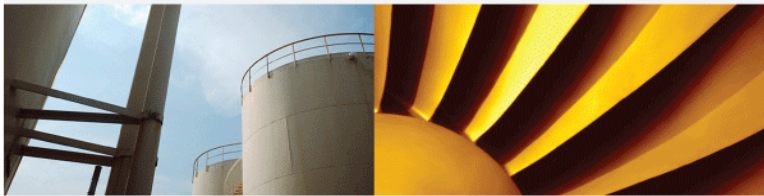


MEDUSA⁴

VERSION 5.0

Parametrics

USER GUIDE



All rights reserved. No part of this documentation may be reproduced in any manner (print, photocopy or other) without the written permission of CAD Schroer GmbH.

CAD Schroer GmbH has made its best effort to ensure that the information in this document is accurate and reliable, but cannot guarantee the accuracy, timeliness, reliability or completeness of any of the information contained herein. CAD Schroer GmbH will not make any warranty nor accept legal responsibility or liability of any kind for consequences resulting from errors or omissions.

Registered Trademarks of CAD Schroer GmbH:
MEDUSA, STHENO

Trademarks of CAD Schroer GmbH:
MEDUSA₄, STHENO/PRO, MEDEA, MPDS

Third-Party Products and Trademarks:
Pro/ENGINEER, Pro/DETAIL and Pro/TOOLKIT are registered trademarks of Parametric Technology Corporation.

All other brand or product names are trademarks or registered trademarks of their respective owners.

July 2010

Copyright © CAD Schroer GmbH
Fritz-Peters-Str. 26 - 30
D - 47447 Moers

Germany

CAD Schroer GmbH
Fritz-Peters-Str. 26-30
47447 Moers

Tel. +49 2841 91 84 - 0
Fax +49 2841 91 84 - 44
e-mail: info@cad-schroer.de
www.cad-schroer.de

France

CAD Schroer France SAS
17, Rue du Docteur Lebel
94300 Vincennes

Tel. +33 1 41 94 51 40
Fax +33 1 43 77 17 68
e-mail: info@cad-schroer.fr
www.cad-schroer.fr

Italy

CAD Schroer Italia S.R.L.
Via Piave, 1/E
20030 Senago (MI)

Tel.: +39 02-38303267
Fax: +39 02-33303399
e-mail: info@cad-schroer.it
www.cad-schroer.it

Switzerland

CAD Schroer AG
Bettlistr. 35
8600 Dübendorf

Tel. +41 44 802 89 - 80
Fax +41 44 802 89 - 88
e-mail: info@cad-schroer.ch
www.cad-schroer.ch

United Kingdom

CAD Schroer UK Ltd
39 Newnham Road
Cambridge
CB3 9EY

Tel. +44 1223 460 408
Fax +44 1223 460 409
e-mail: info@cad-schroer.co.uk
www.cad-schroer.co.uk

USA

CAD Schroer US, Inc.
34 Rand Place
Pittsford, NY 14534

Tel: +1 866-SCHROER (866-724-7637)
Fax: +1 866-724-1701
e-mail: info@cad-schroer.com
www.cad-schroer.com

TABLE OF CONTENTS

Preface	7
Introduction	9
What Is Parametric Design?	10
How Parametric Design Works	14
Tooltray	23
Dimensioning	25
Dimensions in Parametric Design	26
Tolerance Definition	29
Linear Dimensions	32
Circle and Arc Dimensions	34
Over-dimensioning and Under-dimensioning	36
Change Dimension Text	38
Getting Started	39
Overview	40
Creating a Viewbox	41
Creating a Reference Point	43
Dimensioning	45
Building the Parametric Grid	46
Examining the Parametric Grid	48
Entering Your Own Parameters	49
Parameterizing an Object	50
Example 1: Using a Prim	53
Example 2: Using Baselines	57

Simple Parameterization Examples	63
Example One	64
Example Two	68
Example Three	71
Geometric Constraints	75
What Are Geometric Constraints?	76
Looking at Inferred Constraints	77
Baselines	78
Specifying Constraints Explicitly	80
Example One	81
Example Two	83
Example Three	86
Variables and Expressions	91
Values in Dimensions	92
Variables	94
Assigning Values to Variables Using in-sheet commands	96
Assigning Values to LCIS Variables	98
Administration, Loading and Updating Variables	100
Errors While Using Variables	104
Expressions	105
Layers and Parametric Switches	107
How Layers Are Used	108
Changing Layer Properties	110
Parametric Graphics Control	111
Parametric Switches and Command Texts	112
Tables	117
Structure of a Table	118
Creating a Table	120
Where to Place a Table	123
Accessing Values From a Table	124
Worked Example	126
Parameterize Variant from a Table	131

Symbols	133
Creating Parametric Symbols	134
Loading Parametric Symbols	138
Worked Example 1	142
Loading Parametric Symbols With Tables	144
Loading Symbols Using CPI Named Groups	148
Worked Example 2 - CPI Named Groups	150
Parametric Groups	159
Introduction	160
Creating a Parametric Group	162
Example 1: Static Groups	163
Example 2: Dynamic Group with One Prim	165
Example 3: Dynamic Group with Two Prims	167
Example 4: Rotating Parametric Groups	171
Example 5: Dynamic Group with Three Prims	172
Post-Parameterization	175
Overview	176
Dialog	177
Creating Post-Parametric Definition Sheets	180
Callbacks	182
Example	186
Demo Sheet	188
Simulating Mechanisms	189
Introduction	190
Dialog	191
Example 1: Repeated Parameterization	193
Example 2: Simulating Linear Motion	195
Example 3: Simulating Rotary Motion	197
Example 4: Using a Program to Simulate Motion	200
Example 5: Simulating a Working Mechanism	202
Plotting	204

Other Applications	205
MEDUSA Applications	206
Parametric Design and 3D Modelling	207
Parametric Design and Sheet Metal Design	212
Appendix A Parametric Design Element Defaults	213
Basic Parametric Elements	214
Tables, Symbols, Groups and Error Messages	215
Graphical User Interface	216
Appendix B Error and Warning Messages	217
List of Figures	225
Index	229

PREFACE

Book Conventions

The following table illustrates and explains conventions used in writing about MEDUSA applications.

Convention	Example	Explanation
Menu	Choose <i>Zoom</i> from the View menu Add button Choose the tool <i>Creates thin solid lines.</i>	Indicates a command, function or button that you can choose from a menu, dialog or tooltray.
Syntax	<code>acos 0.345</code> The <code>ciaddobj</code> command Return or Control-g	User input, commands, keywords and keys to press on a keyboard.
SyntaxBold	Enter command> plot_config	Where system output and user input are mixed, user input is in bold.
<i>SyntaxItalic</i>	<code>tar -cvf /dev/rst0 filename</code>	Supply an appropriate substitute for each variable; for the given example replace <i>filename</i> with an actual file name.
<i>Filename&path</i>	<code>medusa\med2d\m2d\src\</code>	Shows path and filenames.
UPPERCASE	MEDUSA or CADCONVERT	Names of products.
<i>italic</i>	<i>left mouse button</i> <i>Drafting User Guide</i>	Indicates the buttons to press on a mouse and names of books.
bold	A temporary group is a collection of ...	Emphasize text.

Please note: Illustrations showing menus and forms are taken from a window system. The display for other platforms can differ slightly.

Online User Documentation (HTML)

Online documentation for each book is provided in HTML format. You can view this online documentation in the MEDUSA installation directory and directly by calling it up within the MEDUSA user interface.

Installation Directory

1. Navigate to the directory where MEDUSA is installed.
<MEDUSA installation directory>/meddoc/doc/<language>/ (Unix)
<MEDUSA installation directory>\meddoc\doc\<language>\ (Windows)
where <language> is either *english*, *german* or *french*.
2. Click on the file *mainmenu.htm*.
3. Click the book title you want to view.

MEDUSA Interface

1. Click left on the entry *Help* inside the main menu.
2. Choose MEDUSA Documentation from the pulldown menu.
A browser opens showing the *mainmenu.htm* listing all available documents.

Printing Version of the Documentation (PDF)

A PDF (Portable Document Format) file is included for each online book. You must have Acrobat Reader installed to view and print PDF files. If you don't have the Acrobat Reader, you can download it for free from the Adobe homepage:

<http://www.adobe.com/products/acrobat/readstep.html>

In order to search for a keyword across PDF files you can use **Acrobat Reader**. Therefore **Acrobat Reader Version 6.0** or higher has to be installed. The Reader provides a multiple search function, i.e. you can specify complete directories containing several PDF files for searching.

INTRODUCTION

This chapter introduces the MEDUSA Parametric Design system.

- [What Is Parametric Design?](#) 10
- [How Parametric Design Works](#) 14
- [Tooltray](#) 23

What Is Parametric Design?

MEDUSA Parametric Design is a system for scaling, or reconfiguring, the size and shape of part geometry according to a defined set of dimensions, or parameters. In MEDUSA 2D, you change the shape of an object using one of the transformation tools such as magnify or shrink, mirror, or rotate, or by redrawing it completely. Using Parametric Design, you change the shape of an object by changing its dimensions, pulling and stretching the geometry into another shape.

You can use Parametric Design to do any of the following:

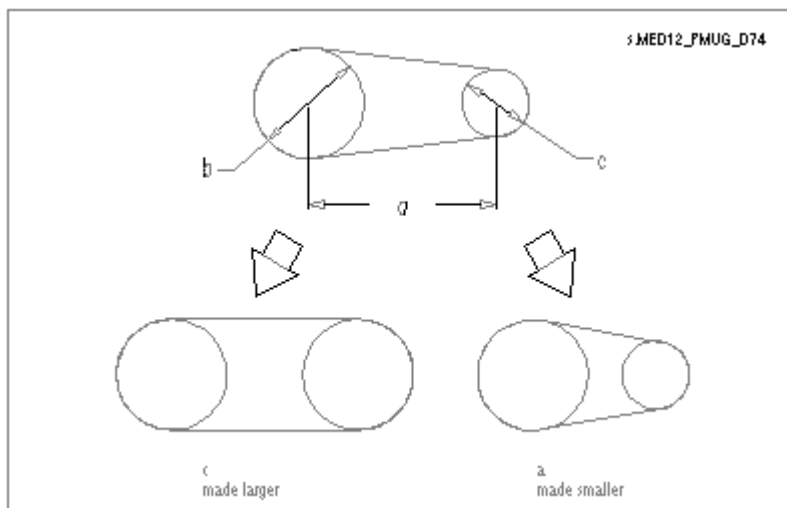
- Reconfigure the geometry of a part
- Create families of parts where one general case of a part is defined with a set of dimensions that you can vary
- Create libraries of symbols
- Simulate the action of mechanisms

Reconfiguring the Geometry of a Part

Parametric Design is used primarily to change one or more dimensions of an object on a sheet to make either fine adjustments to a design or radical changes to a drawing.

For example, the master drawing of the part shown in the figure below has three dimensions. Any of these can be changed to produce a new shape. If required, all three dimensions can be changed.

Figure 1 Changing the Dimensions of an Object

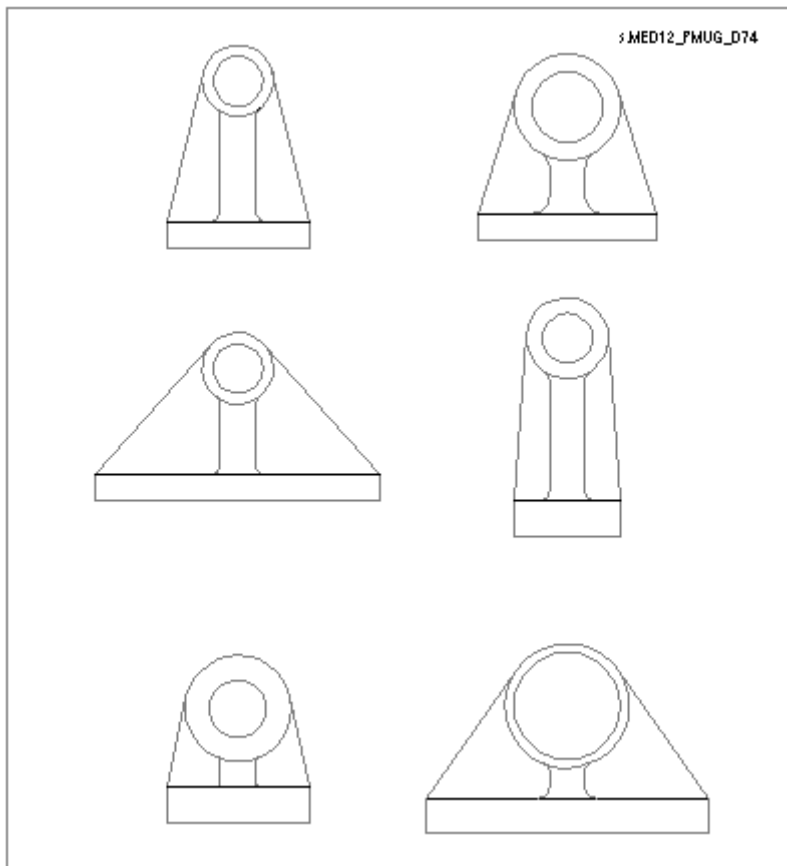


Creating Families of Parts

The Parametric Design system makes it easy to produce families of parts, where components which have similar functions but vary in shape and size. A good example is a set of spanners of various sizes.

Figure 2 shows another example, a set of brackets. Parametric Design allows you to produce a whole family of parts automatically from a single master drawing using a set of dimension variables.

Figure 2 A Family of Parts Made From a Single Master Drawing

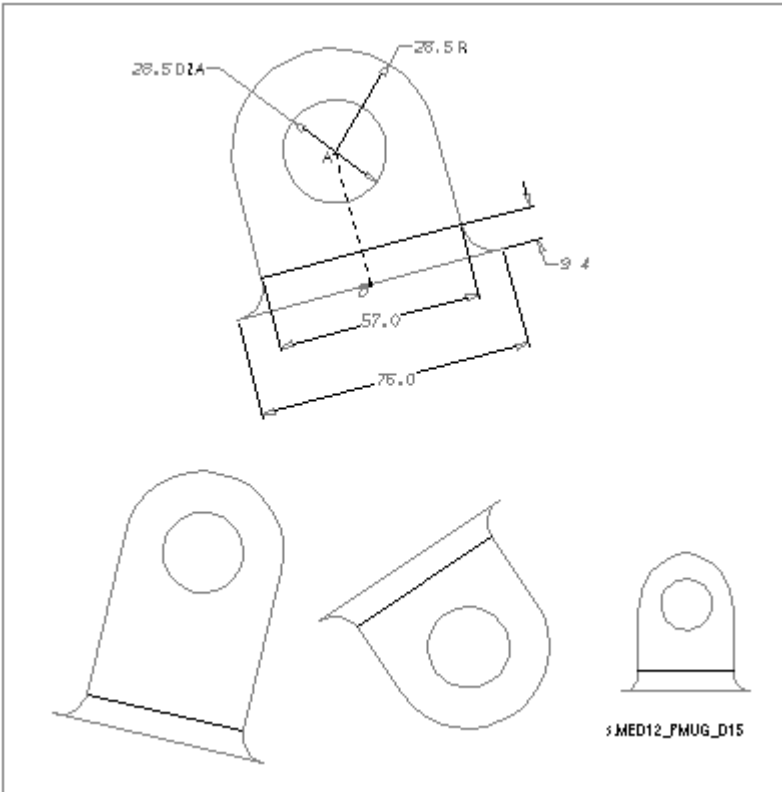


Creating Symbol Libraries

You often need to load standard symbols onto a MEDUSA sheet. Using Parametric Design, you can create libraries of parametric symbols to suit a range of different applications without having to redraw the symbol each time.

This example shows a master drawing of a tab and some of the results obtained by loading the symbol with different relative dimensions.

Figure 3 A Symbol Library Created Using Parametric Design

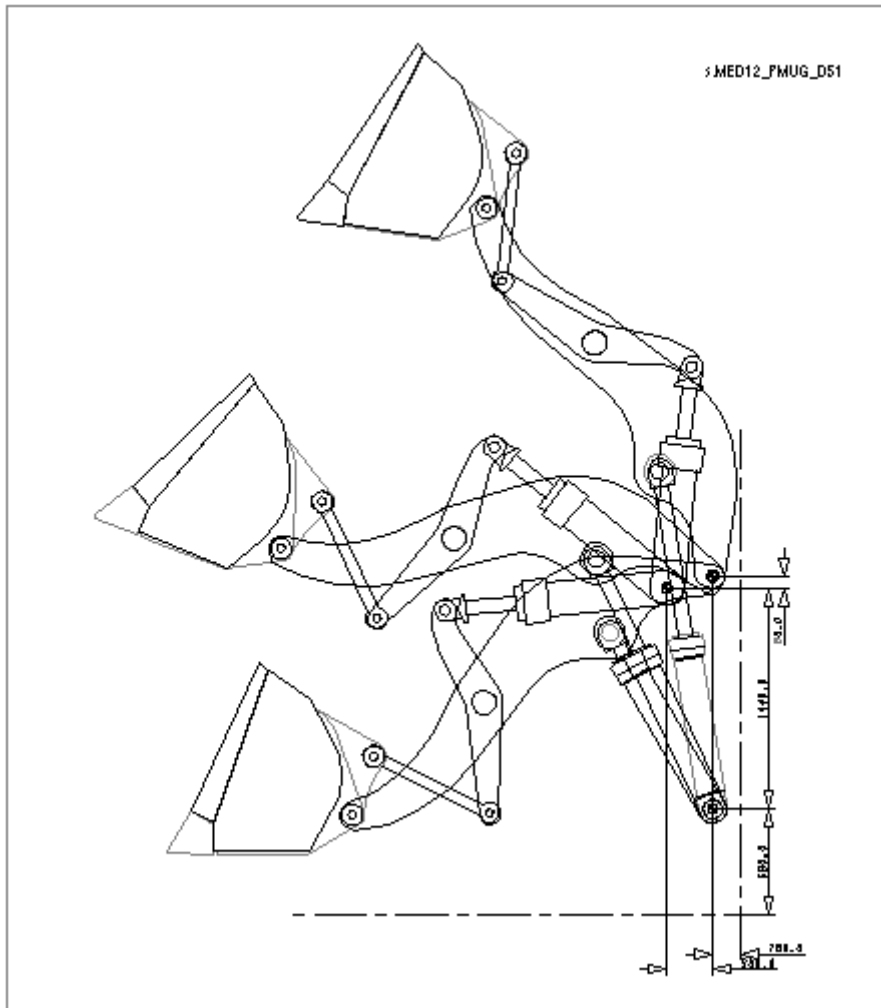


By adding a table of dimensions to the master drawing, all the required sizes may be pre-defined. In this way you can produce a drawing of a particular size of component simply by selecting a set of dimensions from the table.

Simulating Mechanical Movement

You have seen how you can use Parametric Design to change the size of an object on a sheet. By progressively altering one or more dimensions and redrawing the new view over the old, you can create an impression on the screen which simulates the movement of a mechanism. You can then investigate potential collisions as the components move and reciprocate. For example, Figure 4 shows a simulation of the motion of a lift arm assembly for a loader shovel.

Figure 4 Simulation of the Movement of a Lift Arm Assembly



How Parametric Design Works

Parametric Design places some or all of the points that make up an object onto a grid. The grid is moved according to the parameters you specify, either angles or lengths, and this process pulls the points of the object to their new positions.

To use Parametric Design, you do the following:

1. Create a parametric viewbox around the object you want to parameterize.
2. Define a reference point that will remain static during parameterization.
3. Decide if all or only part of the object needs to be parameterized.
4. Dimension the object using MEDUSA dimensioning facilities to place relevant object points on the grid.
5. Check that all the moveable points of the object lie at grid intersections.
6. Exclude some parts of the object from parameterization, if you want to.
7. Enter new parameters.
8. Execute the Parametrization that moves the grid.
9. Check the result.

Viewboxes

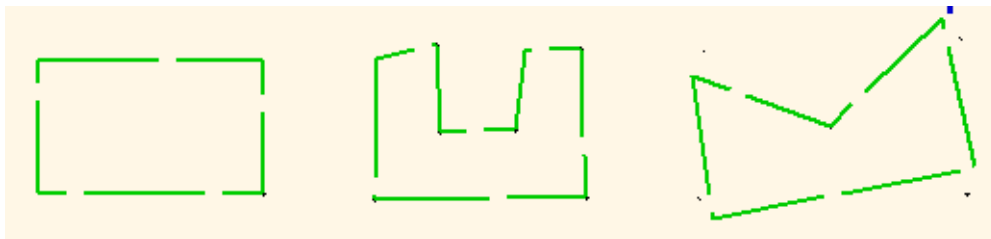
Before you can parameterize any geometry, you must place a parametric viewbox on the sheet around the geometry.

A parametric viewbox is a closed line of type LPV, consisting of straight line segments. Only elements wholly within a viewbox are affected by Parametrization. Viewboxes have the following features:

- They may be any shape
- They must not overlap or be nested

Viewboxes are processed in turn independently of one another, in the order that you place them on the sheet. The maximum number of viewboxes allowed per sheet is 20.

Figure 5 Examples of Parametric Viewbox Shapes



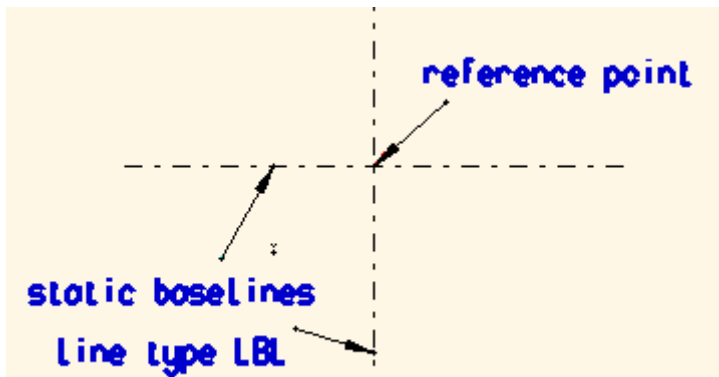
Reference Points

The geometry that you intend to parameterize must have a datum or reference point. This is the point from which all movement occurs. The reference point remains fixed on the sheet during parameterization. You can use intersecting static baselines or a prim to specify a reference point.

Static Baselines

One way of specifying a reference point is using a pair of intersecting static baselines. A baseline is a line of type LBL. A static baseline has static points (FUNV0) at each end. You would normally draw a horizontal baseline and a vertical baseline through the geometry. The intersection of these two lines is the fixed reference point which does not move during parameterization.

Figure 6 Intersecting Static Baselines



Intersecting static baselines such as those shown above in Figure 6 generate six grid lines, of which you can normally see just two. Each baseline supports three grid lines: one along its length, and two at the ends of the line, perpendicular to it. The default length of these perpendicular grid lines depends on the grid tolerance. The default grid tolerance is 0.1 mm, so you cannot normally see these grid lines at all.

PVG Prims

Another way to specify a reference point is using a prim of type PVG. The datum of the prim is the reference point. Horizontal and vertical grid lines are generated through the datum.

