DENEBACAD Object menu.

Please refer to the DENEBACAD documentation for more information about any command.

Note: A PTF file created from the DENEBACAD environment does not contain Combine commands, but contains the final description of the composite objects created or modified by the commands.

The general syntax for these commands is:

Command_Name

Where *Command_Name* is one of the following:

```
Unite
Intersect
PunchFromBack
PunchFromBack&Trim
PunchFromFront
PunchFromFront&Trim
```

Miscellaneous Object commands

The Miscellaneous Object commands in PTF perform operations on selected objects. The commands are the same as those found in the Object menu in DENEBACAD. Please refer to the DENEBACAD documentation for particulars about any command.

Note: A PTF file created from the DENEBACAD environment does not contain these commands, but contains the description of objects created or modified by the commands.

The general syntax for these commands is:

Command_Name

Where *Command_Name* is one of the following:

```
Chain
Unchain
Group
Ungroup
```

Object Extrusion commands

The PTF Object Extrusion commands are the same as those found in the Extrusion submenu in DENEBACAD. Please refer to the DENEBACAD documentation for particulars about these commands.

Note: A PTF file created from the DENEBACAD environment does not contain these commands, but contains the final descriptions of objects created or modified by the commands.

3DAxis

Defines a 3D Spin axis to be used for extrusions with the Spin command.

Parameters *View_String*
*Center_Point_*

Values *View_String:* Vertical | Lateral | Front
Coord1 Coord2

Syntax `3DAxis` *View_String Coord1 Coord2*

3DPlane

Defines a 3D plane set to be used for extrusions with the Linear command.

Parameters *View_String*
Distance1_Real
Distance2_Real

Values *View_String:* Vertical | Lateral | Front
Distance1_Real: Real
Distance2_Real: Real

Syntax `3DPlane` *View_String Distance1_Real*
Distance2_Real

Angled3DPlane

Defines an inclined 3D plane set to be used for further Linear Extrude commands.

Parameters *View_String*
Distance1_Real
Distance2_Real

Values *View_String:* Vertical | Lateral | Front
Distance1_Real: Real
Distance2_Real: Real

Syntax `Angled3DPlane` *View_String Vector1 Vector2*

Solid Faces

Specifies which faces of an extruded object are to be generated.

Parameters *NearCap_Boole*
FarCap_Boole
Sides_Boole

Values True | False

Syntax `Solid Faces` *NearCap_Boole FarCap_Boole Sides_Boole*

SpinOptions

Specifies the parameters to be applied by the next Spin command.

Parameters *Nsteps_Integer*
StepAngle_Angle
AxialStep_Real
SpiralStep_Real

Values *Nsteps_Integer:* Positive non-zero integer
StepAngle_Angle: Decimal Angle Value
AxialStep_Real: Real
SpiralStep_Real: Real

Syntax `SpinOptions` *Nsteps_Integer*
StepAngle_Angle AxialStep_Real
SprialStep_Real

SweepOptions

Specifies the parameters to be applied by the next Sweep command.

Parameters *DirectionString_String*
HScale_Real
VScale_Real

Values *DirectionString_String:* Vertical | Perpendicular
HScale_Real: Positive real
VScale_Real: Positive real

Syntax `SweepOptions` *DirectionString_String*
Hscale_Real Vscale_Real

SweepSection

Specifies the name of the section object to be used for the next Sweep command. Use this command before you execute the Extrude Sweep command.

Parameters *Section_String*

Syntax `SweepSection` *Section_String*

ExtrudeLinear

Extrudes the currently selected 2D object between the current 3D planes with the current Solid Faces options.

Parameters None

Syntax `ExtrudeLinear`

ExtrudeSpin

Spin extrudes the currently selected 2D object around the current 3D axis with the current Solid Faces options.

Parameters None

Syntax `Extrude Spin`

Extrude Sweep

Sweep extrudes the current Sweep Section along the currently selected 2D object with the current sweep options. DENEBACAD displays an error if you do not have a Sweep Section defined when you use this command.

Parameters None

Syntax `Extrude Sweep`