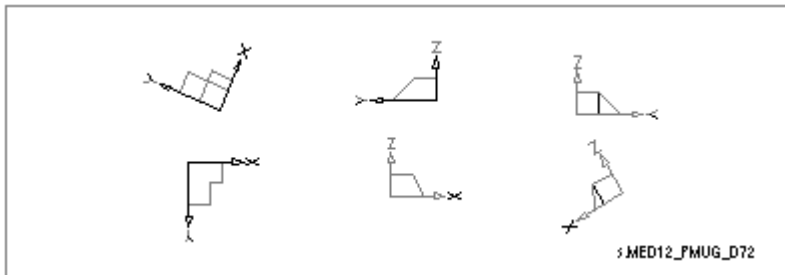
Figure 7 The Parametric Design Datum Prim (Type PVG)



Orthogonal View Prims

You can also use an orthogonal view prim to specify a reference point. The datum of the prim acts as a static reference point and grid lines are generated along the arms of the prim. A worked example showing how to use a view prim as a reference point is given on "Getting Started", "Example 1: Using a Prim" on page 53.

Figure 8 Orthogonal View Prims



Positioning Reference Points

Choosing the right place for a reference point is not always easy. The reference point should usually be at the point from which you would choose to dimension your drawing. In general, a good reference point is a central point on a line of symmetry, for example the center of a circle. Do not choose a point which should move during parameterization.

Static baselines: Static baselines can be drawn at any orientation although they are frequently at right angles, as shown in Figure 9.

Figure 9 Positioning Baselines

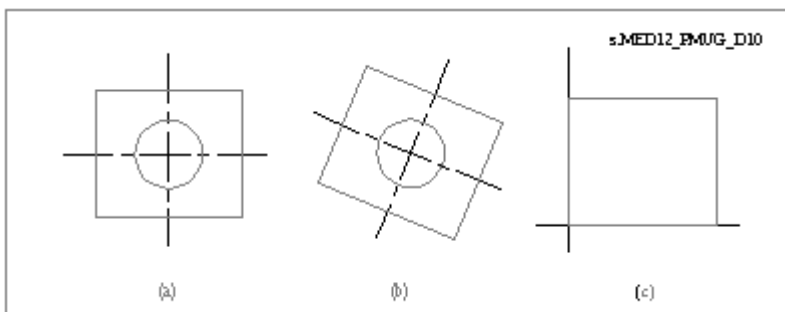


Figure 9 (a)	The center of the circle is also the center of symmetry. This makes an ideal reference point.
Figure 9 (b)	If the component lies at an angle, draw the baselines at the same angle, so that they lie along lines of symmetry.
Figure 9 (c)	The bottom left point is a suitable reference point because it is used when you dimension the width and length. Any point from which dimensions are measured is a useful reference point.

Prims: Figure 10 shows some examples of how you might position a prim in relation to object geometry. You can use a PVG prim or an orthogonal view prim.

Figure 10 Positioning a Prim

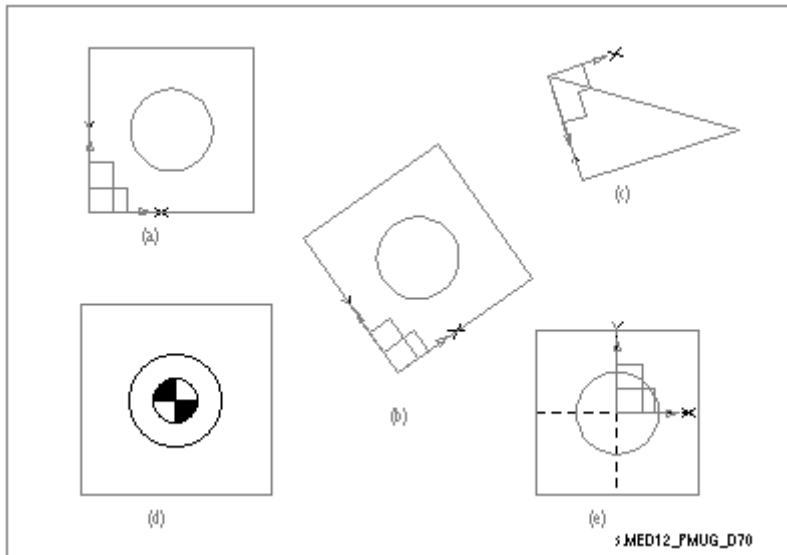


Figure 10 (a)	The datum point of the prim is placed precisely at the bottom left corner point of the object. Grid lines are generated in the direction of each arm.
Figure 10 (b)	If the component lies at an angle, adjust the prim so that its arms run along the lines of the component. To do this, position the prim in its normal orientation then rotate it until it aligns with the component.
Figure 10 (c)	Make sure that at least one of the arms of the prim lies along an existing line. This helps when the grid is drawn at a later stage.
Figure 10 (d)	It is perfectly valid to place a prim at a center of symmetry. Notice that this PVG prim does not lie on any existing lines.
Figure 10 (e)	You may sometimes need to add supporting lines to help create the grid. In this example, two dashed lines have been added.

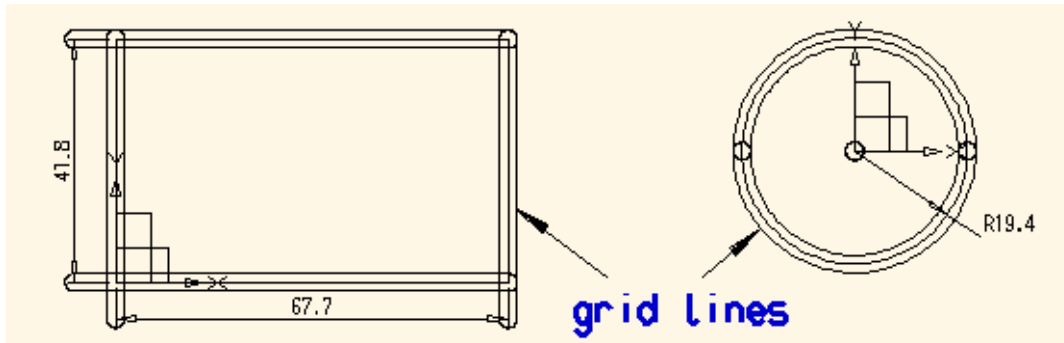
The Parametric Grid

Parametric Design works not by moving individual points but by moving a grid upon which all moveable points are placed. The grid consists of lines at any angles and also of circles. It is not a regular orthogonal grid like the construction grid in the 2D drafting system.

Grid Lines

Parametric grid lines are frankfurter lines, type STK, and can be oriented at any angle and spacing. They are drawn on top of existing lines in your drawing. If your drawing contains diagonals, arcs, or circles then grid lines are also drawn in the shape of diagonals, arcs, or circles.

Figure 11 Parametric Grid Lines



Dimensions and the Grid

You place moveable points upon the grid by dimensioning them. Each dimensioned point in the object geometry is fixed to a grid line intersection. Points that are fixed to grid intersections move with those grid intersections during parameterization.

Looking at Grid Lines

It is very useful to look at the grid that has been calculated each time you create a new dimension group. This enables you to see if the drawing has been adequately dimensioned.

Geometric Tolerance

The system uses a geometric tolerance or grid tolerance to match object points to grid line points. The default tolerance is very small (0.1 mm) and you are advised not to change it.

Some Special Drafting Techniques

You can use any MEDUSA drawing as input to the Parametric Design system. However, you should be aware that you cannot always draw and dimension an object in the normal way and then parameterize it. You may need draw an object in a particular way to avoid constraining the movement of points during parameterization.

This section describes some of the ways you can avoid placing unnecessary constraints on the way geometry is transformed during parameterization. Constraints can occur if you have either coincident or collinear points in the master drawing:

Avoiding Coincident Points

Coincident points in object geometry can cause problems for Parametric Design. If you create points on top of each other in the master drawing, these points will remain coincident whenever you parameterize the drawing. The result may not be what you intend. If there are small details in your drawing, such as fillets, it is best to exaggerate them or the distance between them. This ensures that points are not coincident. The system uses a geometric tolerance to decide whether or not two points are coincident.

Figure 12 shows how you might want to parameterize an object. The drawing on the right can only be generated from the master drawing on the left if the master is drawn correctly.

Figure 12 Possible Parameterization

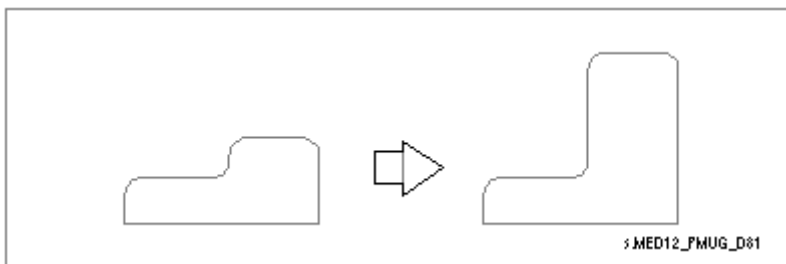
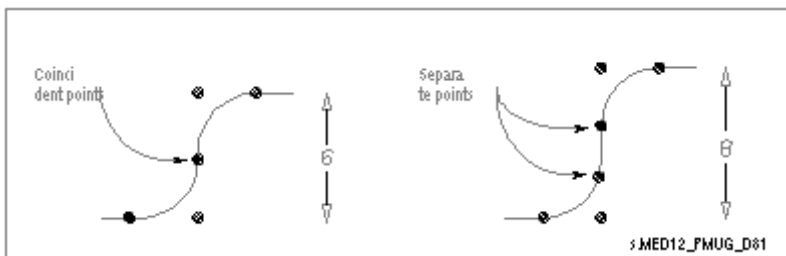


Figure 13 shows the fillet detail and illustrates how to solve the problem of coincident points.

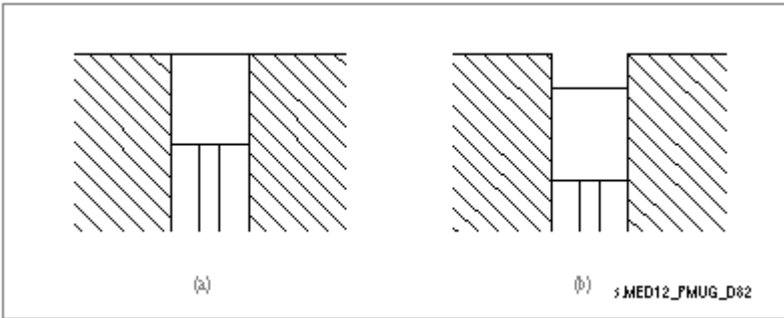
Figure 13 Avoiding Coincident Points



In this example, if you draw two 3 mm radius fillets in an 8 mm gap, the end points are completely separate. This allows you to move the points apart during parameterization, as in Figure 12. However, if you create the 3 mm fillets in a 6 mm gap, the end points are coincident and are locked together. The points cannot move apart during parameterization, and therefore you cannot parameterize the object freely.

Figure 14 shows another example. Here, there are two versions of the master drawing of a mechanism in which a piston is required to move up and down a cylinder.

Figure 14 Coincident Points



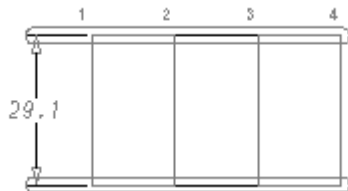
In Figure 14 (a), the piston cannot move during parameterization because its top two points are coincident with the points at the top of the cylinder. To allow the piston to move freely you must draw the piston in the position shown in Figure 14 (b).

Collinear Points

Points are considered to be collinear if a straight line can be drawn through them. During parameterization, collinear points which lie along the same grid line remain collinear even if you attempt to change their orientation during parameterization. The following example illustrates this.

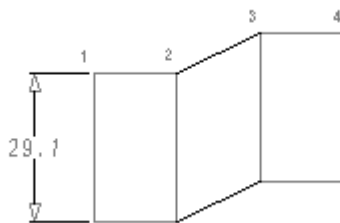
Figure 15 shows geometry containing several collinear points. The vertical dimension generates grid lines running in the same direction as the dimension leader lines. The grid lines appear as frankfurter lines (line type STK).

Figure 15 Collinear Points on the Same Grid Line



Any points lying along these grid lines are kept collinear during parameterization. This means that it is not possible to parameterize the drawing to produce the shape shown in Figure 16. Points 3 and 4 must remain collinear with points 1 and 2. The same restriction applies to points lying along the lower grid line in Figure 15.

Figure 16 Aim of Parameterization

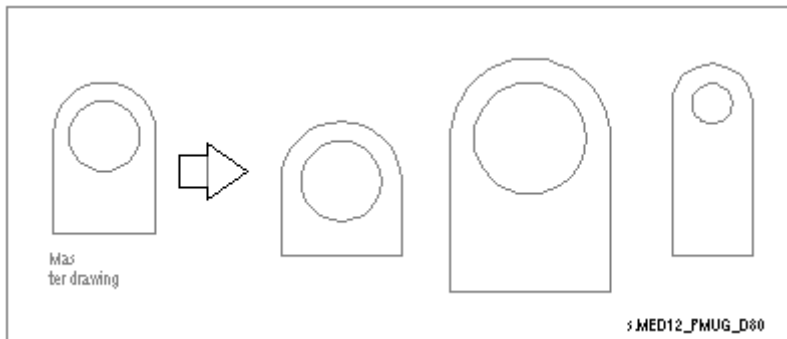


How Geometric Properties are Maintained During Parameterization

Geometric properties are automatically preserved during parameterization. For example, lines that are perpendicular are kept perpendicular. Tangential lines remain tangential. For example, in [Figure 1](#), “[Changing the Dimensions of an Object](#)” on page 10, the straight line segments remain tangential to the arcs whenever this component is parameterized.

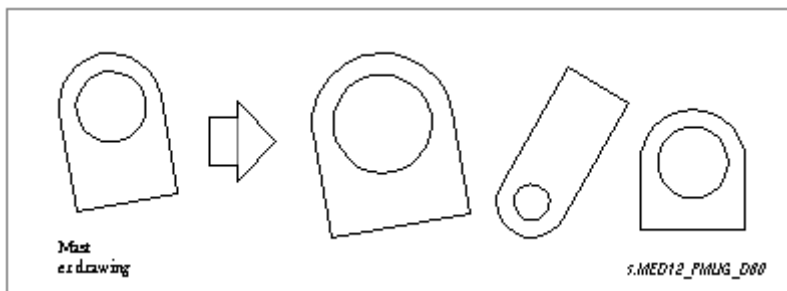
Horizontal and vertical lines also remain in the same orientation. In [Figure 17](#), horizontal and vertical properties of the master drawing are maintained whenever the object is parameterized.

Figure 17 Example of Parameterizing a Specific Case



By **rotating the master drawing** before parameterizing it, you can produce several versions of the master, in a variety of shapes and orientations. For example, the master drawing in [Figure 18](#) has been rotated by a few degrees and consequently there are no horizontal or vertical lines. You can now produce any sized or shaped variant of the original object by parameterizing the master drawing. Notice that this method also allows you to create new objects with vertical or horizontal sides.

Figure 18 Example of an Ideal Master Drawing



In many cases a horizontal master drawing may be sufficient. It is only when you need to rotate the drawing during parameterization that the master drawing must be drawn at an angle. This is particularly important when creating parametric symbols or when simulating mechanical movement. In these situations objects frequently need to be rotated as they are parameterized.

Special Cases

As shown above, a master containing horizontal and vertical lines may be treated as a special case. Restrictions can also be caused by lines drawn at angles of 30, 45, and 60 degrees. These are also regarded as special cases. So if you need to rotate the master drawing, use a fractional angle, such as 13.5 degrees. This ensures that the rotated drawing contains no lines at angles which are regarded as special cases. If a drawing fails to parameterize, always check for special cases, particularly the angles of the lines in the drawing.

Always Parameterize the Master Drawing

When you parameterize geometry, always use the master drawing. You may not be able to parameterize a drawing which is itself a parameterized version of the master. This is because a drawing produced by parameterization may contain restricting features not present in the master drawing. These might be coincident points or lines, or lines at particular angles, or any of the other special cases mentioned above.




















If you want to produce several parameterized versions of an object, create the master drawing and save the sheet. To produce the first parameterized version, take the master drawing and parameterize it. The new version must be stored under a new name, so rename the sheet and save it. To produce the second parameterized version of the object, and all subsequent versions, reload the master drawing and parameterize it with new parameters each time.

Tooltray

Please note: Before you can work with 2D Parametrics, you have to activate the license by choosing Licenses -> 2D Parametrics inside the main menu bar.

The following figure shows the tooltray available inside the graphical user interface of MEDUSA.

Figure 19 2D Parametrics Tooltray

Undo			Control Parametric Layers
Parameterize			Grid
Create Reference Point			Create Viewbox
Create group lines and PPG prim			
Creates parametric symbol attachment text			Parametric Switches and command
Query and modify LCIS variables			Creates user LCIS variables
Display the current settings of the parametric line styles			Modify selected dimension text
Tables			Sets options for fillet behavior
CPI Named Groups			Tolerance Settings
Display Post Param Dialog			Select and parameterize variant

Details on most of the tools is given in the following chapters.

DIMENSIONING

This chapter describes how to prepare a drawing for parameterization by dimensioning the object geometry. This places the object points onto a parametric grid.







- Dimensions in Parametric Design..... 26
- Tolerance Definition 29
- Linear Dimensions 32
- Circle and Arc Dimensions..... 34
- Over-dimensioning and Under-dimensioning..... 36
- Change Dimension Text..... 38

Dimensions in Parametric Design

To place points onto grid intersections you must dimension them. You can dimension all or only part of an object, but all points which are to move must be dimensioned. Points that are not dimensioned generate an error message during parameterization.

Dimension Types You Can Use


Normal dimensioning methods do not always work when using Parametric Design and there are some dimension types that you cannot use. You can use any of the following dimension types to prepare a drawing for parameterization.

- Chain 
- Coordinate 
- Datum 
- Angular 
- Radial 
- Diameter 

Dimensions with tolerances may be parameterized normally. Dual unit dimensions can also be parameterized.

Dimension Types You Cannot Use

You cannot parameterize geometry that has been dimensioned with either of the following dimension types:

- Axonometric dimensions 
- Tolerance limit dimensions (LIM)

If you try to parameterize drawings that contain axonometric or tolerance limit dimensions, you will see the following error message:

Illegal dimension type

Refer to the *MEDUSA Parametric Design Reference Guide* for details of how to convert tolerance dimensions into a format that can be parameterized.

Dimensioning Techniques

When you dimension geometry which will be parameterized, use the following guidelines:

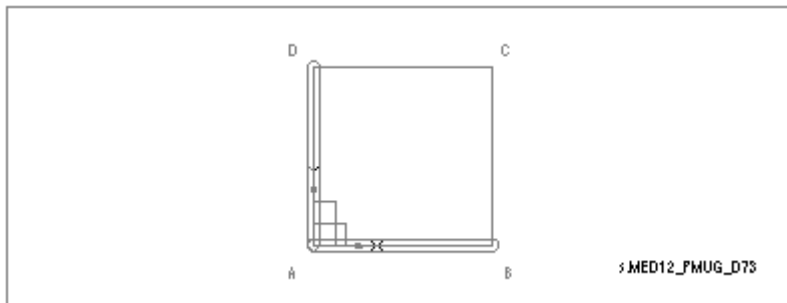
- Always use near probes to create the dimension construction line, except for the first point which can be created with any kind of probe.
- Start by dimensioning points in relation to the reference point. This allows the grid to grow outwards from the reference point.

For every new dimension group you create, one of the points you are dimensioning must already lie at a grid line intersection. The exception to this rule is if you are using chain dimensioning, where you can use center support to place points on the grid (see "[Linear Dimensions](#)", "[Center Support](#)" on page 32).

Example

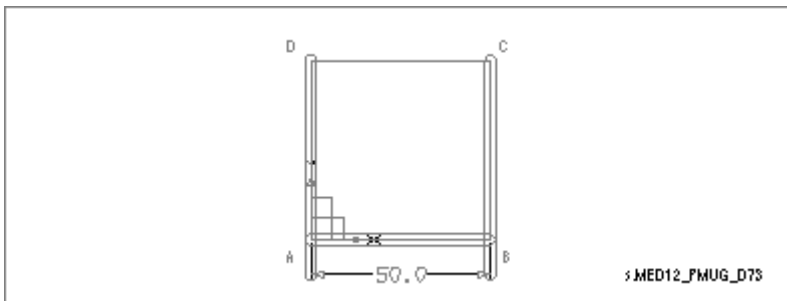
The reference point in Figure 20 generates two grid lines, one horizontal and one vertical, through the datum of the prim. Point A coincides with the reference point, and therefore it is at a grid intersection. Points B, C, and D must be placed at grid intersections by dimensions.

Figure 20 Grid Lines Generated By Reference Point



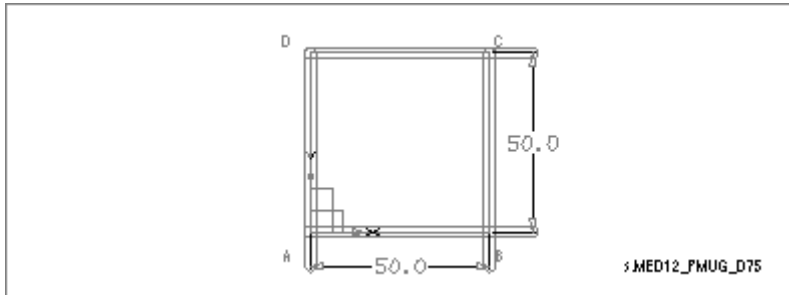
Dimensioning the line AB places point B on the grid, as shown in Figure 21.

Figure 21 Extra Grid Intersection Generated by Dimensioning Line AB



When the line BC is dimensioned as in Figure 22, all the points of the object lie at grid intersections and are said to be supported on the grid.

Figure 22 **Grid Completed by Second Dimension**

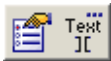


Tolerance Definition

This section shows you how to create permissible upper and lower tolerance for dimensions. For this you can define variable tolerances and its usage for parameterization.

Toolset

Figure 23 Toolset for Tolerance Settings



The tools are:

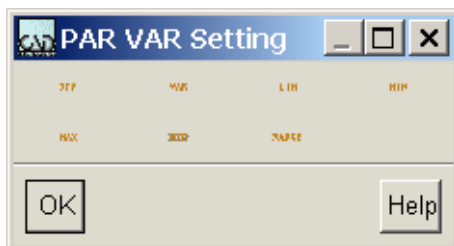
Displays the current tolerance settings

opens the dialog PAR VAR Setting, which allows you to define tolerance settings for the current MEDUSA session. For details see [“Dialog PAR VAR Setting” on page 30](#).

Creates parametric tolerance text

opens the dialog PAR VAR Setting, which allows you to place texts on the sheet, which define the usage of variable tolerances.

Figure 24 Dialog for Placing Tolerance Setting Texts



From left to right the succession of switches is the same as it is in the dialog PAR VAR Setting from top to bottom, explained below.

Dialog PAR VAR Setting

PAR VAR Settings are used for specifying how the system calculates new dimension values during parameterization for those dimensions, which have upper and lower tolerances defined.


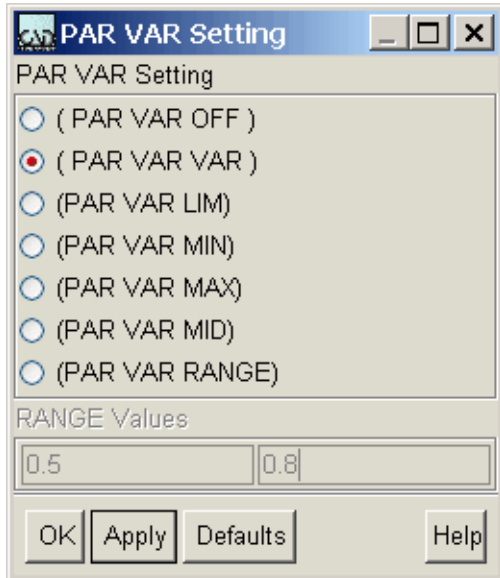
The following dialog opens if you *click left* on the tool  Displays the current tolerance settings.

Figure 25 Dialog for Tolerance Settings



The following list explains the options:

Option PAR VAR	Explanation	Example
OFF	Specifies that any changes made to the original tolerance text are ignored. The original tolerance appears in the parameterized drawing.	<pre>LET A = 7 LET B = 3 A+B 87.45 B-A</pre>
VAR	Ensures that any expressions which have replaced original tolerance text are evaluated during parameterization. The resulting value is used to calculate the new dimension and resulting values are placed in the new dimension. This is the default setting.	<pre> +10 87.45 -4</pre>
LIM	Converts VAR tolerance dimensions to LIM format. The new tolerances are calculated by evaluating the original tolerance texts and add them to the nominal value.	<pre>97.45 83.45</pre>

Option PAR VAR	Explanation	Example
MIN	Specifies that the lower tolerance is used for all tolerance dimensions. In some cases, such as holes, you may need to indicate maximum or minimum diameters explicitly. You can do this by editing the individual tolerance texts.	<u>83.45</u>
MAX	Specifies that the upper tolerance is used.	<u>97.45</u>
MID	Calculates the tolerance using the following formula: $(MAX - MIN) * 0.5 + MIN$	<u>90.45</u>
RANGE	Calculates the tolerance using the following formula: $(MAX - MIN) * factor1 + MIN + factor2$ factor1 must be a factor in the range 0 to 1. factor2 is an optional extra factor which can get any value. If it is left out it is set to 0.	<u>90.45</u> factor1=0.5 no factor2

All the options are explained in detail in the *Parametrics Reference Guide*, chapter „Dimensioning“.

Create Variable Tolerances

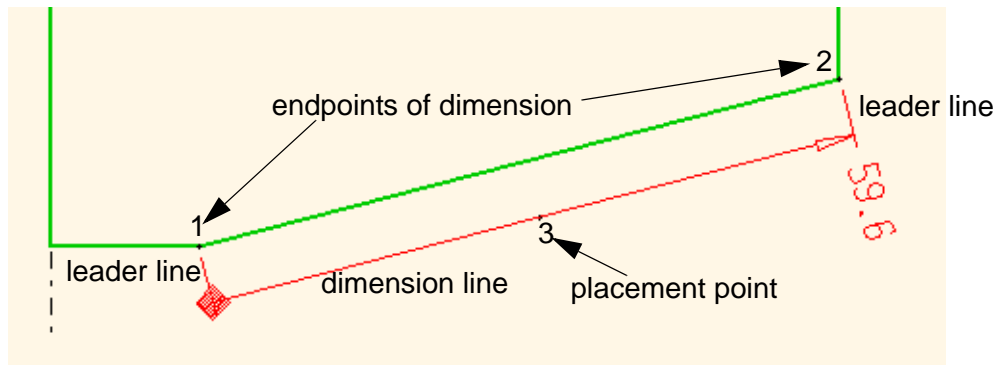
For creating variable tolerances do the following steps:

1. Select a dimension.
2. Open the properties window with the *right mouse button*.
3. In the section Dimension and Tolerance style on top of the dialog choose the icon Tolerances shown as dimensions plus variations.
In the bottom of the dialog the entry fields below Tolerance become active.
4. Type in the expression for the upper tolerance in the upper field (e.g. A+B).
5. Type in the expression for the lower tolerance in the lower field (e.g. A-B).
At this step we assume that the variables A and B are defined inside the sheet as command texts of style Parametric command text, e.g. LET A=7 and LET B=3.
6. Choose the button OK for applying the settings and closing the properties dialog.
7. Now define the variable tolerance settings. You have the following options:
 - a. Choose the tool Displays the current tolerance settings for defining the tolerance settings for the MEDUSA session.
 - b. Choose the tool Creates parametric tolerance text for placing the tolerance definitions on the sheet. Such a text overrides the settings inside the dialog PAR VAR Setting.
8. Now you can parameterize your geometry (see ["Getting Started"](#), ["Parameterizing an Object"](#) on page 50).

Linear Dimensions

Linear dimensions, for example chain or datum, generate grid lines along the leader lines. When you create linear dimensions, the end points of the dimension must be at grid line intersections. The placement point does not need to be at a grid line intersection. Figure 26 shows the significant points of a linear dimension.

Figure 26 A Linear Dimension



Which Dimensions Types To Use

Horizontal and vertical: Use horizontal and vertical dimensioning only if the part of the object being dimensioned is to remain horizontal or vertical.

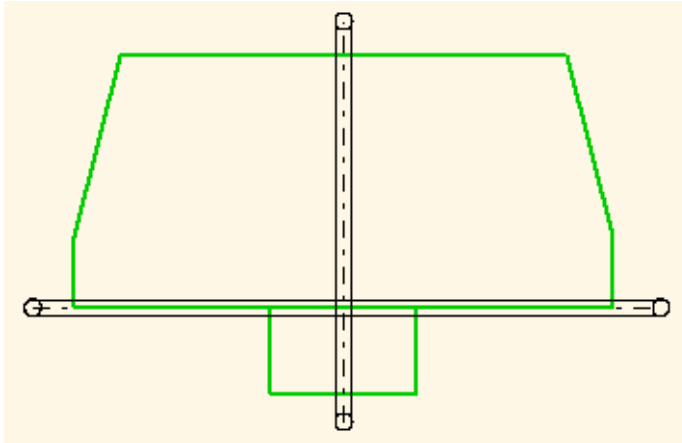
Parallel and perpendicular: If you intend to rotate an object during parameterization, use parallel or perpendicular dimensioning. This is important when working with parametric symbols, which are often rotated as they are loaded (see [“Symbols” on page 133](#)).

Center Support

Normally, you dimension geometry outwards from the reference point. If the reference point does not lie on a line of geometry, but instead lies at a center of symmetry then you can create certain chain dimensions using center support. This means that the reference point does not have to be specifically dimensioned. Center support is possible when an existing grid line perpendicular to the dimension passes through the center of the dimension.

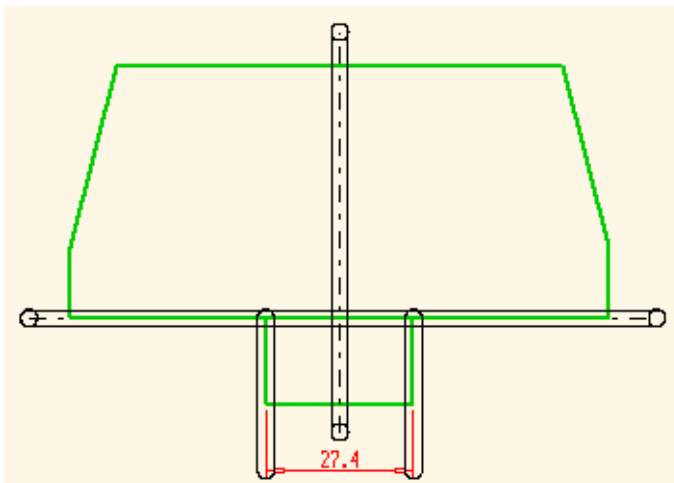
In the example in Figure 27, the reference point is the intersection of two static baselines. A horizontal baseline passes through the center of diameter dimension on the left and does not coincide with any lines of geometry.

Figure 27 Object With Intersecting Baselines



To dimension the diameter on the left of the object in Figure 27 from the reference point you would need to use a three point dimension construction line including points A and B and the reference point. Using a chain dimension you can use a two point construction line from points A to B. Figure 28 shows how the diameter dimension is supported by the horizontal baseline.

Figure 28 Chain Dimensioning With Center Support



Datum and Coordinate Dimensioning

Datum and coordinate types of dimensioning can only be drawn horizontally or vertically and so you should not use them when an object is in a rotated position or will be rotated at a later stage.

Unlike chain dimensions, datum and coordinate dimensions cannot use center support.

Circle and Arc Dimensions

You can create circle and arc dimensions in the usual way.

For fillets, you can dimension each fillet individually or you can use the PARFIL in-sheet command to set fillet dimensions automatically during parameterization. This offers the advantage that the whole geometry can be parameterized with its fillets without dimensioning each fillet individually


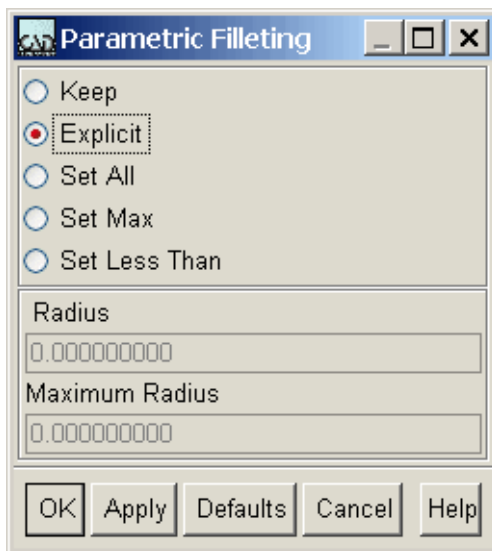
2D Parametrics provides the following dialog for defining filleting settings for undimensioned fillets. It is opened by the tool Sets options for fillet behavior during parameterization .

Figure 29 Dialog for Filleting Settings



Please note: The options of the dialog are valid for undimensioned fillets. Fillets which have a dimension or for which dimensions are set automatically with a PARFIL command are parameterized as defined.

- Keep
If there is no dimension for the fillet, it is taken without changes.
- Explicit
(default) If this option is chosen only those fillets are parameterized which have a value or expression defined explicitly, e.g. with a PARFIL text (see example on the next pages). Undimensioned fillets produce error messages.
- Set All
If set, undimensioned fillets get the defined Radius.
- Set Max
If set, undimensioned fillets which are smaller than or equal to the Maximum Radius are

taken without changes. Undimensioned fillets which are larger are set to Maximum Radius.

Set Less Than

If set, undimensioned fillets which are smaller than or equal to the Maximum Radius get the Radius.

Radius

is the value used for undimensioned the fillets for the options Set All and Set Less Than.

Maximum Radius

is the maximum value for undimensioned fillets used for the conditions Set Max and Set Less Than.

Please note: You find a complete example for parameterizing components with fillets in chapter "Simple Parameterization Examples", "Example Two" on page 68.

Over-dimensioning and Under-dimensioning

When you prepare a drawing for Parametric Design, it is important that all the points you want to scale are placed onto the grid using dimensions. However, it is possible to over-dimension an object so that conflicts prevent the Parametric Design system from working correctly. Similarly, if the object is under-dimensioned, you will not be able to parameterize it.

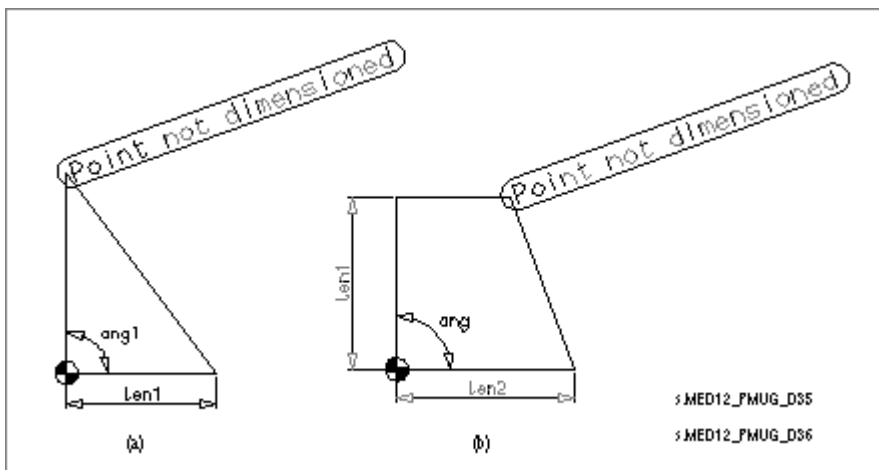
Under-dimensioning

Under-dimensioning means that you have not dimensioned all the points in the geometry and, therefore, there are some points that are not on the grid. Adequate dimensioning ensures that a grid line is drawn on top of every line in the object, and that every point lies at a grid line intersection. If you try to parameterize a drawing which is inadequately dimensioned, you receive the following error message on the sheet with its datum at the offending point:

Point not dimensioned

Figure 30 shows some examples of objects which are under-dimensioned and the error messages resulting from trying to parameterize these objects with new dimension parameters.

Figure 30 Examples of Under-Dimensioned Drawings



Neither of these could be parameterized because not enough dimension information is provided:

Figure 30 (a)	The apex point is not dimensioned.
Figure 30 (b)	The position of the top right point is uncertain.

You cannot draw these shapes from the information provided. Therefore Parametric Design cannot draw them either.

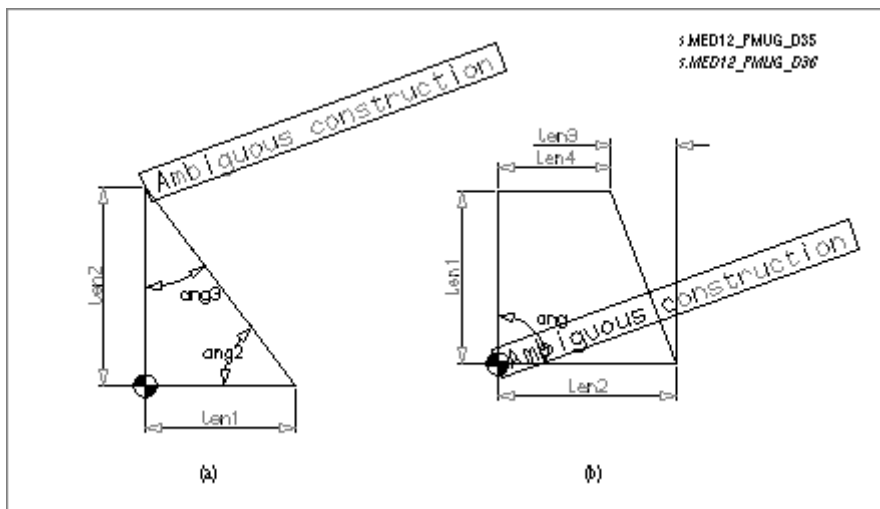
Over-dimensioning

It is a common error to dimension an object so that some of the object points are located at an intersection of more than two grid lines. You will realize that you have made a mistake when you try to parameterize the object. You will receive the following error message written onto the sheet with its datum point indicating the problem point:

Ambiguous construction

Figure 31 shows some examples of objects which have been over-dimensioned and the error messages resulting from trying to parameterize them. In both cases, it is possible to specify values for the dimension variables which would make the objects impossible to draw.

Figure 31 Examples of Over-dimensioned Drawings




In both of these cases, too much information is given:

Figure 31 (a)	Any changes to the angle dimensions could conflict with the linear dimensions.
Figure 31 (b)	The sum of dimensions len3 and len4 must equal len2.

Dimensioning Hints

To avoid over-dimensioning, study the geometry and plan dimensions carefully. Keep in mind that each object point needs to lie at a simple grid intersection of no more than two grid lines. You must give enough information for the whole of the drawing to be created without conflict or ambiguity.


Change Dimension Text

If you want to change texts of dimension groups, for example to replace numerical values by variables or re-name a variable, the tool Change selected dimension text  is provided.

Please consider the following when using this tool:

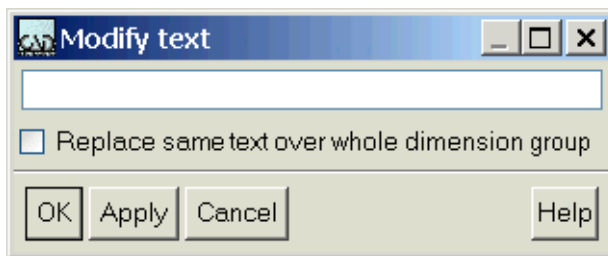
- When selecting several dimension groups only texts in the first selected group are changed.
- Only texts will be replaced, which have the text string found first in the dimension group. Texts with other strings remain unchanged.
- If the first dimension text differs from the others, only this first text will be changed.

Following steps show how to change texts of a selected dimension:

1. Select a dimension.
2. Choose the tool Change selected dimension text .

The following dialog opens:

Figure 32 Dialog for Changing Texts



3. Enter the new text.
4. Switch on the option Replace same text over whole dimension group.
If this option is off, only the first text in the dimension group will be replaced.
5. *Click left* on the button OK or Apply.
In both cases the change of texts is executed. OK additionally closes the dialog. With Apply the dialog remains open.

GETTING STARTED

This chapter contains two simple examples which show you how the MEDUSA Parametric Design system works. Before working through the examples in this chapter, you should have read about viewboxes, reference points and grid lines in "Introduction", "How Parametric Design Works" on page 14.

- Overview 40
- Creating a Viewbox..... 41
- Creating a Reference Point..... 43
- Dimensioning 45
- Building the Parametric Grid 46
- Examining the Parametric Grid 48
- Entering Your Own Parameters 49
- Parameterizing an Object 50
- Example 1: Using a Prim 53
- Example 2: Using Baselines 57

Overview

The following procedure gives an overview for simple parameterization of a part. You can perform steps 1 through 4 in any order.

Step	Action	What You Do
1	Draw the object you want to parameterize.	Use any line type.
2	Draw a parametric viewbox on your sheet.	Enclose the objects you wish to scale with a line of type LPV. See “Creating a Viewbox” on page 41 .
3	Define a datum or reference point which will remain static during parameterization.	Place a prim or two intersecting baselines at your chosen reference point. See “Creating a Reference Point” on page 43 .
4	Place all object points on the grid.	Dimension the component. See “Dimensioning” on page 45 .
5	Check that the grid supports each of the points to be scaled.	See “Building the Parametric Grid” on page 46 and “Examining the Parametric Grid” on page 48 .
6	Give new values for the dimensions you want to change.	Place the new value into the text edit field. See “Entering Your Own Parameters” on page 49 .
7	Parameterize the object.	Use temporary parameterization until you are satisfied with the parameterized drawing. See “Parameterizing an Object” on page 50 .

Creating a Viewbox

Before you can use the Parametric Design system you must have a sheet displayed on the screen. The sheet must contain a viewbox which surrounds the object you intend to parameterize. If you use a standard drawing sheet, the sheet may already contain a viewbox. The viewbox may be difficult to see at first because it coincides with the outline of the sheet, and some parts of it are invisible.

Toolset

The following figure shows the toolset for creating view boxes.

Figure 33 Toolset for Creating Viewboxes



The tools are:

Creates parametric viewbox lines

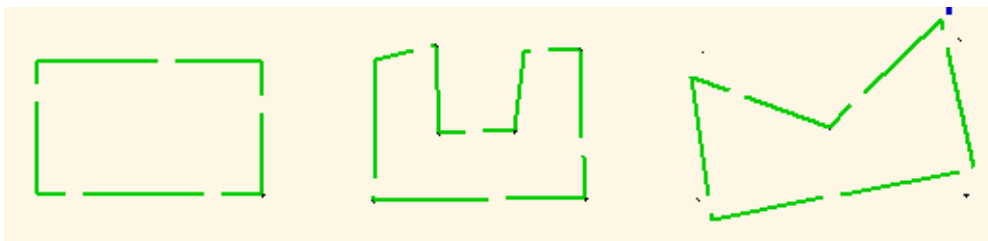
Creates parametric viewboxes

Both tools create a line of style *Parametric view line*, type LPV. This line encloses the geometry which shall be parameterized.

How to create a viewbox

If you are not using a standard sheet, or if your standard sheet does not have a viewbox, you must create a line of type LPV and draw a viewbox for yourself. Use the tools given above. The viewbox can be of any shape, as shown below.

Figure 34 Examples of Parametric Viewbox Shapes



Placing geometry inside the viewbox

When the drawing is parameterized, only those parts of the drawing inside the viewbox are affected. The geometry which you intend to parameterize must be completely inside the viewbox. All dimensioning, grid lines, and baselines must also be inside the viewbox.

Please note: Note that after parameterization all elements should be still inside the view box otherwise you get error messages for those parts which leave the view box.

Number of viewboxes

You can place up to 20 viewboxes on a sheet, but they must not overlap or be nested. Each viewbox is processed separately.

Creating a Reference Point

The geometry that you intend to parameterize must have a datum or reference point. This is the point from which all movement occurs. The reference point remains fixed on the sheet during parameterization.

Toolset

The following figure shows the toolset for creating reference points.

Figure 35 Toolset for Creating Reference Points



The tools from left to right are:

DXV prim to act as a static reference point

attaches a DXV prim to the cursor for placing as reference point for parameterization.

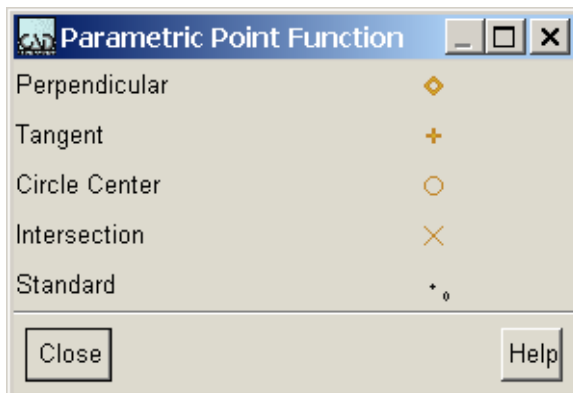
PVG prim to act as a static reference point

attaches a PVG prim to the cursor for placing as reference point for parameterization.

Creates static base lines

activates line creation and opens the following dialog:

Figure 36 Parametric Point Function



The point functions in this dialog are disabled until you placed the first point of the base line by probing into the sheet. If you have done this, the point functions are enabled.

A point function is always applied to the point you have drawn last.

If you probed the first point of a baseline and then choose a point function, it is not applied to the first point until you have drawn the second point.

If you probed the second point and then choose a point function, the function is applied to the current second point immediately. This is also valid for all further points of the baseline.

For details on the meaning of the point functions, see the next section.

Point Functions

The point functions at the ends of baselines indicate how the geometry is constrained when the drawing is parameterized.

The available point functions are:

Perpendicular

Perpendicular point, point function FUNV 10.

This point function guarantees that the corresponding segment on which it is placed is perpendicular to the element segment on which the starting point of the appropriate baseline is.

Tangent

Tangent point, point function FUNV 12.

This point function guarantees that the segment on which it is placed is tangential to the element segment on which the starting point of the appropriate baseline is.

Circle Center

Circle Center point, point function FUNV 26.

This point has to be on a grid line and it marks the center of a circle.

Intersection

Intersection point, point function FUNV 11.

This point marks an intersection of existing grid lines.

Standard

Static point, point function FUNV 0.

If this point function is at both ends of a line it is a static baseline, which does not move during parameterization.

Details and examples for the point functions are given in the *Parametrics Reference Guide*, chapter „Geometric Constraints“.

Dimensioning

For parameterizing geometry you need to dimension it and then replace the dimension texts either with new values or expressions. For details on dimensioning see the appropriate chapter in the *MEDUSA Drafting Guide*.

Dimensioning for parameterization was already explained in [“Dimensioning” on page 25](#).

Building the Parametric Grid

Grid lines are generated first through any reference points in the drawing, through the prim datum or along the static baselines. Then they are generated through points that you have dimensioned.

Toolset

The following figure shows the toolset for drawing grids.

Figure 37 Toolset for Drawing Grids



The tools from left to right are:

Draws complete circular lines for arcs

Draws tangent lines for all tangent arcs

Draws grid lines over fillets affected by PAR FIL

This tool works only on tangent arcs and if a parametric command PAR FIL exists. It is not applicable on circular arcs.

Draws the grid corresponding to the original drawing

Displays the grid built up from the reference point and any dimensioning. Grid lines are frankfurter-shaped lines on layer *Graphical Error Messages* and they are drawn over the geometry. Grid lines created with this tool are placed permanently on the sheet but you can delete them with the tool *Parametric Layers* (see "[Layers and Parametric Switches](#)", "[Parametric Graphics Control](#)" on page 111).

Draws the potential grid

Displays the grid built up from all lines inside the viewbox. Grid lines are frankfurter-shaped lines on layer *Graphical Error Messages* and they are drawn over the geometry. Grid lines created with this tool are placed permanently on the sheet but you can delete them with the tool *Parametric Layers* (see "[Layers and Parametric Switches](#)", "[Parametric Graphics Control](#)" on page 111).

Draws baselines inferred when the baseline switch is on

This tool makes visible constraints which are used automatically for parameterization. If you want, the drawn baselines can remain inside the drawing.

Draws lines connecting all collinear straight lines

This tool creates lines connecting collinear not-overlapping line segments. Lines drawn with this tool are temporary. Refreshing the graphics deletes these lines.

Draws lines along all segments and completes circles for arcs

Lines are drawn along each segment finishing at the viewbox. Arcs are represented by their appropriate circles. Lines drawn with this tool are temporary. Refreshing the graphics deletes these lines.

Draws the new grid using parameters specified

Grid drawn with this tool corresponds to the parameterized (new) geometry.

Draws the grid corresponding to the original drawing and undoes it

Works the same way as the tool `Draws the grid corresponding to the original drawing` but grid lines remain visible only until you refresh the screen.

Sequences through the grid lines

This tool assumes that you already have drawn a grid, for example with the tool `Draws the potential grid`.

This tool is used for showing the built of a grid. It displays the grid lines one after another. If you choose this tool a dialog opens. In there, each time you click on the button `Next` a further grid line is displayed in addition to already drawn ones. Each time you click on `Previous` an already drawn grid line is removed. If you click on `Start`, all grid lines are removed and you are back at the beginning of building the grid, before drawing the first grid line.

Details on using this tool are given in [“Examining the Parametric Grid” on page 48](#).

When to draw the grid

You can dimension a drawing one step at a time, displaying the grid at each stage, or you can dimension the whole of the drawing then draw the grid as a final check.

Throughout this manual, the grid is displayed after each new piece of dimensioning is added. This is to help you to understand how grid lines are created. While you are learning to use the system it is better to add one dimension at a time, and then use `Draws the grid corresponding to the original drawing and undoes it` to display the grid so far.

Deleting grid lines

When you have checked to see that the grid is developing adequately, you should remove all grid lines.

If you used the tool `Draws the grid corresponding to the original drawing and undoes it` for drawing the grid, you only need to refresh the graphical area for deleting the displayed grid lines.

For the other tools use the `Parametric Layers` dialog:

1. Open the dialog `Parametric Layers` (see [“Layers and Parametric Switches”](#), [“Parametric Graphics Control” on page 111](#)).
2. Switch on the option `Grid Lines` and *click left* on the button `Delete`.
The permanent grid lines are removed immediately.

Examining the Parametric Grid

The following procedure will help you if you have any problems with building the grid or if you want to see how the grid is developing. You may find it useful to perform them after you have completed each example in this chapter. This diagnostic routine uses the tools *Draws the grid corresponding to the original drawing* and *Sequences through the grid lines* (see [Figure 37, “Toolset for Drawing Grids” on page 46](#)), so grid lines and any error texts are drawn permanently on the screen.


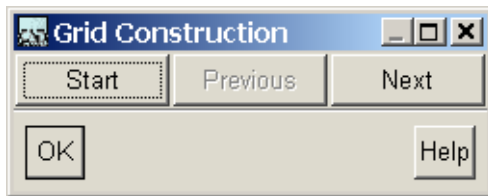
1. Refresh the screen with the *Refresh Graphics* tool  from the toolbar to remove any temporary grid lines.
2. Choose the tool *Draws the grid corresponding to the original drawing* to draw grid lines permanently.
3. Choose the tool *Sequences through the grid lines*.
The following dialog opens.

Figure 38 Dialog Grid construction



- *Click left* on the button *Next* to display the first grid line. If you *click left* *Next* again, a further grid line is drawn and so on.
 - *Click left* on the button *Previous* to hide the currently displayed grid line. If you *click left* *Previous* again, a further grid line is hidden. The button *Previous* is active until no grid line is displayed anymore.
 - *Click left* on the button *Start* to return to the beginning of examining grid lines. No grid line is displayed.
 - *Click left* on the button *OK* to close the dialog and display all grid lines.
4. You can remove the permanent grid lines as explained in [“Deleting grid lines” on page 47](#).

Entering Your Own Parameters

Before you can create different parameterized versions of a component, you must replace the original dimension text values with your own parameters.

In this section the value will be replaced by another value. But you also can use

- variables and expressions, which is explained in detail in [“Variables and Expressions” on page 91](#), and
- tables to store one or more groups of values on a sheet which is described in [“Tables” on page 117](#).

How to Enter Your Own Parameters

Use the following procedure to enter your own parameters:

1. Make the dimension text current.
2. Place the new dimension value into the text edit field below the dashboard.

Prefix and Suffix Texts

When you replace dimension values with your own parameters, be careful which text string you alter. Dimension texts consists of the numeric value text and optional prefix or suffix texts. These give you more information about the dimension, for example whether it is a radius or a diameter:

- Prefix text is placed before the numeric value. Examples are R, DIA, and j.
- Suffix text is a separate text element (type TDM) placed after the numeric value, for example the R and DIA texts in the ANSI dimensioning standard.

The Parametric Design system only uses the numeric value text when it calculates the new dimensions of a component. You must replace this text with your own parameters. Prefix and suffix texts are ignored when the dimension is evaluated and you can delete them if you want to make the drawing clearer.

Dimensions After Parameterization

When the dimension is redrawn after parameterization, the dimensions appear in the style you used originally. The value or expression you use to specify the new value does not affect the final format, which will have prefix and suffix texts corresponding to the original drawing. For example, you could replace a dimension of 25.0 DIA with the numeric value 40 (the DIA text can be either ignored or deleted). After parameterization, the dimension appears as 40.0 DIA.

Parameterizing an Object

When you have placed all of the object points onto the parametric grid by dimensioning them, and replaced some of the dimension texts with your own parameters, you are ready to parameterize the geometry.

Toolset

The following figure shows the toolset for parameterization.

Figure 39 Toolset for Parameterizing



The tools from left to right are:

Parameterizes the geometry

Permanently parameterizes geometry inside the viewbox using the parameters you have specified in the dimension groups. The geometry is stretched, shrunk or rotated to fit the new values. The output is a drawing that is fully dimensioned.

Parameterizes the geometry then immediately undoes

Works the same way as the tool Parameterizes the geometry except that it is cancelled immediately. The parameterized graphics remain visible on the screen until you refresh the screen. Always use this tool until you are sure that you want parameterization to be permanent. If you cancel the parameterization, you can experiment with different parameters, returning the geometry back to its original shape each time

Simulates mechanism motion

opens the dialog for defining start and increment values for variables in order to do several different parameterizations on a geometry. For details see [“Simulating Mechanisms” on page 189](#). Parameterization during simulation is done like using the tool Parameterizes the geometry then immediately undoes.

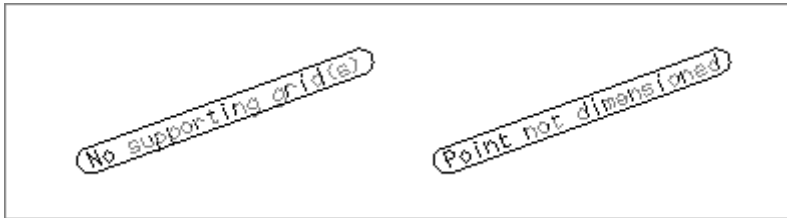
Please note: The format of dimension text on the parameterized drawing is the same as in the original drawing. Any prefix and suffix is retained and the original number of decimal places is displayed regardless of the format you gave when you supplied the new parameters.

Parameterization Errors

If the drawing is satisfactory, each part of it will be highlighted on the screen as it is processed. If there is a problem then one or more error messages of text type TS1 or TR1 will be placed on the sheet so that the datum point coincides with the point that is causing the problem. This enables you to identify any problem points and to correct the drawing. If an error message appears, you must diagnose the problem and correct it before proceeding.




Error messages are placed on layer 99. Figure 40 shows two of them.

Figure 40 **Error Messages**



Common error messages

You will probably see the following two error messages frequently while you are learning to use Parametric Design.

No supporting grid(s)	This message may appear when you use the tool  Draws the grid corresponding to the original drawing and undoes it  . The reason is a point somewhere in the object which is not fixed to a grid intersection. The datum of this error text is placed on the offending point. This error is usually caused when, although you have dimensioned a point, it is not at a grid intersection.
Point not dimensioned	This message may appear when you use the tool  Parameterizes the geometry. The cause is an inaccurately dimensioned or completely undimensioned point. Always investigate problem points by zooming in to check that the dimensioning is accurate. You may have to delete the dimensioning and draw it again. A typical cause of inaccurate dimensioning is the failure to use near probes (where they are required) when drawing a dimension construction line.

These are just two of the most common errors. A full list of possible error messages is given in [“Appendix B Error and Warning Messages”](#) on page 217.

Removing Error Messages

When you cancel a drawing grid or parameterization function immediately, any error messages will disappear when you refresh the screen.

If you do not cancel the command, error messages are written onto the sheet permanently. To remove them, delete the layer on which they are drawn, layer 99. Layer 99 contains grid lines and error messages.

To delete permanent error messages:

1. Open the dialog `Parametric Layers` (see "[Layers and Parametric Switches](#)", "[Parametric Graphics Control](#)" on page 111).
2. Switch on the option `Grid Lines` and *click left* on the button `Delete`.
The permanent grid lines are removed immediately.

Example 1: Using a Prim

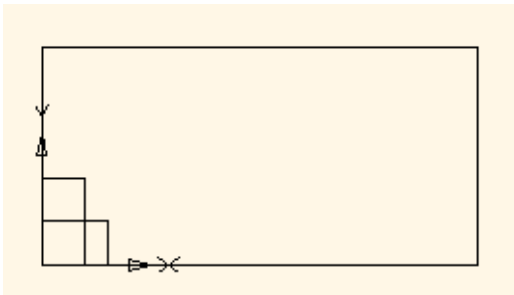
This example shows you how to parameterize a rectangle using a prim of type DXY as the reference point. The master drawing can be used to produce rectangles of any given size.

1. Draw a rectangle, as in Figure 41. The length of the sides is not important.

Adding a Reference Point

2. Create a prim by choosing the tool DXY prim to act as a static reference point from the tooltray MEDUSA Parametric.
3. Position the new prim on a corner point.
 The arms of the prim are aligned with the outline of the rectangle, so there is no need to rotate the prim into a better position.

Figure 41 Example1: Rectangle with DXY Prim Reference Point




When the rectangle is parameterized, the corner point will remain fixed. The rectangle will grow larger or smaller as the other three points are moved.

Drawing the Viewbox

4. Create a viewbox using the tool *Creates parametric viewbox lines* or *Creates parametric viewboxes*.

Please note: Make sure that the rectangle is inside a viewbox.

Displaying the Grid

5. Choose the tool . Figure 42 below shows what happens.

Grid lines are drawn along any lines that coincide with the arms of the prim. In this case, two grid lines are drawn. The prim supports the two grid lines.


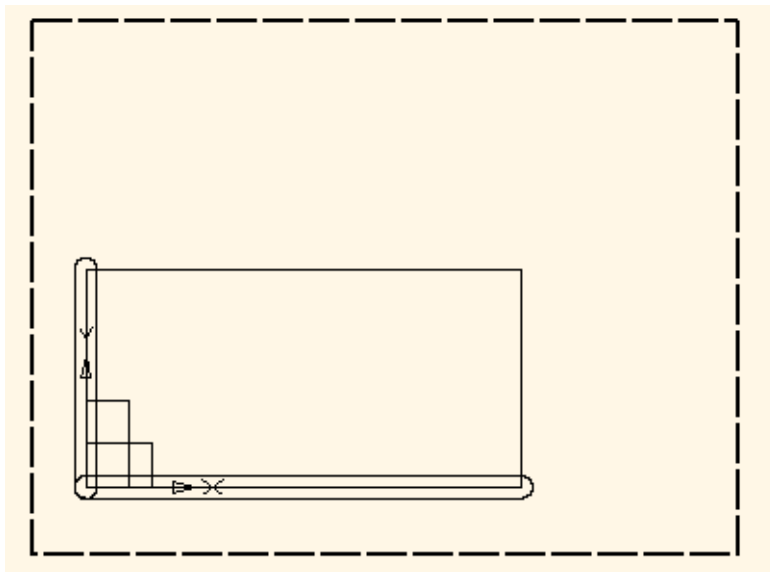

Please note: Notice that the grid lines are not permanent. If you refresh the screen by choosing the tool Refresh graphics  from the toolbar the grid lines disappear.

Figure 42 Example1: Effect of Drawing Grid



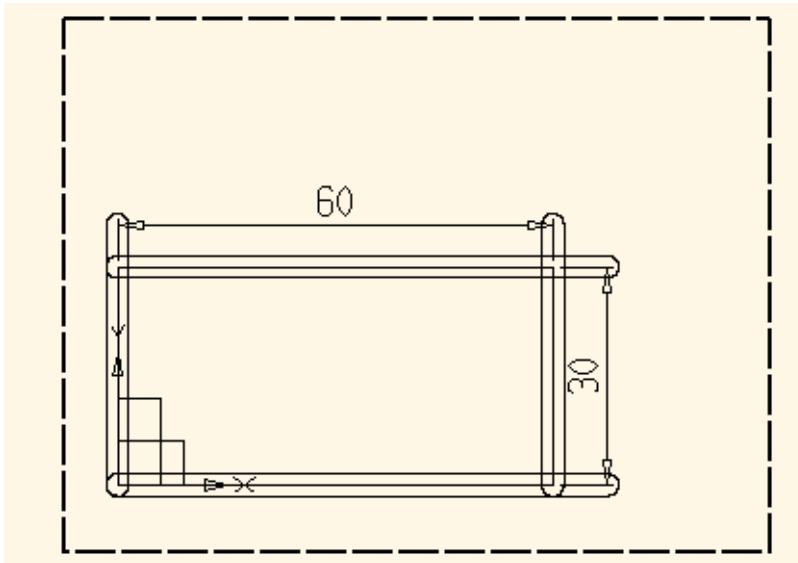
Dimensioning the Object

The rectangle is not adequately supported by the grid at the moment. Only the bottom left corner point lies at a grid intersection. Remember that the aim is to make sure that all points lie at grid line intersections.

6. Dimension two sides of the rectangle, one vertical and one horizontal.
7. You can see how the grid develops by using the tool  each time you create a new dimension group. When the rectangle is adequately dimensioned, every point will lie at a grid intersection.


If you see any error messages while you are creating the dimensioning refer to [“Parameterization Errors” on page 51](#).


Figure 43 Rectangle Dimensioned for Parameterizing



Please note: There is no message to tell you that the rectangle is adequately dimensioned and supported by the grid. It is up to you to decide for yourself. When you are satisfied that the grid is adequate, you can go on to replace the dimensions with new parameters.

Testing the Definition Drawing

Before specifying any new parameters, test the drawing to make sure it can be parameterized using the tool . If an error message appears, you must diagnose the problem and correct it before proceeding. See [“Parameterization Errors” on page 51](#) for a description of some possible errors.

8. Choose the tool . You should not get any error message in order to be sure that you can parameterize the drawing.

Inserting Your Own Values

9. Replace the dimension text values with your own values by clicking on a dimension and then replace the value in the text edit field below the dashboard.

Parameterizing the Object


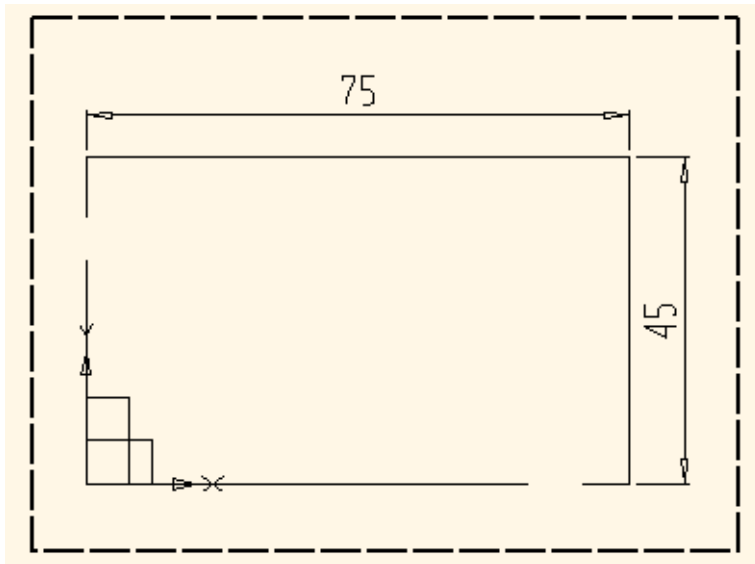

10. Now choose the tool . The rectangle is redrawn with the points moved to represent the desired geometry. The result is shown in Figure 44 below.

Figure 44 After Parameterization



Please note: Notice that every point has moved except for the fixed reference point. This transformation is only temporary. If you redraw the screen, the rectangle will be drawn in its original shape, but with the most recent set of dimension values. If the new shape is not quite what you want, you can change the dimension texts, and then try again.

11. When you are happy with the result you can make the transformation permanent using the tool .

Example 2: Using Baselines

In this example, you will parameterize the rectangle you created in the last example again. This time the reference point is fixed by a pair of intersecting static baselines rather than a prim. If an error message appears during this example, refer to “[Parameterization Errors](#)” on page 51.

1. Draw a rectangle with center lines.

Creating a reference point


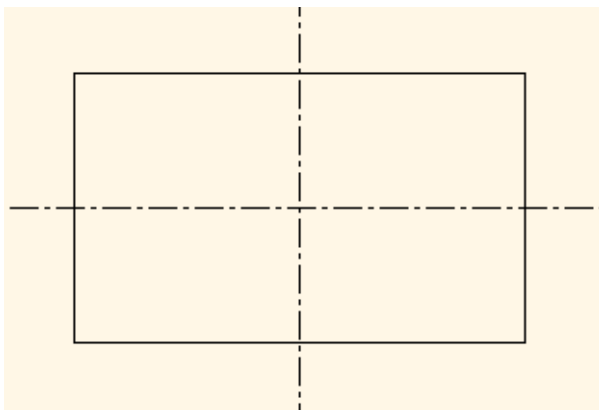
2. Add two baselines, line style *Static Base Line*, line type *LBL*, which intersect at the center of the rectangle by choosing the tool *Creates static base lines* . Use the end points of the center lines as points for the base lines.

Figure 45 Example2: Rectangle with Intersecting Baselines





The intersection of the baselines is the reference point. This does not coincide with a point on the object geometry. This position is used to calculate how far to move the other points during parameterization.



Please note: Notice that the ends of the baselines extend beyond the rectangle. This is important. Never draw a static baseline that ends on an existing line. If you do so, the line will be fixed in its present position and this may create problems when you try to parameterize the drawing.



3. Choose the tool *Parametric Layers*.
4. Switch on the option *Baseline* and *click left* on the button *Hide*.
5. Now you can delete the centerlines easily.
6. Switch on the baselines by *clicking left* on the button *View*.

Drawing the Viewbox

7. Create a viewbox using the tool  Creates parametric viewbox lines **OR**  Creates parametric viewboxes.

Displaying the Grid

8. Choose the tool  Draws the grid corresponding to the original drawing and undoes it .
You can see two STK type grid lines, one horizontal and one vertical. However, there are in fact six grid lines. Each baseline supports three grid lines: one along its length, and two at the ends of the line, perpendicular to it.

Please note: Notice that the grid lines are not permanent. If you refresh the screen by choosing the tool  Refresh graphics  from the toolbar the grid lines disappear.

Changing the grid tolerance

Grid lines are drawn using a tolerance which is automatically set to 0.1 mm. Since the grid lines at the ends of the static baselines are only 0.1 mm long it is very difficult to see them. You can see them more clearly by changing the grid tolerance.







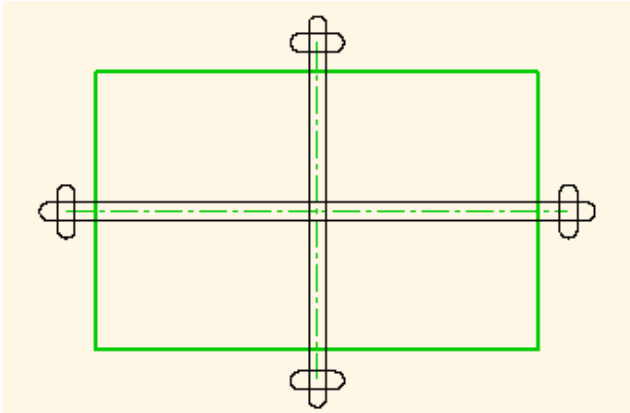
9. Open the dialog  Parametric Switches by choosing the tool  Displays the current settings of the parametrics system.
10. Increase the  Grid Tolerance by typing 3.
11. Choose the button  Apply.
12. Choose the tool  Draws the grid corresponding to the original drawing and undoes it .
The result is shown in Figure 46. You can clearly see six grid lines now.

Figure 46 Effect of Changing the Grid Tolerance



Please note: Normally, you should never change the grid tolerance setting. If you do, you may find that grid lines are being drawn through the wrong points, and that your drawing will not parameterize. Changing the grid tolerance is used here because it helps to illustrate how grid lines are built up. Do not forget to return the grid tolerance to its normal setting.

Dimensioning the Object

13. Now dimension the rectangle using chain dimensioning.

Please note: Note that it is essential to use chain dimensioning as other dimensioning methods such as datum dimensioning do not recognize the center support provided by the horizontal grid line (see "Dimensioning", "Center Support" on page 32).


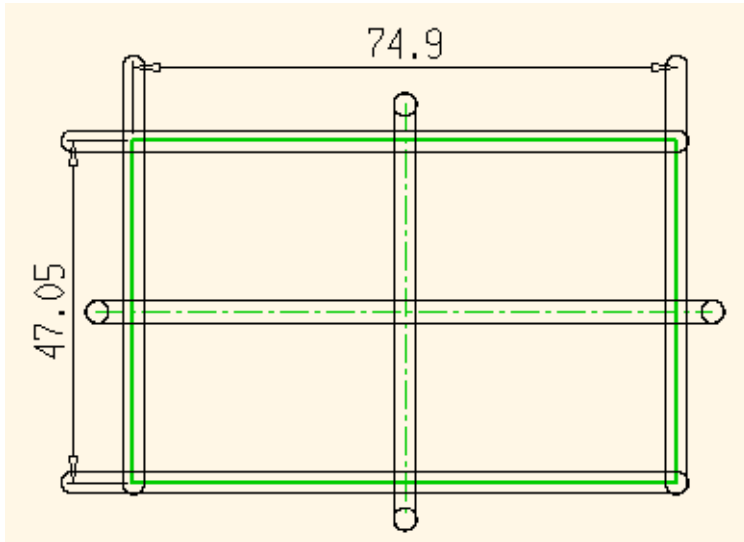


14. Choose the tool  to display the grid after every new dimensioning group. Each new dimensioning group creates new grid lines, which in turn support more grid lines.

Figure 47 Fully Dimensioned Drawing



Every point in the rectangle now lies at a grid line intersection. The next step is to test your drawing.

Testing the Drawing

15. Choose the tool  then immediately undoes .

There is a brief delay and then each part of the drawing is highlighted on the screen as it is processed. Do not proceed unless the drawing is tested successfully with no error messages. If an error message appears, you must diagnose the problem and correct it before proceeding. See [“Parameterization Errors” on page 51](#) for a description of some possible errors.

Inserting Your Own Values

When you have successfully tested the drawing:

16. Replace the dimension values with some of your own.

If you choose values that are too large, the rectangle may exceed the height or width of the maximum drawing area. If this happens, you will see a warning message.

Parameterizing the Object


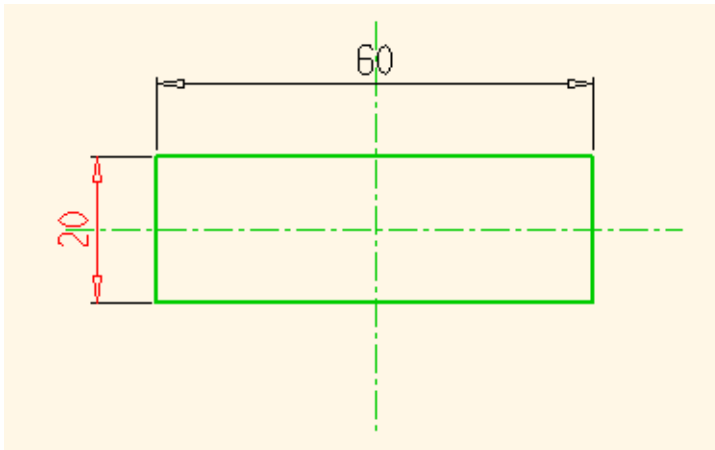

17. Choose the tool . The rectangle transforms according to the dimension values you have specified. Figure 48 shows the result.

Figure 48 Result of Parameterizing



Remember that the reference point is the center of the rectangle, where the baselines cross. Every point has been moved, but the center of the rectangle remains in the same position on the sheet.

18. At this stage there are two things you can do.
- Permanent parameterization
If you are happy with the result you can make the transformation permanent using the tool . This permanently transforms the geometry, and you can then save the sheet.
 - Restoring the original geometry
If you wish to continue parameterizing the rectangle, refresh the screen and the original rectangle will be restored. The dimension values are always the last set of values that you supplied.

SIMPLE PARAMETERIZATION EXAMPLES

This chapter contains three worked examples. Each example illustrates the use of a different kind of dimension, chain, radial and angular.

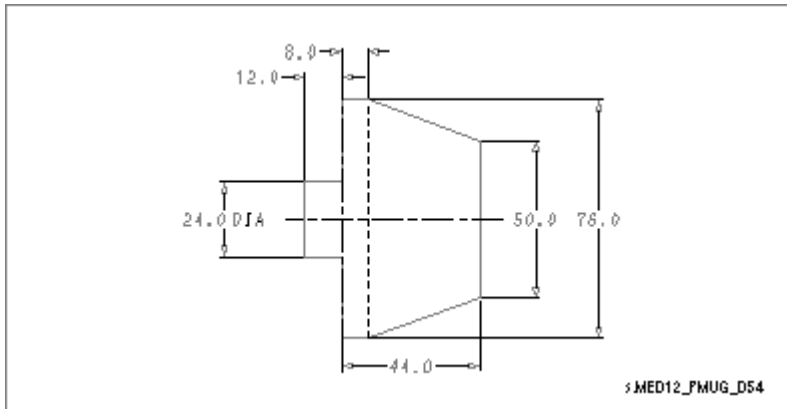
Before working through the examples you should read the section on ["Getting Started"](#), ["Examining the Parametric Grid"](#) on [page 48](#) explaining how to examine the grid. This introduces a very useful diagnostic routine you can use to find out what is wrong with a drawing that has produced an error message.

- [Example One](#) 64
- [Example Two](#) 68
- [Example Three](#)..... 71

Example One

Draw the component shown in Figure 49. Do not dimension the drawing yet. Draw the vertical dashed line using any suitable line type, such as L1 or L6.

Figure 49 Component to be Drawn





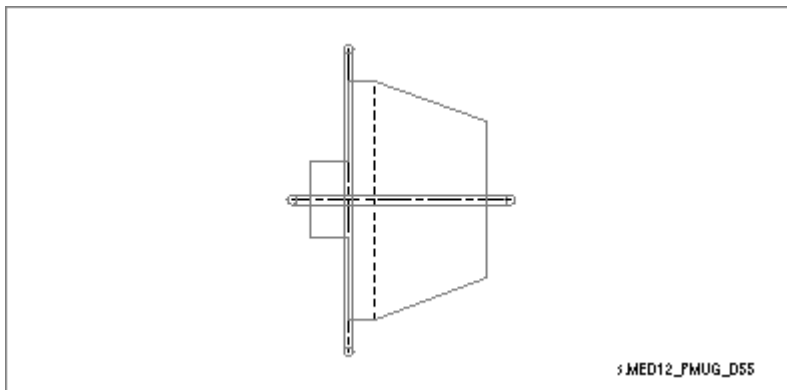



Add the intersecting baselines. Notice that the horizontal baseline lies along the line of symmetry. As you will see in a moment, this enables three chain dimensions to be supported. Use the tool  Draws the grid corresponding to the original drawing and undoes it  to look at the grid. At this stage, grid lines are drawn only along the baselines. You will build up the grid by dimensioning the points in the object.

Figure 50 Displaying the Grid



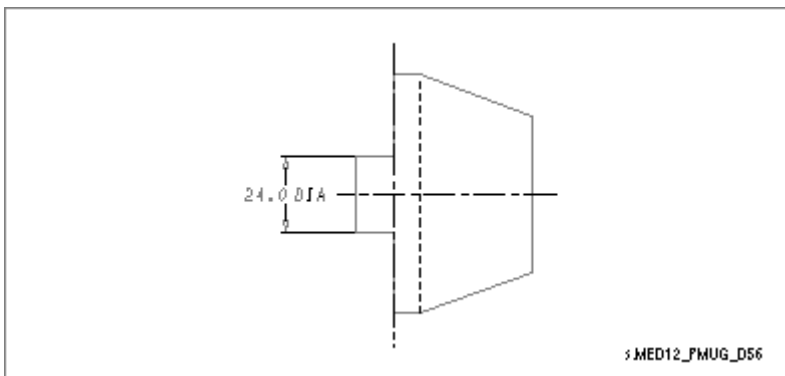
What would happen if you tried to parameterize the drawing at this stage? Try it for yourself. Refresh the screen using Refresh Graphics  from the toolbar and then select the tool Parameterizes the geometry then immediately undoes . You will find that the error message *Point not dimensioned* is written on the sheet beside every point that is not at a grid intersection. Remove all the error messages with Refresh Graphics .



Dimensioning the Object

Dimension the component as shown in Figure 49, “Component to be Drawn” on page 64, using chain dimensions for the diameters. Add the dimension groups in the order indicated below.

First stage: Add the diameter dimension shown in Figure 51. This is supported by the horizontal baseline, because the baseline passes through its center. It is essential that you use chain and not datum or coordinate dimensioning for the diameter dimensions so that they can pick up the center support from the horizontal baseline.

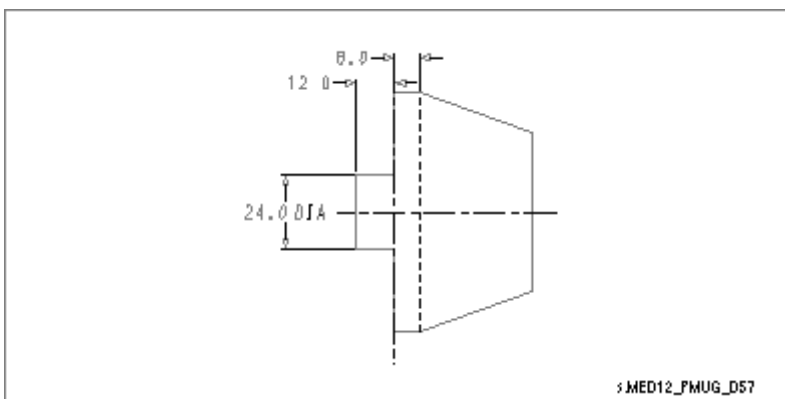
Figure 51 Addition of First Chain Dimension



Each time you add a new dimension, use the tool  Draws the grid corresponding to the original drawing and undoes it  to see how the grid is developing. At this stage, only two of the points lie at grid intersections when you draw the grid.

Second stage: Add the 8.0 and 12.0 mm dimensions, as in Figure 52. You can use any dimension type, for example datum. When you draw the grid you will see that grid lines are drawn along the leader lines of the dimensions and continue along other collinear lines.

Figure 52 Adding More Dimensions

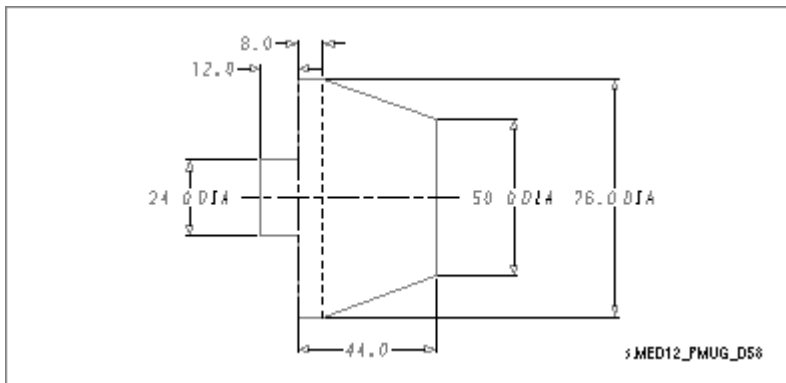


When you display the grid, you will see that the dashed line helps to extend the grid. The grid line drawn along the leader lines of the 8.0 mm dimension extends along the dashed line and

through any collinear points. The dashed line enables this grid line to extend to the collinear point on the other side of the component. Without the dashed line, you have to add a second 8.0 mm dimension.

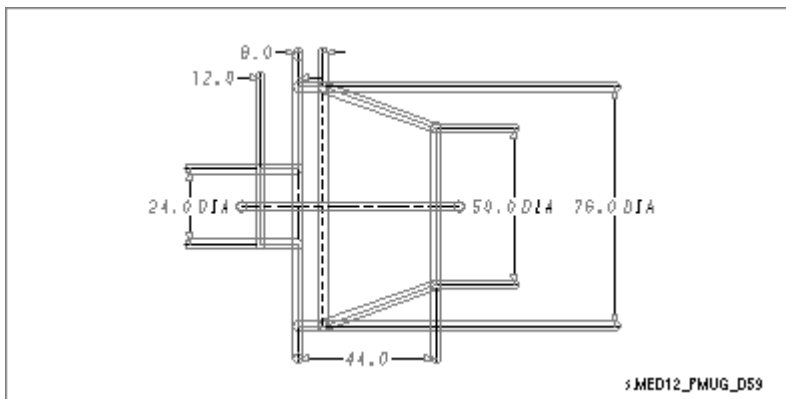
Third stage: Add the 44.0 mm chain dimension and then the two diameter dimensions, as shown in Figure 53. You can use any dimension type for the 44.0 mm dimension but you must use a chain dimension for the 50.0 mm and 76.0 mm diameters. This ensures that the horizontal baseline running through the center of the dimension supports the new dimensions on the grid.

Figure 53 Fully Dimensioned Component





When you draw the grid, as in Figure 54, you will see that grid lines have been added along the angled edges of the component. These grid lines have been added automatically by the system. The grid is now complete, and every point lies at a grid intersection.

Figure 54 Complete Parametric Grid




If you wish, you can now use the procedure on ["Getting Started"](#), ["Building the Parametric Grid"](#) on page 46 to draw the grid one step at a time.

Testing the drawing

Use the tool  then immediately undoes  to test the drawing. If there are no problems, each part of the drawing will be redrawn as it is processed. Error messages will be displayed at the offending points if the system cannot parameterize the drawing. You may see one of the following messages written on the sheet near to the point that is causing the problem:

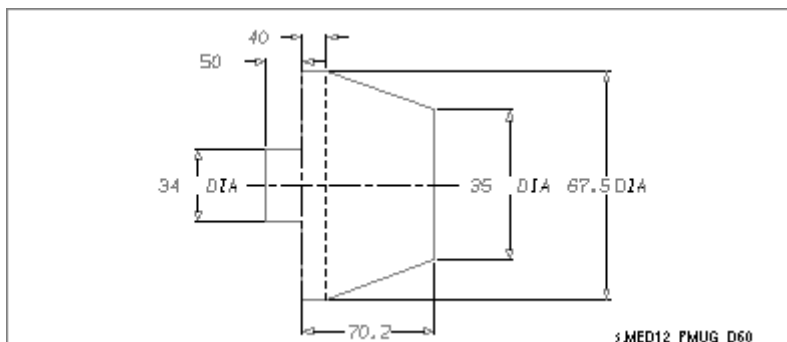
No solution possible
Point not dimensioned

If you see one of these error messages, use the tool  (see "Getting Started", "Parameterizing an Object" on page 50) to investigate which dimension is causing a problem.

Parameterizing the object

Replace the dimension texts with the new values shown below in Figure 55. Leave out the DIA text (or the diameter symbol) when you give a new value to the diameter dimensions: these extra texts are not required to calculate the new geometry.

Figure 55 Old Component With New Parameters






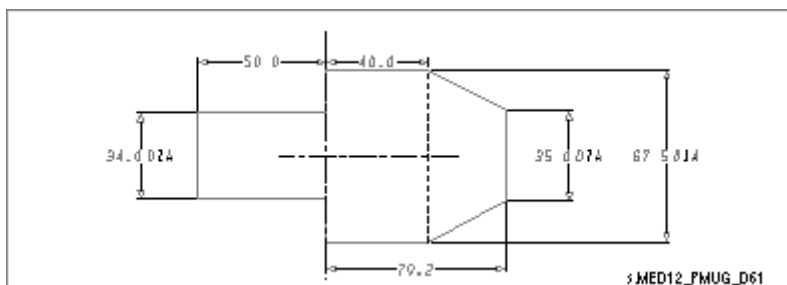
Now choose the tool  then immediately undoes . The result is shown in Figure 56. If you are happy with the new component, parameterize it permanently using the tool  and then save the sheet.

Figure 56 Parameterized Component



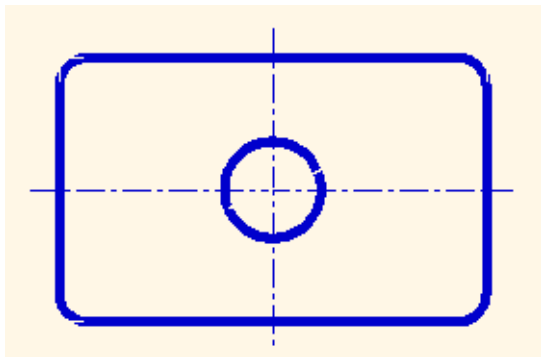
Example Two

This example shows how to deal with fillets for parameterization.


1. Draw the object in Figure 57 but do not dimension it yet.
Use the tool **Fillet Lines** from the **Lines + Edit** tooltray to create the fillets.

Please note: The fillets have to be changed to tangent point arcs (tool **Convert the selected circular arc into a tangent point arc** in the tooltray **Lines + Edit**) because they are circular arcs, which is the default for fillets.

Figure 57 Object to be Dimensioned

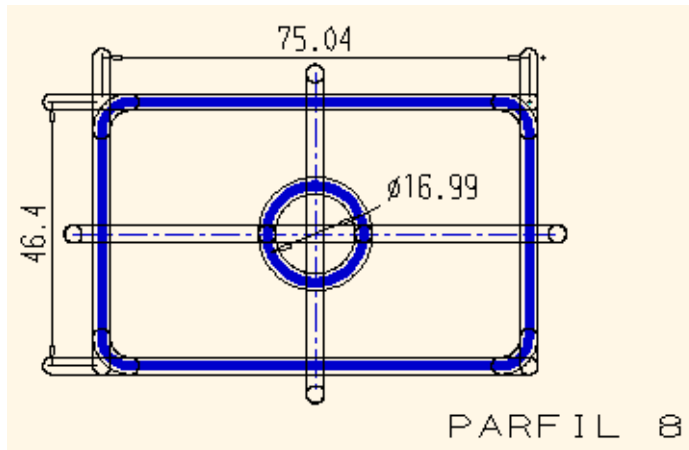


Dimensioning the Component

2. Add static baselines, which should intersect at the center of the object.
3. Dimension the component.
Use chain dimensions to ensure center support from the horizontal baseline.
For the circle use the tool **Dimensions the diameters of circles and holes**.
Instead of dimensioning the fillets individually, use an in-sheet **PARFIL** command to set fillets to 8.0 mm:
4. Choose the text tool **Creates free parametric command text**.
Below the dashboard the text edit field opens.
5. Type the text you want to place as command on the sheet:
e.g. `PARFIL 8`
6. Place the text on the sheet.
7. Choose the tool **Draws the grid corresponding to the original drawing and undoes it**  for looking at the grid.
Every point in the drawing now lies at a grid intersection (Figure 58).

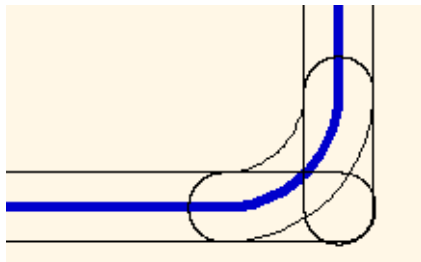
If you have any problems then refer to "Getting Started", "Example 1: Using a Prim" on page 53 or "Appendix B Error and Warning Messages" on page 217.

Figure 58 **Displaying the Grid**








Notice that small grid lines are drawn over the arcs of the fillets, as shown in the detailed illustration in Figure 59. The end points of the fillets lie at grid intersections because the ends of the arcs intersect with the straight grid lines.

Figure 59 **Detail Showing Fillet Grid Lines**



Parameterizing the Component

If you have tested the drawing with the tool  Parameterizes the geometry then immediately undoes  and there are no problems, you can go ahead and replace the dimension texts with new values.

8. Choose the tool  Sets options for fillet behavior during parameterization for opening the dialog Parametric Filleting (see "Dimensioning", "Dialog PAR VAR Setting" on page 30).
9. Set the options and values you want to use and Apply your settings.
10. Use again the tool  Parameterizes the geometry then immediately undoes  to parameterize the drawing.
You can continue parameterizing the component by refreshing the screen and replacing the dimension texts with new values.

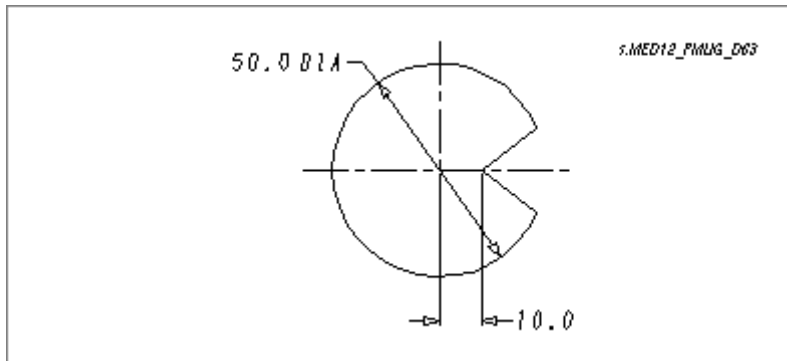
Dimension Format

The format of dimension text on the parameterized drawing is the same as in the original drawing. For example, the affixes j, R, and DIA are retained, and the original number of decimal places are displayed regardless of the format you gave when you supplied the new parameters.

Example Three

Draw the object shown in Figure 60. The angle of the notch is not important, but the notch must be symmetrical about the horizontal center baseline. In this case the angle is 75.5 degrees. If the notch was not symmetrical about the horizontal baseline, you would have to dimension it as two separate angles.

Figure 60 Object With Angular Notch



Adding dimensions


Add the 50.0 mm diameter dimension and the 10.0 mm chain dimension and then draw the grid as shown in Figure 61 using the tool . Draws the grid corresponding to the original drawing and undoes it.

Figure 61 Result of drawing the grid

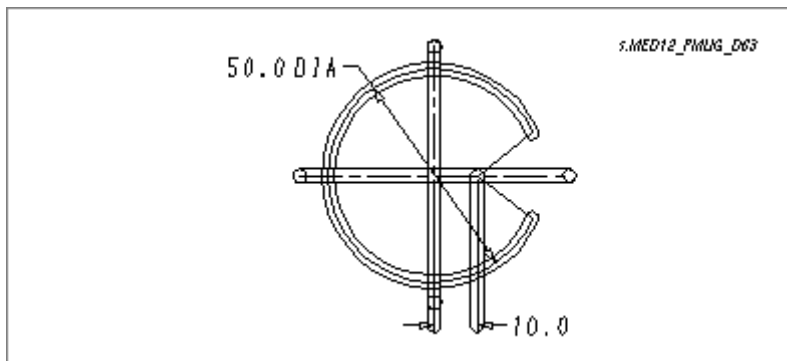
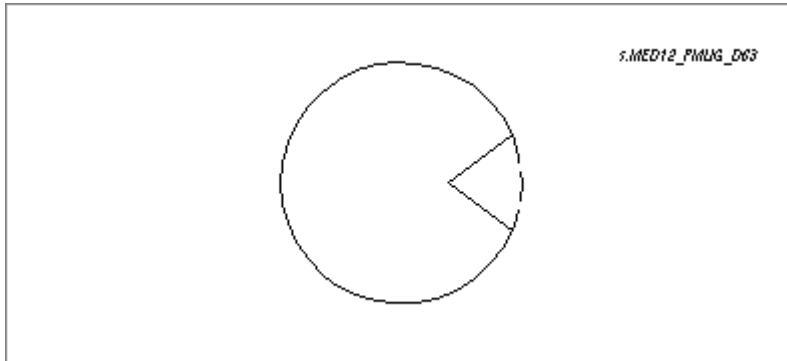


Figure 61 shows that the grid is still incomplete. All points must be at grid intersections. To make the circular grid line cross the gap in the object circumference caused by the angled notch, add a line as shown in Figure 62 across the notch (any line type will do).

Figure 62 Adding a Line



Dimensioning the Angle


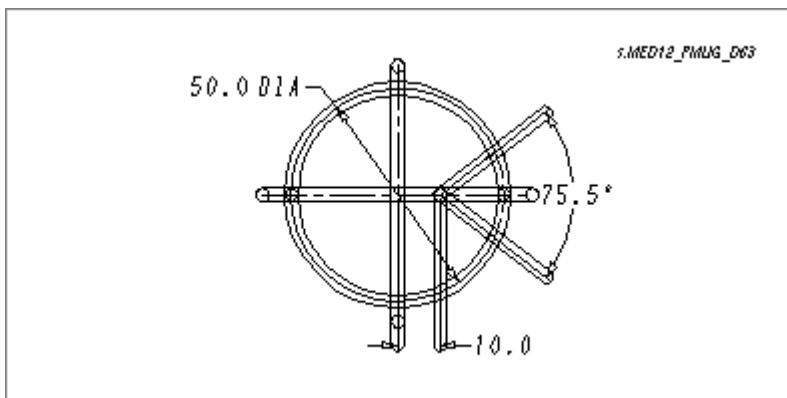


Draw the grid again and you will see that the grid line created by the diameter dimension now runs in an uninterrupted circle around the circumference of the object. Finally, place the angled sides of the notch on the grid by dimensioning the angle. When you display the grid with the tool , as in Figure 63, you will see that every point now lies at a grid intersection.

Figure 63 Dimensioned Angle With Grid



Testing the Drawing

Test the drawing with the tool . Parameterizes the geometry then immediately undoes . If there are no errors, you can now change any of the dimensions and parameterize the object. If you change the angle, the notch will be redrawn with the specified angle. As the original notch was symmetrical about the baseline, the new notch will also be symmetrical. Try it for yourself. When you

change the text of an angle dimension, it does not matter if you miss out the degree sign: Parametric Design does not need it, and it will replace it for you anyway in the new dimension.

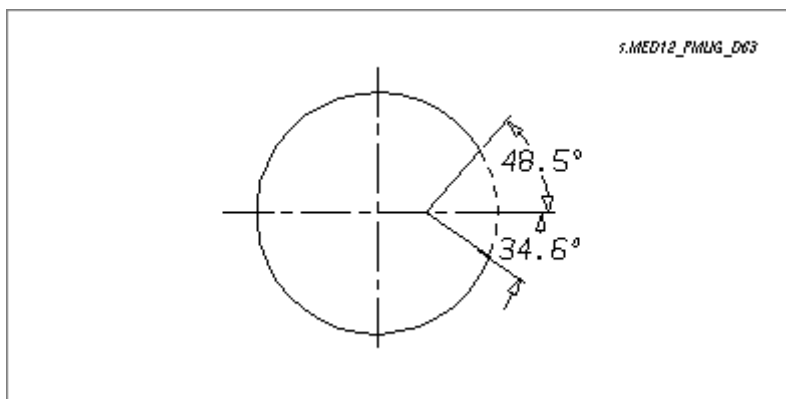
Examining the Grid

If you wish, draw the grid one step at a time to see how it was built up using the procedure in "Getting Started", "Examining the Parametric Grid" on page 48. Notice that the angled grid lines drawn on the notch are supported by the vertical grid line 10 mm from the circle center.

Dimensioning an Unsymmetrical Angle

If the notch was not symmetrical about the horizontal baseline, you would have to dimension it in a slightly different way, as shown in Figure 64. Here, the angle is treated as two separate angles, each dimensioned from the horizontal baseline.

Figure 64 **Dimensioning an Unsymmetrical Angle**



GEOMETRIC CONSTRAINTS

This chapter shows how certain properties of object geometry, such as tangency or perpendicularity, are preserved during parameterization.

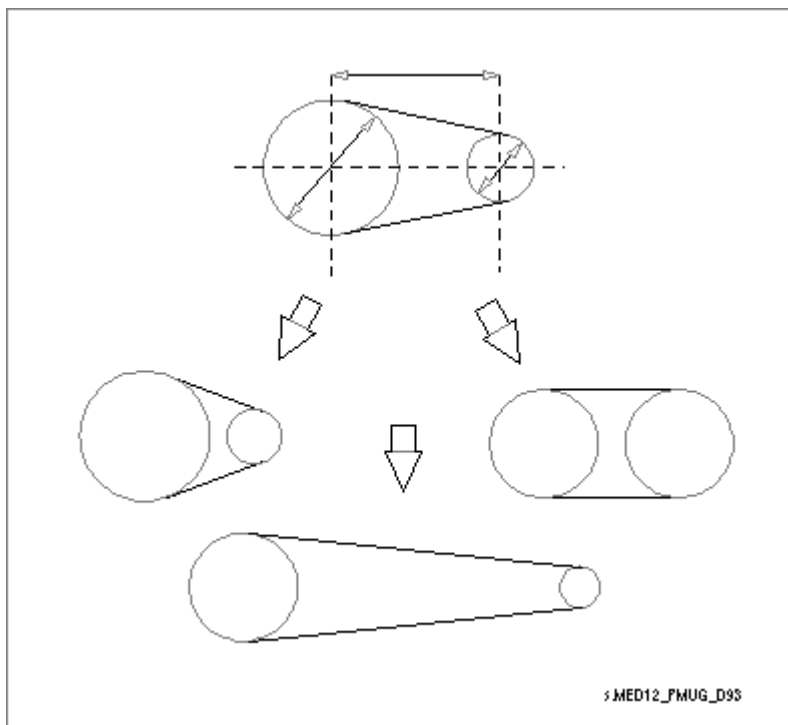
- What Are Geometric Constraints? 76
- Looking at Inferred Constraints 77
- Baselines 78
- Specifying Constraints Explicitly 80
- Example One 81
- Example Two 83
- Example Three..... 86

What Are Geometric Constraints?

Geometric constraints are automatically inferred by the Parametric Design system to ensure that certain properties of the original geometry are preserved when you parameterize a drawing. For example, a tangent line remains tangential and a perpendicular line remains perpendicular.


Figure 65 shows a simplified diagram of a bicycle chainset. The chainset consists of a large sprocket, a small sprocket and a chain that fits tightly around them. The chain is constructed from two separate line segments, each drawn tangential to the sprockets. To parameterize the drawing, you would begin by dimensioning the diameter of each sprocket and the distance between them. The chain is not dimensioned in any way.

Figure 65 Simplified Diagram of a Bicycle Chainset




You may want to change the size of the sprockets or the distance between them. The chain, of course, should always remain tangential to the sprockets. Parametric Design automatically ensures this whenever you parameterize the drawing. All you have to do is to dimension the drawing in the usual way.

Looking at Inferred Constraints

If geometric constraints are applied automatically by the system, how do you know what constraints will be applied? The answer is to look at the current geometric constraints either with the tool  or the PAR GRIS BAS Bacis1 command (which is explained in the *Parametrics Reference Guide*, chapter “Geometric constraints”, section “Automatically Inferred Constraints”).

Displaying Inferred, Dynamic Baselines

The tool  draws a dynamic baseline on top of any lines in the drawing where constraints have been automatically inferred from the geometry. These lines are chain lines of type LBL and are placed on layer 16. The point functions at the ends of the baselines indicate how the geometry is constrained when the drawing is parameterized.

The automatically inferred baselines are drawn permanently. For parameterization these lines can be kept (see “[Faster Parameterization by Dynamic Baselines](#)” on page 79) but you also can delete them, see below.

Deleting Inferred, Dynamic Baselines

Dynamic baselines are drawn only to help you to understand how the drawing is constrained during parameterization. You do not need to leave them on the sheet once you have found out where constraints have been inferred.

For deleting inferred baselines,






1. Open the dialog *Parametric Layers* (see “[Layers and Parametric Switches](#)”, “[Parametric Graphics Control](#)” on page 111).
2. Switch on the option *Inferred Baselines*.
3. *Click left* on the button *Delete*.
The inferred baselines are removed immediately.

Baselines

Every point in a baseline has a particular function that determines the way in which the line is constrained during parameterization.

Baseline Point Functions

The point functions shown below have special meaning in Parametric Design.

FUNV	Symbol	Point function
0		Static point
10		Perpendicular point
11		Intersection point
12		Tangent point
26		Circle center point

A baseline point with any of the point functions listed in the table above moves according to a particular geometric constraint during parameterization. For example, if the end of a baseline has point function FUNV11, that line segment remains fixed to a particular point or intersection throughout parameterization. If a baseline segment has point function 10, the line segment will remain perpendicular.

Static Baselines


A static baseline is a baseline with a FUNV0 point function at each end. A FUNV0 point does not move during parameterization. Static baselines remain in the same position on the sheet when you parameterize a drawing, regardless of how the object geometry changes.

Baselines and The Parametric Grid


Static baselines can be drawn anywhere within the viewbox and are used to support the parametric grid.

Dynamic baselines are constructed along existing lines and must be supported on the grid.

Faster Parameterization by Dynamic Baselines

Leaving any dynamic baselines created with the tool *Draws baselines inferred when the baseline switch is on*  in the master drawing can speed up parameterization. Parametric Design automatically looks to see where it can infer geometric constraints before parameterizing geometry. If you use dynamic baselines to specify explicitly any constraints that you need, you can then tell Parametric Design not to infer any constraints automatically. The system does not then spend any time trying to infer constraints from the geometry.

You use the PAR BAS switch to switch automatic inference of geometric constraints ON and OFF. If you want to leave baselines in the drawing, use the following procedure:

1. Create a master drawing and test it to make sure it can be parameterized.
2. Use the tool *Draws baselines inferred when the baseline switch is on*  to add dynamic baselines where constraints may be inferred.
3. Save the sheet.
4. Set the parametric switch PAR BAS to OFF.
You can do this interactively or using an in-sheet command.
5. Parameterize the drawing.

The PAR BAS Switch

The PAR BAS switch controls whether geometric constraints are inferred automatically or not. When PAR BAS is OFF, constraints are not automatically inferred, so you need to specify them explicitly. When PAR BAS is ON, constraints are inferred automatically from the object geometry.

You can change the current PAR BAS setting by giving an in-sheet PAR BAS ON or OFF command or in the appropriate dialog, see ["Layers and Parametric Switches"](#), ["Parametric Switches and Command Texts"](#) on page 112.

PAR BAS is ON by default. For more information on the PAR BAS switch refer to the *MEDUSA Parametric Design Reference Guide*.

Specifying Constraints Explicitly

Normally Parametric Design calculates for itself which geometric properties need to be maintained. When the drawing is parameterized, lines such as tangents and perpendiculars are adjusted automatically. Most of the time, therefore, geometric constraints are deduced for you.

If you want to impose additional limitations on geometry you can also add dynamic baselines to a drawing manually. This may be necessary,

- when insufficient baselines are inferred automatically
- when you want to constrain a component or mechanism in a particular way, for example to force one end to remain static with the other end linked to a point which moves.

To create a dynamic baseline:


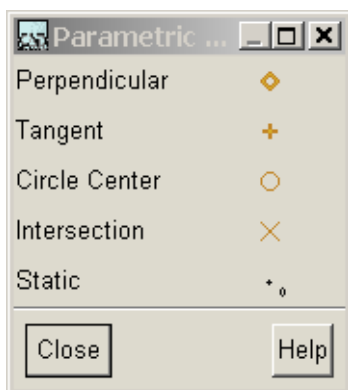
1. Choose the tool **Creates static baselines** .
The dialog **Parametric Point Function** opens. All entries are deactivated until you have placed the first point of the line.
2. Place the first point of the baseline.
The point functions in the dialog are activated.


Figure 66 Dialog **Parametric Point Function**, activated



3. Choose a point function for the first point of the line.
If you choose no point function, **Static**, FUNV 0 is used, which is the default.
Consider that the point function of the first point is not displayed on the sheet until you placed the second point of the baseline. For all further points the point function is drawn immediately.
4. Place the second point of the line.
5. Choose a point function for the second point of the line.
6. Go on like this for further points of the dynamic baseline.
7. If you want to finish the line choose **New line** or **Exit** tool from the popup menu.

If you create any dynamic baselines manually, leave them in the master drawing when you parameterize it.

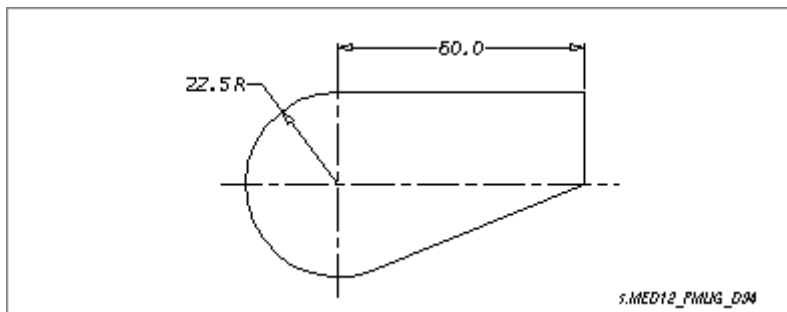
Example One

This example shows how you can use tool **Draws baselines inferred** when the baseline switch is on  to display constraints inferred by the system. These are constraints of intersection, perpendicularity, and tangency.

Drawing the Object

Draw and dimension the object in Figure 67. Make sure that the angled line segment and the horizontal line segment are both tangential to the arc.

Figure 67 Object to be Parameterized



Looking at Constraints




Now draw the parametric grid using the tool **Draws the grid corresponding to the original drawing** and undoes it . This example might look quite ordinary but something special has happened. Refresh the screen with **Refresh Graphics**  and then use the tool **Draws baselines inferred** when the baseline switch is on  to see which line segments are being constrained automatically.

Figure 68 Inferred Baselines

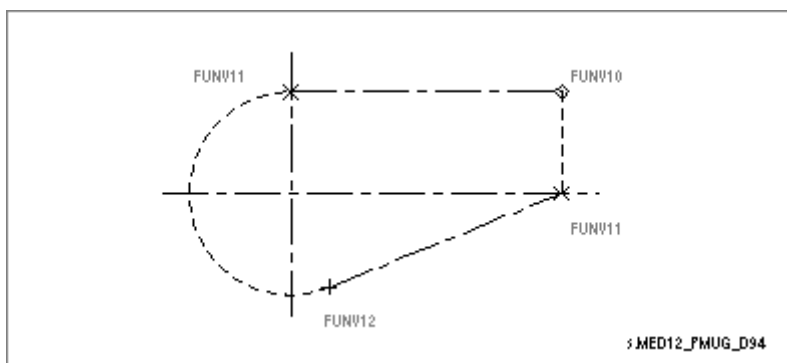


Figure 68 gives you an idea of what you will see. The outline of the component has been drawn using a dashed line so that you can see the dynamic baselines more clearly and the point functions have been labelled. Two LBL lines are drawn over the line segments that will be constrained automatically when you parameterize the component. The end of each line is marked with a special point function. The end with the FUNV12 point function is a tangent point, and the ends with FUNV11 point functions are intersection points. The end with a FUNV10 point function will remain perpendicular to the vertical edge of the component.

Taking a Closer Look

You can see the dynamic baselines and their point functions more clearly if you clear the screen and then draw just the layer 16 (Parametrics - automatically generated baselines) containing the baselines:

The important feature of this example is that you can change the arc radius or the horizontal dimension, but the angled straight segment will remain tangential to the arc, and the horizontal line will remain perpendicular to the intersection point. If you wish, you can now delete the dynamic baselines.

Parameterizing the Object



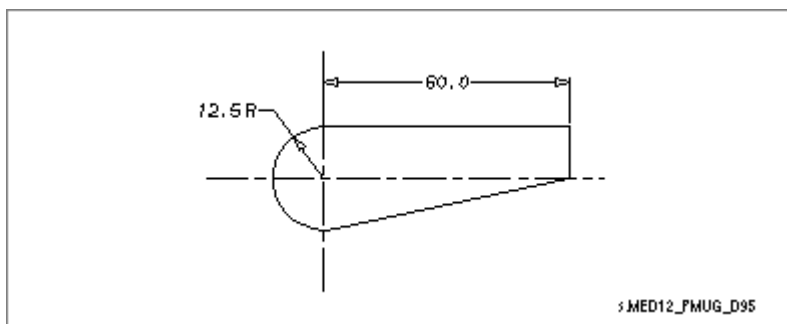
Test the drawing with the tool  Parameterizes the geometry then immediately undoes . If there are no problems, parameterize it with a new set of parameters. Figure 69 shows the drawing after it has been parameterized with the arc radius set to 12.5 mm.

Figure 69 Object with New Set of Dimensions



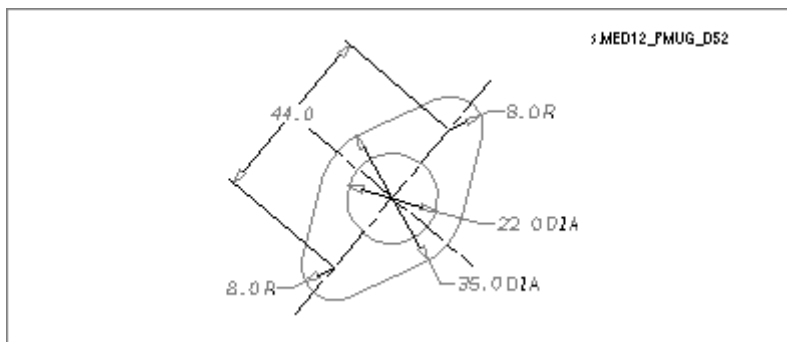
Example Two

This example shows how the system infers tangency constraints on a component.

Drawing the Object

Draw the component shown in Figure 70 in a horizontal position and then rotate it. Add static baselines but do not dimension the component yet. All straight segments are tangents.

Figure 70 Object to be Parameterized

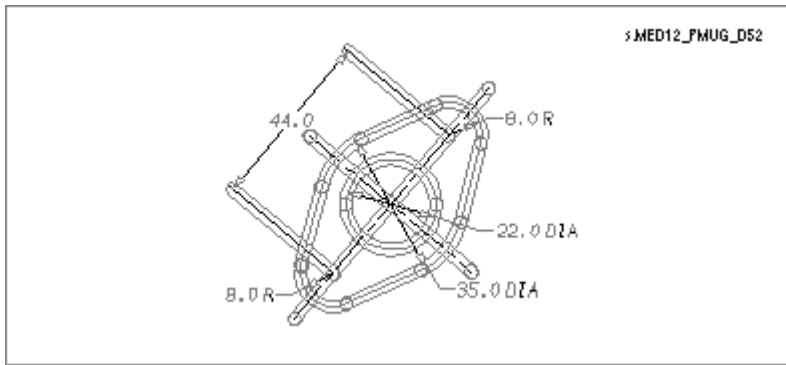


Dimensioning the Object

To dimension the object, first add the 44 mm chain dimension. This is symmetrical about one of the baselines. The center of each small arc now lies at a grid intersection. Add the two diameter dimensions and then dimension the arcs. When the drawing is fully dimensioned, draw the parametric grid. The result is shown in Figure 71.

If you look closely at the drawing you will see that four grid lines have been added automatically where there is no dimensioning. These are the four straight segments forming the outline of the component. Each straight segment is tangential to an arc at both ends.

Figure 71 Displaying the grid



Looking at Constraints


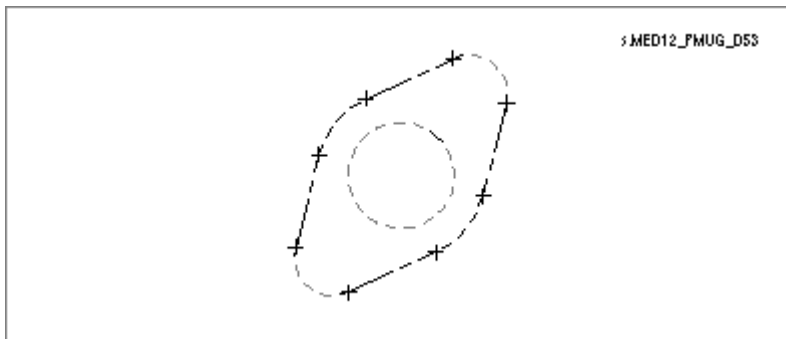
When you run the tool *Draws baselines inferred* when the baseline switch is on , a dynamic baseline is drawn over each part of the drawing that is to be constrained automatically by the system. The added lines are shown in Figure 72. The dimensioning and static baselines have been removed here and the outline of the component is shown as a dashed line so that you can see the dynamic baselines more clearly.

Figure 72 Inferred Baselines




You can see four LBL lines have been drawn over those line segments constrained to remain tangential during parameterization. The end of each line has a FUNV12 point function. When this object is parameterized, the four line segments indicated by the dynamic baselines will always remain tangential to the arcs.

Taking a Closer Look

You can see the dynamic baselines and their point functions more clearly if you clear the screen and then draw just the layer 16 (Parametrics - automatically generated baselines) containing the baselines:

Parameterizing the Object

Use the tool Parameterizes the geometry then immediately undoes  to test the drawing then change some of the dimensions and parameterize the object. Note how the straight line segments always remain tangential to the arcs.

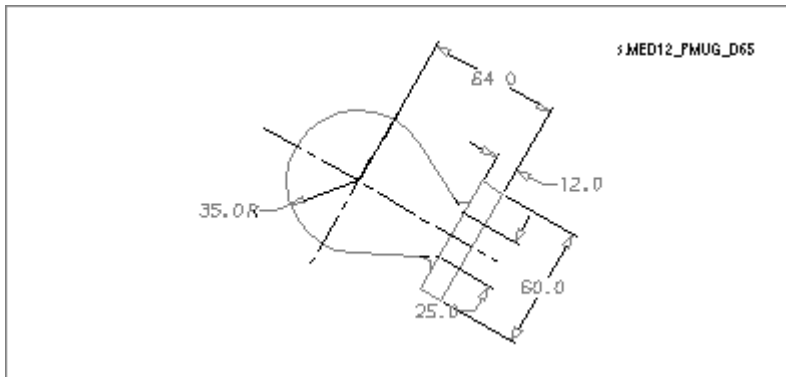
Example Three

This example shows how the system infers intersection and tangency constraints on a component.

Drawing the Object

Draw the component in Figure 73 in a horizontal position using the hints given below. Add the static baselines but not the dimensions, and then rotate the component through an arbitrary angle to avoid restrictions imposed by special cases.

Figure 73 Object to be Parameterized



Add two supporting lines to place the fillet arcs on the grid, as shown in Figure 74. These can be any line type. When you have done this, draw the grid, as shown in Figure 75.

Figure 74 Detail of Fillet Construction from Component in Figure 73

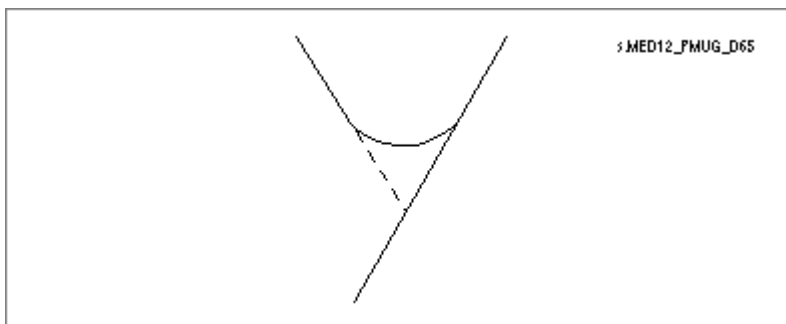
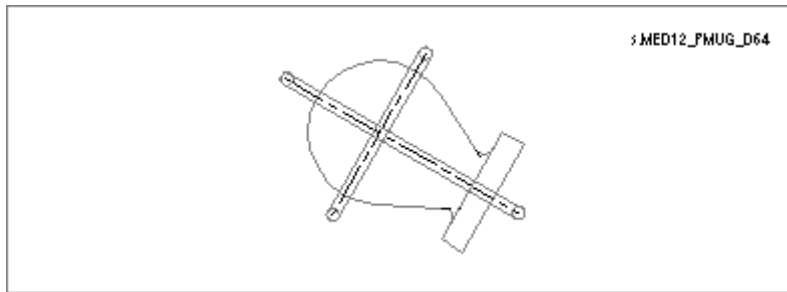


Figure 75 **Displaying the grid**



Dimensioning the Component

First step: Add the 35.0 mm radial dimension then draw the grid. Notice how grid lines extend along the arc to the tangent points. Next dimension the distance between the reference point and the intersection point between the baseline and the base of the component (64.0 mm). Add the two parallel chain dimensions across the base of the component. Notice that the 25.0 mm dimension places the fillet tangent points on the grid.

Second step: Next add the 12.0 mm parallel chain dimension. This creates three extra grid lines, not just one. These extra grid lines linking the fillet tangent points to tangent points on the large arc have been inferred automatically.

Adding an in-sheet Command

Finally, place an in-sheet PAR FIL command inside the viewbox to set the fillet dimension values to 5 mm. The grid for the fully dimensioned component is shown in Figure 76 and a more detailed view of the fillets is shown in Figure 77.

Figure 76 **Fully Dimensioned Drawing With Grid**

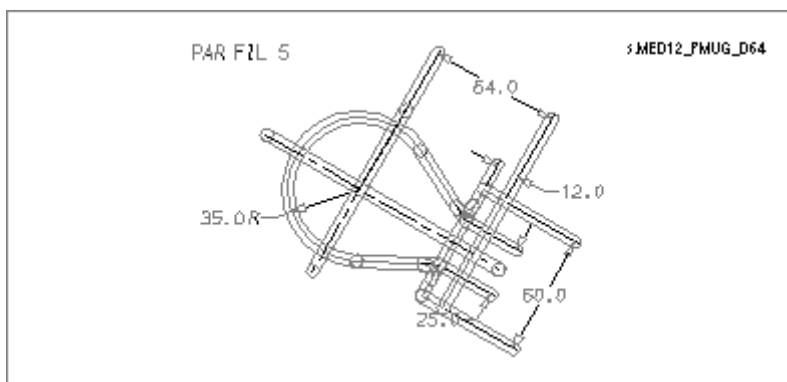
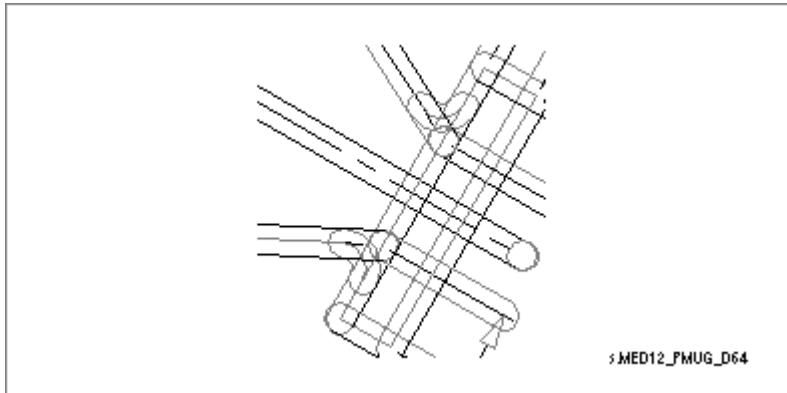


Figure 77 Fillet Detail from Figure 76



Looking at Constraints


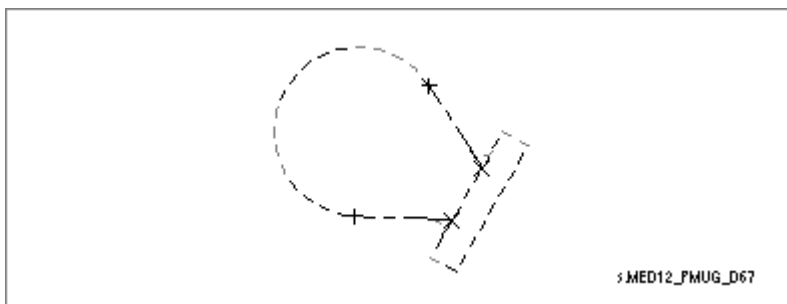
Now you can use the tool **Draws baselines inferred** when the baseline switch is on  to see what inferences have been made from the object geometry. Figure 78 shows where dynamic baselines are drawn. In this example, two baselines are added. Both have an intersection point function (FUNV11) at one end and a tangent point function (FUNV12) at the other. When you parameterize the drawing, these line segments may move, but the FUNV12 points will remain tangential to the large arc, and the FUNV11 points will be tied to intersection point on the base of the component.

Figure 78 Dynamic Baselines



Taking a Closer Look

You can see the dynamic baselines and their point functions more clearly if you clear the screen and then draw just the layer 16 (Parametrics - automatically generated baselines) containing the baselines:

Parameterizing the Object


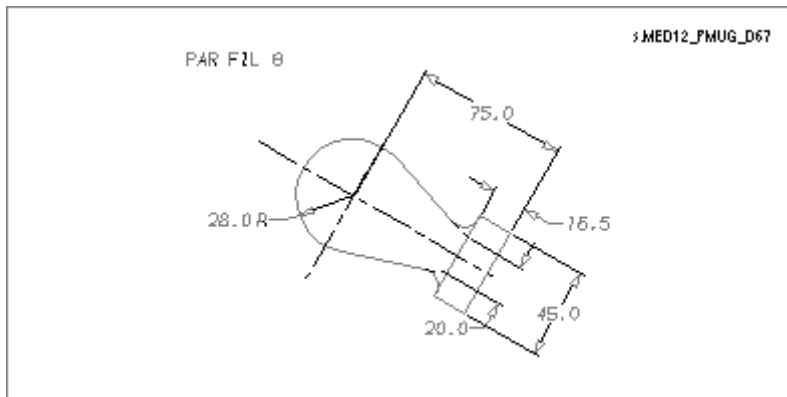
Delete the dynamic baselines if you wish, then test the drawing with the tool *Parameterizes the geometry* then immediately undoes . Now experiment by changing the dimensions and parameterizing the object. Figure 79 shows one possible result.

Figure 79 After Parameterization



VARIABLES AND EXPRESSIONS

This chapter describes how to replace values in dimensions with variable parameters. MEDUSA Parametric Design calculates the value of any variables during parameterization and changes the object dimensions to reflect the new values.

- Values in Dimensions..... 92
- Variables 94
- Assigning Values to Variables Using in-sheet commands 96
- Assigning Values to LCIS Variables 98
- Administration, Loading and Updating Variables 100
- Errors While Using Variables 104
- Expressions 105

Values in Dimensions

So far, whenever you have prepared a drawing for a parametric operation, you have changed its dimensions from one set of numbers to another. One of the most useful features of Parametric Design is that you can use variables such as WIDTH or DIAM and expressions instead of particular values.

Variables

Figure 80 shows a drawing of a joist section. Using variables to define the dimensions enables you to scale the section to whatever size you require. Each time you parameterize the drawing, you can give the variables a different value.

Figure 80 Use of Variables in Joist Definition

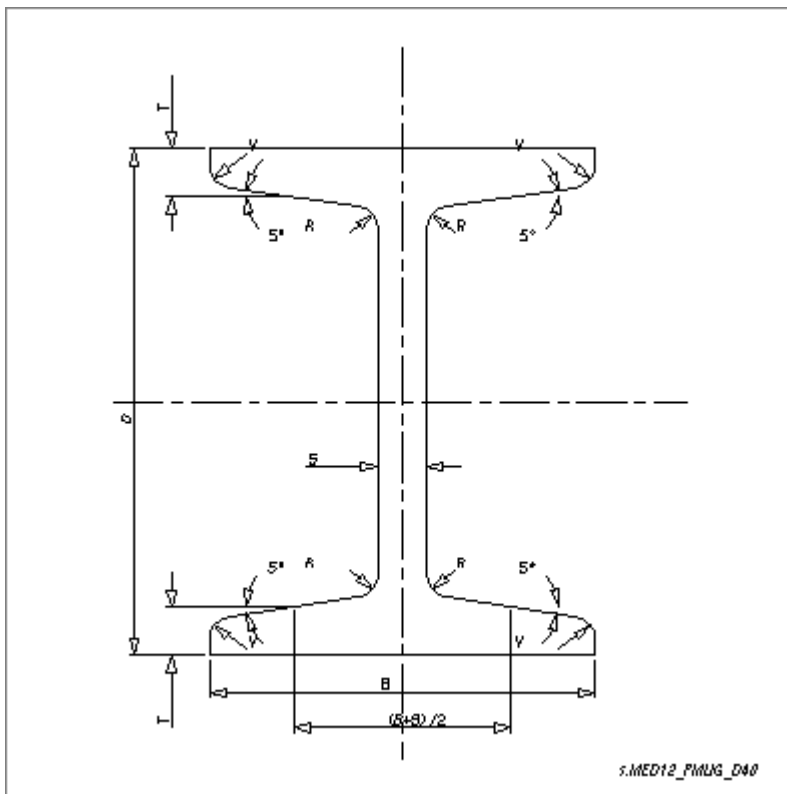
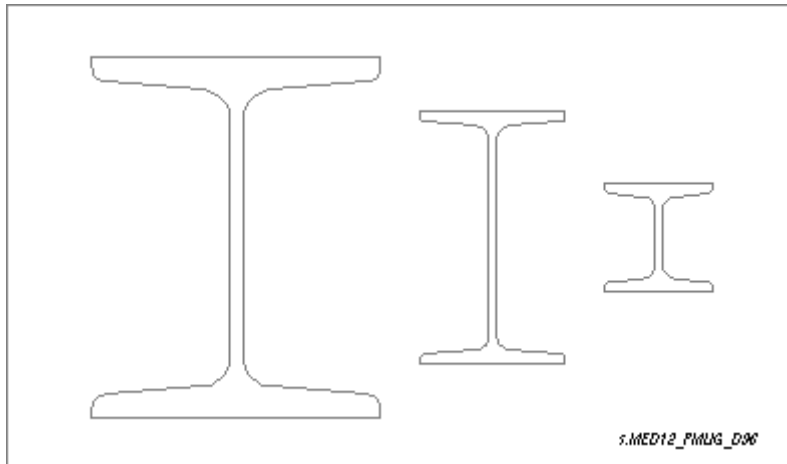


Figure 81 shows some joist sections which are the result of parameterizing the drawing shown in Figure 80. A LET command was used to give different values to the following variables each time:

- D (section depth)
- B (section width)

- S (web thickness)
- T (flange thickness)
- R (root radius)
- V (toe radius)

Figure 81 Result of Changing Joist Dimension Variables



Expressions

Expressions are particularly useful when you want to scale a component. Using expressions you can specify relationships between dimensions within the same drawing. For example, if the width of a component always needs to be half of the length, you could replace the length dimension value with the variable `LENGTH` and the width dimension value with the expression `(LENGTH/2)`.

In [Figure 80, “Use of Variables in Joist Definition” on page 92](#), the expression $(B + S) / 2$ is used to locate the mid-flange position. Whenever this drawing is parameterized, the mid-flange dimension value is derived by adding the current values for the width of the section (B) and the web thickness (S) and dividing the result by two.

Variables

When you have dimensioned your drawing in the usual way and checked that the drawing can be parameterized you can replace any of the dimension text with variable names.

Replacing Dimension Texts

It is not necessary to change all dimensions to variables. You can leave some as numerical values. For example, if a number of components differ only in diameter, change the dimension text for the diameter to the variable name DIAM, and leave all other dimensions as numbers.


Rules for Creating Variable Names

A variable name must meet the following requirements:

- The maximum length is 6 characters.
- The first character must be a letter of the alphabet.
- The name must consist of the letters A- Z, the numbers 0-9 or the characters comma (,) # or %. Spaces are not allowed.
- Letters can be uppercase or lowercase characters. FILLET is identical to `fillet` and `Fillet`.

Some typical variable names are: A, B1, width, LIN6, dia, fillet, RAD.

After Parameterization

After parameterization all parametric dimension variable names and expressions are evaluated and replaced by their numeric values. By selecting the Refresh Graphics tool , you can restore the original shape of the drawing and dimension variable names and expressions are also restored. If you do not cancel the parameterization, these variables and expressions are permanently changed into values.

Defining Radius Dimensions

If you use a variable to define a radius dimension, do not use variable names that begin with the letter R and are followed by a number, for example, R15. If you do, Parametric Design assumes this is a radial dimension with prefix text and interprets these ambiguous texts as values during parameterization. When this happens, the following warning message is displayed on the sheet:

Warning - ambiguous dimension texts interpreted as values

Example


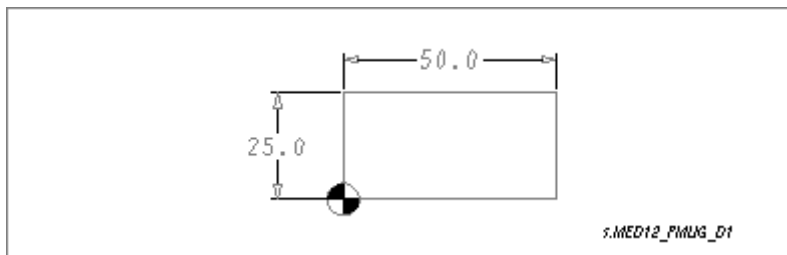
Draw and dimension a rectangle, as in Figure 82. The size of the rectangle is not important. Make sure the drawing can be parameterized by testing it with the tool `Parameterizes the geometry` then immediately undoes .

Figure 82 **Dimensioned Drawing**

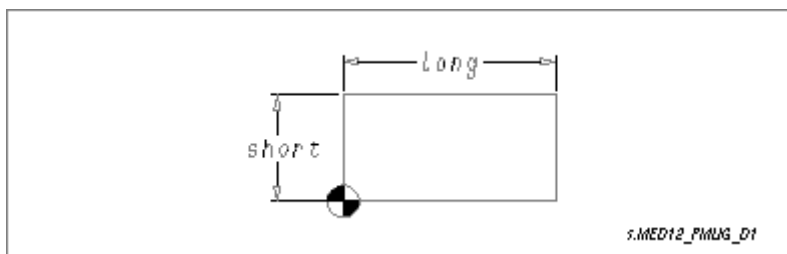


Now replace the dimension texts with variables:

1. Make the dimension text current.
2. Place the variable name into the text buffer, for example, `short`.

In Figure 83 the dimension texts have been replaced by the variable names `long` and `short`.

Figure 83 **Replacing Dimension Texts With Variables**



Before you can parameterize this drawing, you must assign values to the variables `long` and `short`. This is described in the next section.

Assigning Values to Variables Using in-sheet commands

There are two ways of giving MEDUSA commands:

- By typing the command at the keyboard (interactively in Bacis1 mode)
- By placing in-sheet command text on the sheet

You can use many Parametric Design commands both interactively and as in-sheet commands. When you enter a command interactively, the command is executed immediately. In-sheet commands are not executed until you run a parameterization tool.

Why Use in-sheet Commands?

The advantage of using in-sheet commands is that they are stored with the sheet, so that, for example, special parametric switch or layer settings are always used with a particular drawing. In-sheet commands are executed when you run a parameterization tool, before the geometry is parameterized.

In-sheet commands only affect the viewbox in which they are placed, so other viewboxes and other sheets are not affected. After parameterization, all settings are restored to the values held before parameterization.

You should always aim to use in-sheet commands rather than interactive commands when working with Parametric Design.

Creating in-sheet Commands

Use the following procedure to create an in-sheet command:


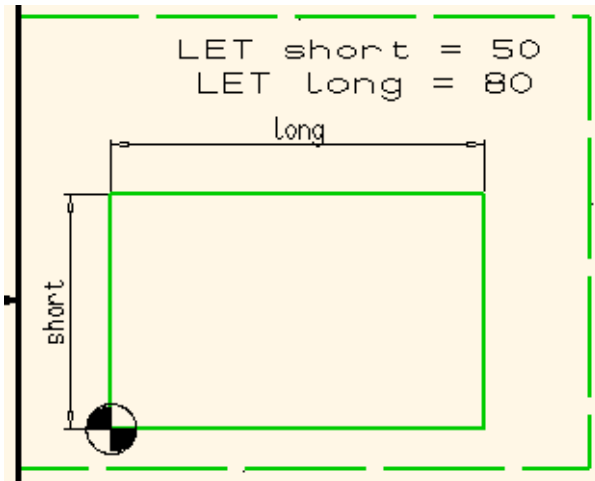


1. Choose the tool  Creates free parametric command text
Below the dashboard the text edit field opens.
2. Type the text you want to place as command on the sheet, for example:
`LET long = 80`
3. Position the new text by probing on the sheet.
4. Type further texts and place them.

Figure 84 below shows how in-sheet commands LET and DEF are used to assign values to the dimension variables `long` and `short`.

Figure 84 Example of In-sheet Commands



When you parameterize the drawing using the tool  Parameterizes the geometry then immediately undoes , the rectangle is redrawn using the dimensions you have specified in the in-sheet command texts.

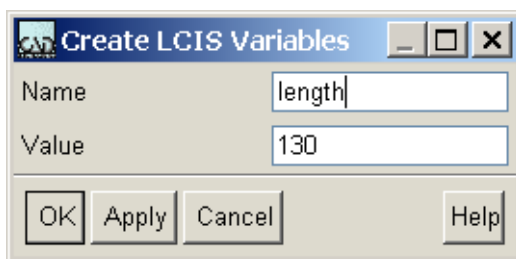
Assigning Values to LCIS Variables

Please note: LCIS variables are available for the current MEDUSA session only. In the next session you have to define them new. For saving LCIS variables use the Basis1 mode.

Creating a LCIS Variable

1. For creating a LCIS a variable choose the tool `Creates user LCIS variables`. The following dialog opens.

Figure 85 Dialog for Creating LCIS Variables



2. Type in the `Name` and the `Value` for the variable.

Please note: `Value` can be either an integer or a floating point value. Expressions are not allowed.

3. Apply the variable.
The dialog remains open for defining further variables.
4. If you have defined your last variable, choose `OK` to apply the variable and close the dialog.

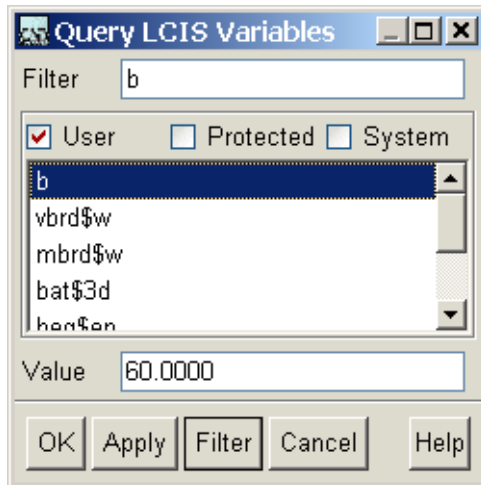
Query and Modify LCIS Variables

You can query any LCIS variable. Additionally to the variables you defined, many system variables can be queried.

Please note: Changing system variables is not recommended.

For querying or modifying LCIS variables choose the tool `Query and modify LCIS variables`. The following dialog opens.

Figure 86 Dialog for Query LCIS Variables



Filter

allows to pre-define the variables which are listed. You can type in any letter, number or special character available on your keyboard. If a variable name contains the **Filter** string than it is listed after choosing the button **Filter**. Wildcards like question mark (?) or asterisk (*) are not allowed. Same is with carriage return and space.

User, Protected, System

allows to pre-define the variables which are listed. You can specify user defined, protected and system variables.

Value

On the right hand side of **Value** there is the edit field displaying the value for the currently selected variable.

OK, Apply

confirms the current values for the variables and now they are used in MEDUSA. Additionally **OK** closes the dialog. If you choose **Apply** the dialog remains open.

Filter

This button updates the display of the variable list according to the defined pre-definitions.

Administration, Loading and Updating Variables

Variables used inside a parametric sheet and its values can be saved with the sheet. For this two tools are provided:

- Administrating parametric variables scans the sheet for existing variables and values. Administration includes editing variables, defining title, picture and macro and saving all this information within the sheet (see "Administration").
- Loading and updating variables allows to change variable values and save them within the sheet (see "Loading and Updating" on page 102).

Please note: The administration of variables can be done in Administration Mode only.

Administration


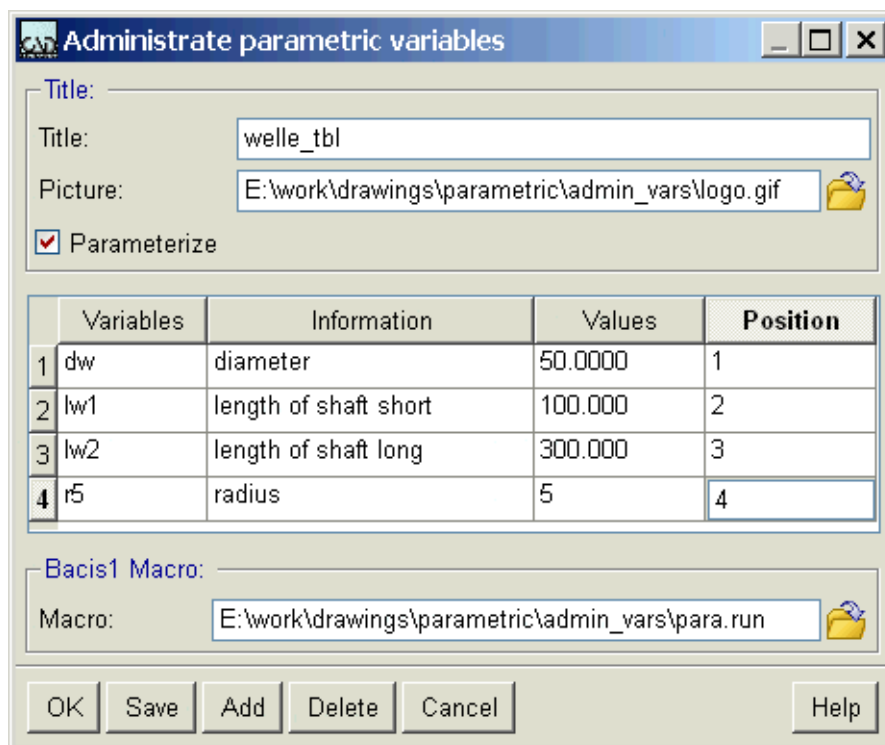
The tool **Administrate parametric variables**  reads all the dimension variables available inside the viewboxes of the current sheet. If there are none, an error message is given. Also the associated default values for the variables are read. If default values cannot be found, the appropriate variable will be initialized with the standard value. All information is displayed inside a dialog.

Figure 87 Dialog Administrate parametric variables



The Administrate parametric variables dialog contains the following **parameters**:

- Title is the name of the variable table and it may not be empty because it is used for identification when using the Load and Update dialog, see [“Loading and Updating” on page 102](#). If the title is not defined the Load and Update dialog will not work.
- Picture gives path and file name (*.gif) of the picture, which will be displayed in the Load and Update dialog.
- Parameterize
If this option is switched on, pressing the button OK parameterizes the drawing.
- Macro gives path and file name of the macro, which will be started after the parametrization using the OK button.

The **table** contains the following columns:

- Variables contains the names of the variables (this column is not editable).
- Information
contains the describing text for the variable.
- Values contains the value of the variable.
- Position contains the position of the variable in the dialog. The column Position is empty by default and the Administrator has to define positions. If no position is defined, the succession of variables is random.

The following **buttons** are provided:


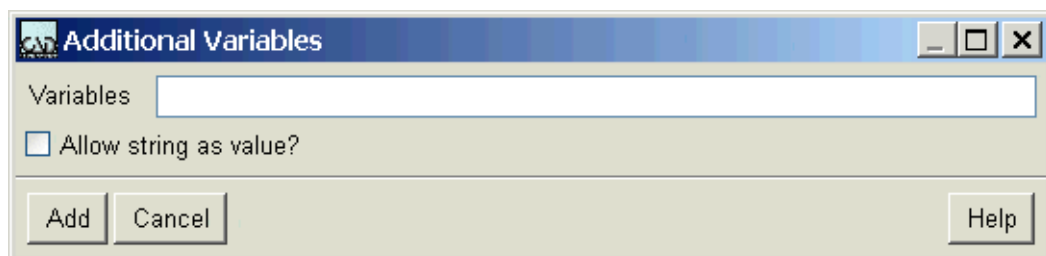
- OK saves all variables with its information, values and positions into the sheet, parameterizes the sheet (if the option Parameterize is on) and starts the macro (if defined in Macro text field). The Administrate parametric variables dialog will be closed.
If the option Parameterize is on, choose Undo  from the toolbar, if you want to test other settings. Consider that after parameterization you cannot open the Administrate parametric variables dialog until you have undone parameterization.
If you defined a Macro you have to press Undo twice.
- Save saves all variables with its information, values and positions into the sheet. Consider that this button does not save the sheet. If you close the sheet without saving it, the settings will be lost.
- Add adds a new parametric variable. It is used for variables, which could not be read from the sheet.


Figure 88 Add new parametric variable



After defining the variable name and setting the option (if required) press Add. The value will be defined directly inside the list of variables.

If the option `Allow string as value?` was switched on, you can define expressions like this: `length/2+val2`, otherwise you can define integer or real values only.

- Delete delete a new parametric variable. Variables read from the sheet cannot be deleted.
- Cancel close the dialog without applying any dialog entries. Change you have made will be lost.

Having made all settings for the parametric variables of a sheet, choose `Save the current sheet`  in the toolbar to save the table with the sheet.

Loading and Updating

Any user can display and update the values of parametric variables saved within a sheet. Updated values can also be saved within the sheet.


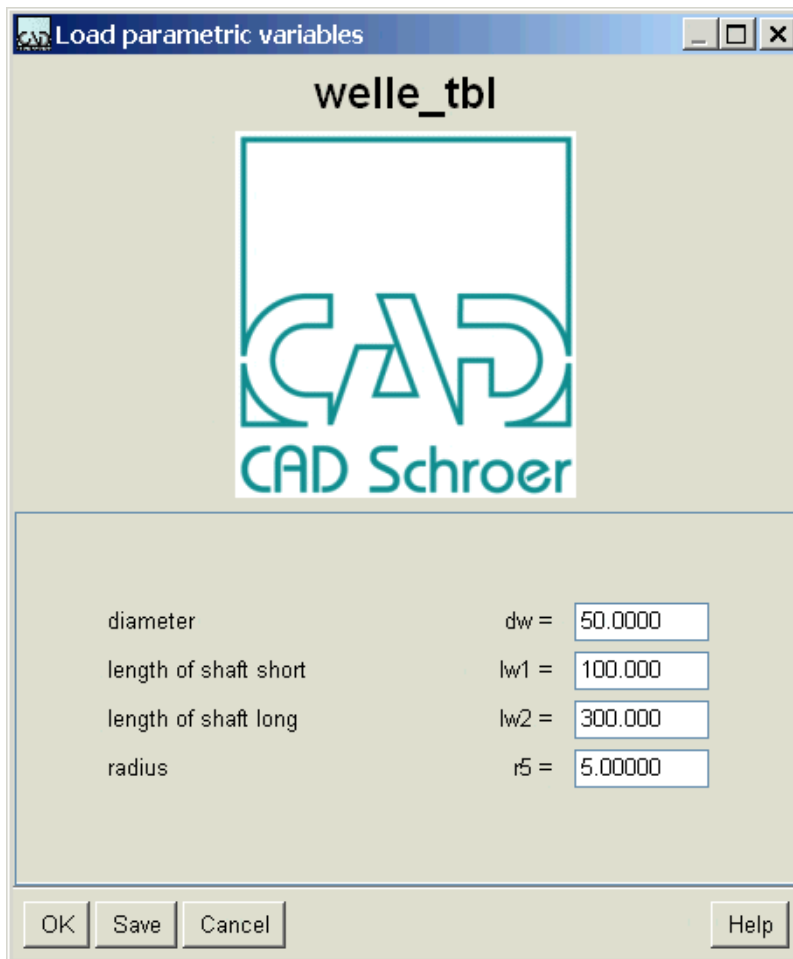
Choose the tool `Load and update parametric variables`  to open the following dialog.


Figure 89 Dialog Load and Update parametric variables



On top of the dialog the title and the picture (if defined) are given as defined by the Administrator. Below this the list of variables is given with its information labels and values as saved within the sheet.

You can edit variable values and save them in the sheet using the `Save` button.

For parameterization use the `OK` button. Consider that parameterization includes running the macro, if defined by the Administrator. The `Load parametric variables` dialog will be closed.

Choose `Undo`  from the toolbar, if you want to test other settings. Consider that after parameterization you cannot open the `Load parametric variables` dialog until you have undone parameterization. If also a macro has been run you have to press `Undo` twice.

Errors While Using Variables

An error message appears if you try to use a variable name without first having given it a value. For example, if your drawing contains the variable WIDTH and you have not executed a command to set its value, such as LET WIDTH = 170, then the following error message is written close to the appropriate dimension line when you try to parameterize the drawing:

```
Undefined variable
```

Expressions

You can also turn dimension values into expressions. Instead of replacing dimension texts with variable names such as DIAM or LENGTH, replace them with expressions containing variables, for example:

```
A+B*2+27
(3.14*(RADIUS**2))
```

You must give the variable names A, B, and RADIUS values using one of the methods described on “[Rules for Creating Variable Names](#)” on page 94. Expressions are particularly useful for specifying relationships between different dimensions in the same drawing.

The DEF Command

You create expressions using either LET or DEF. Use DEF to define an unevaluated expression. DEF is useful for defining expressions containing variables that are constantly changing. For example, the expression (WIDTH/LENGTH)**2 contains variables which can change each time the drawing is parameterized. To obtain the correct value, this expression must be evaluated each time the object is parameterized, using the current values of the variables WIDTH and LENGTH. Use single or double quotation marks to enclose an expression, for example:

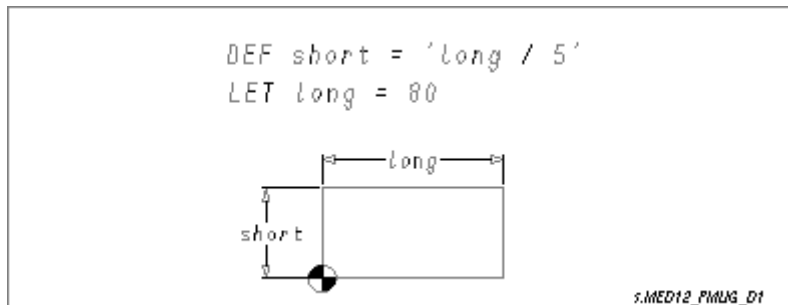
```
DEF WIDTH = 'LENGTH/2'
```

As with LET, you can use DEF both interactively and as an in-sheet command. Refer to the *MEDUSA Basis1 Design Commands Guide* for more information about LET and DEF.

Example

Using the drawing from the last example, replace the first LET command with a DEF command that defines `short` as an expression, as shown in Figure 90. This command sets the value of the variable `short` to one fifth of the value of the variable `long`.

Figure 90 In-sheet DEF Command



Parameterize your drawing using different values for `long` and `short`. You will see that replacing dimension text with expressions provides a useful mechanism for defining relationships between different dimensions.

Arithmetic Operators

You can use the following arithmetic operators in expressions:

- + Add
- Subtract
- / Divide
- * Multiply
- ** Raise to the power of

Use rounded brackets to enclose those parts of the expression to be calculated first. Brackets can be nested to any level. On all MEDUSA platforms, expressions can be enclosed in round brackets ((...)) or angle brackets (<... >). Note, however, that you cannot use angle brackets within expressions. For example:

```
Valid expressions
<LEN + 10>
<A*(B-4)>
(A*(B-4))
Invalid expressions
<A*<B-4>>
(A*<B-4>)
```

Logical Operators

You can also use a variety of logical operators to define conditions for parameterization. Details of these operators can be found in the *MEDUSA Basis 1 Guide*. The *MEDUSA Parametric Design Reference Guide* describes how you can use logical operators in the Parametric Design system.

LAYERS AND PARAMETRIC SWITCHES

This chapter describes how you can change the way drawings are parameterized by changing parametric switches and layer properties. Layers have properties that affect whether or not they are drawn and whether elements on them can be changed.

- [How Layers Are Used](#) 108
- [Changing Layer Properties](#) 110
- [Parametric Graphics Control](#)..... 111
- [Parametric Switches and Command Texts](#) 112

How Layers Are Used

Layers have properties that affect whether or not they are drawn and whether elements on them can be changed. There are two properties that are used only in Parametric Design, transformable (TRN/UNTRN) and deletable (UNDEL/DEL).

Layer Defaults

Elements created by Parametric Design are placed on the following layers by default:

Layer	Element type
4	Dimensioning, baselines
13	Attachment points, instance groups
14	In-sheet command text, table elements
15	Parametric groups
16	Automatically inferred baselines
17	Special grid lines created by PAR GRIS COL, CIR, and TAN
28	Viewbox lines
56	Orthogonal 3D view prims
99	Error messages, grid lines (type STK)

If you wish, you can change these layer defaults using the PAR DDL Bacis1 command. PAR DDL is not covered in this book but it is described in the *MEDUSA Parametric Design Reference Guide*, chapter “Switches and Layers”, section “Changing Layer Properties”.

Layer Properties

There are 1024 layers, numbered from 0 through 1023. Each of these layers has five properties which are listed below. The **default** setting for all layers is indicated using bold text:

- **Transformable** or Untransformable (TRN/UNTRN)
- Deletable or **Undeletable** (DEL/UNDEL)
- **Visible** or Invisible (VIS/INVIS)
- **Hittable** or Unhittable (HIT/UNHIT)
- Protected or **Unprotected** (PRO/UNPRO)

You can use the Q LIS Bacis1 command to query the current layer properties.

Transformable/Untransformable (TRN/UNTRN)

Default: Transformable (TRN).

Elements on transformable layers can be parameterized, providing that the layers are also hittable (HIT) and unprotected (UNPRO). If you place an element on a layer that is untransformable (UNTRN), it may not be parameterized, but it is used to create the parametric grid.

Deletable/Undeletable (DEL/UNDEL)

Default: Undeletable (UNDEL).

Elements on deletable layers (DEL) are deleted when the drawing is parameterized. This is useful for example, for deleting dimensioning and in-sheet commands from the final parameterized drawing.

Visible/Invisible (VIS/INVIS)

Default: Visible (VIS).

Elements on visible (VIS) layers are displayed on the screen. If you make a layer invisible (INVIS) then all elements on that layer are parameterized normally, but they are not redrawn afterwards.

Hittable/Unhittable (HIT/UNHIT)

Default: Hittable (HIT).

If you place an element on a layer that is unhittable (UNHIT), it is completely ignored by the Parametric Design system. You can not locate it with probes or make it current. It is not used to build the grid, and it is not transformed when the rest of the drawing is parameterized.

Protected/Unprotected (PRO/UNPRO)

Default: Unprotected (UNPRO).

Elements on unprotected layers can be parameterized or deleted. Elements on protected layers (PRO) are not parameterized, but they may be used to create the parametric grid.

Changing Layer Properties

Most layer properties like visible and hittable can be changed in the graphical user interface using the menu *Layers -> Layer Manager*. Details on this are given in the *MEDUSA Drafting Guide*, chapter “Layers”.

If you want to change the properties transformable and deleteable, use the `Bacis1 LAY` command.

The LAY Command

You can use the `LAY` command both interactively and as an in-sheet command. It is advisable to use in-sheet commands to change layer settings rather than entering the commands interactively. This is because in-sheet commands are stored with the sheet, so that special layer settings are always used with a particular drawing and do not affect any other sheets. Refer back to “[Variables and Expressions](#)”, “[Creating in-sheet Commands](#)” on page 96 for information about creating in-sheet commands.

How to Use Layer Commands

You can use layer commands to do the following:

- Delete parts of the drawing during parameterization. For example, you may not want dimensioning, in-sheet commands or tables to appear in the final drawing.
- Prevent elements being parameterized, for example, table elements. You will see an example of this when you work through the next chapter.

You can do several things with a single layer command. For example, the following command when placed on a sheet prevents layers 4, 13 and 14 being transformed during parameterization and also deletes all elements on those layers from the final parameterized drawing:

```
LAY 4 13 14 UNTRN DEL
```

Parametric Graphics Control


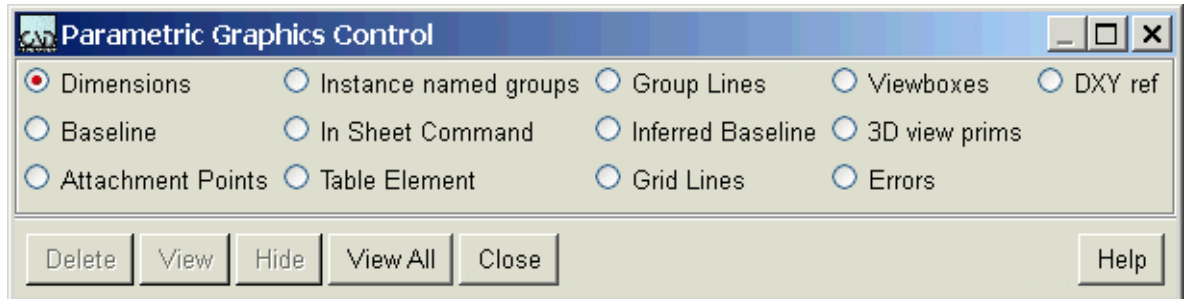
This tool is used to view, hide or delete parametric elements. For opening the dialog Parametric Graphics Control choose the tool Control visibility of parametric elements  from the tooltray Parametric.

Figure 91 Dialog Parametric Graphics Control




In the bottom of this dialog the buttons Delete, View, Hide and View all work on the elements defined by the selected option. Elements of the selected option are displayed highlighted in the drawing area. If there are no elements for the chosen option, the buttons are deactivated.

The buttons delete, hide or view the following elements:

Dimensions

Instance named groups


Group Lines

Option for elements of style Parametric Group Line created, for example, with the tool Creates parametric group lines  (see "Parametric Groups", "Creating a Parametric Group" on page 162).

Viewboxes

DXY prims

Baseline

Option for elements of style Static Base Line created, for example, with the tool Creates static base lines  (see "Getting Started", "Creating a Reference Point" on page 43).

In Sheet Command

Inferred Baseline

Option for automatically created baselines which can be drawn with the tool Draws baselines inferred when the baseline switch is on, see section "Getting Started", "Building the Parametric Grid" on page 46.

3D view prims

Option for 3D prims which can be used for parameterization, see "Appendix A Parametric Design Element Defaults", "Basic Parametric Elements" on page 214.

Attachment Points

Table Element

Grid lines

Errors

Option for error message texts.

Parametric Switches and Command Texts

Parametric switches affect, for example, how grid lines are generated. You can define them globally for the MEDUSA session and you can create them as in-sheet command texts. The following figure shows the toolset for setting parametric switches and command texts:

Figure 92 Toolset for Switches

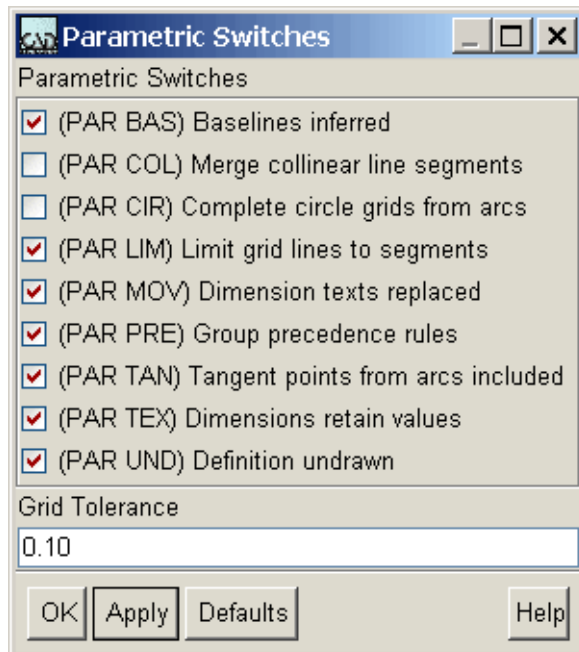


The tools from left to right are:

Displays the current settings of the parametrics system

provides the switches which affect the operation of the parametric design system and the Grid Tolerance globally for the MEDUSA session.

Figure 93 Dialog Parametric Switches



The following table gives a short description on each switch:

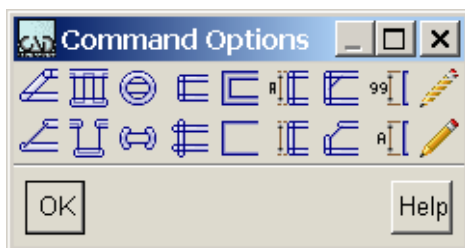
Switch	Description
PAR BAS	controls whether geometric constraints are inferred automatically or not. When PAR BAS is ON (default), the system uses the potential grid lines to infer geometric constraints automatically. When PAR BAS is OFF, constraints are not automatically inferred, so you need to specify them explicitly.
PAR COL	controls the generation of grid lines along non-overlapping collinear line segments. When PAR COL is ON, grid lines extend across non-overlapping collinear line segments. Default setting is OFF.

Switch	Description
PAR CIR	controls the generation of circular grid lines for all arcs. When the PAR CIR switch is ON, complete circular grid lines are generated from arcs of circles in the viewbox. Default setting is OFF.
PAR LIM	makes grid lines either limited (ON, default) or unlimited (OFF). Unlimited grid lines are limited only by the edges of the parametric viewbox.
PAR MOV	When the PAR MOV switch is OFF, dimension texts are replaced by new parameters but the original geometry is not transformed. If PAR MOV is OFF when loading parametric symbols, the symbol will be loaded at the same position it had in the definition sheet. Default setting is ON.
PAR PRE	When the PAR PRE switch is OFF, dimensioning within parametric groups is ignored and points move with the group. Default setting is ON.
PAR TAN	When the PAR TAN switch is turned OFF, the tangent lines of tangent point arcs are not used to create the potential grid. The default setting for PAR TAN is ON because the tangential grid lines are almost always needed to place the tangent points of tangent point arcs at grid intersections.
PAR TEX	When PAR TEX is OFF, geometry is transformed according to the new parameters in the dimension groups but the dimension texts display their original values after parameterization. Default setting is ON.
PAR UND	When PAR UND is ON (default), the original object definition is undrawn when you parameterize. This is done before drawing the new object, so the old component disappears before the new one is drawn. This switch is used when simulating mechanical movement.

Creates parametric command text

opens the dialog for placing the switches as in-sheet commands.

Figure 94 Dialog Command Options



From left to right the succession of switches is the same as it is in the dialog [Parametric Switches](#) from top to bottom. In the upper row you find the buttons for turning on switches and in the bottom row you turn switches off. Selecting one of the icons attaches the appropriate text to the cursor for placing it in the sheet.

Creates free parametric command text

is the same as [Creates parametric command text](#) but the text to place is free and not pre-defined. This tool works in the same way as other text tools from [Text + Dimension](#)

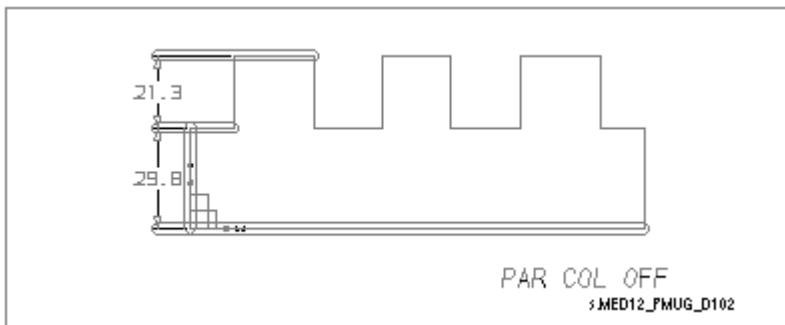
tooltray. For an example of using this tool see ["Variables and Expressions"](#), ["Creating in-sheet Commands"](#) on page 96.

The following sub-sections explain some of the switches more detailed.

The PAR COL switch

This switch controls the generation of grid lines along non-overlapping collinear line segments. When PAR COL is ON, grid lines extend across non-overlapping collinear line segments when you display the grid. You can change the setting for PAR COL by creating in-sheet command text and placing it inside the same viewbox as your drawing. The default setting for PAR COL is OFF.

Figure 95 Collinear Points With PAR COL OFF



The object in Figure 95 has several non-overlapping collinear lines. With PAR COL set to OFF, the grid lines extend only along the dimensioned line segments. To make the grid line extend to the other collinear lines you would have to dimension each one individually.

Figure 96 Collinear Points With PAR COL ON

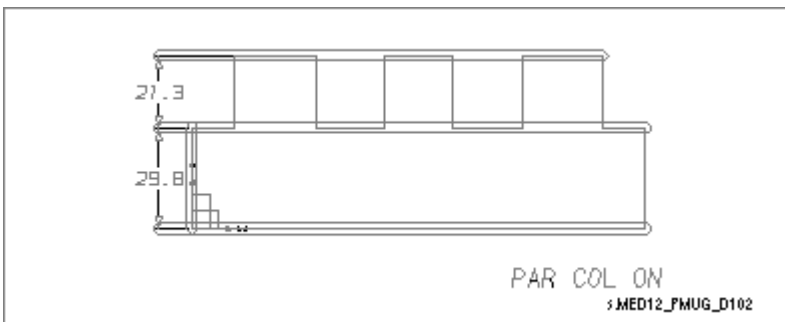


Figure 96 above shows the grid with PAR COL set to ON. The grid lines span the gaps between any points in the viewbox that are collinear. This gives a complete grid without having to add further dimensions.

The PAR BAS switch

The PAR BAS switch controls whether geometric constraints are inferred automatically. When PAR BAS is OFF, constraints are not automatically inferred, so you need to specify them explicitly. When PAR BAS is ON, constraints are inferred automatically from the object geometry.

Refer to ["Geometric Constraints"](#), ["The PAR BAS Switch"](#) on page 79 for information about how to use PAR BAS to speed up parameterization.

The PAR CIR switch

This switch controls the generation of circular grid lines for all arcs. When the PAR CIR switch is ON, complete circular grid lines are generated from arcs of circles in the viewbox when you display the grid. The default setting for PAR CIR is OFF.

The object in Figure 97 is dimensioned radially. When you draw the grid with PAR CIR set to OFF, as shown, only the dimensioned arc is supported.

Figure 97 Effect of PAR CIR OFF

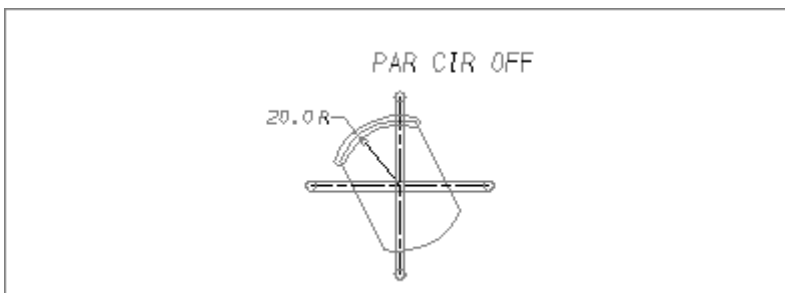
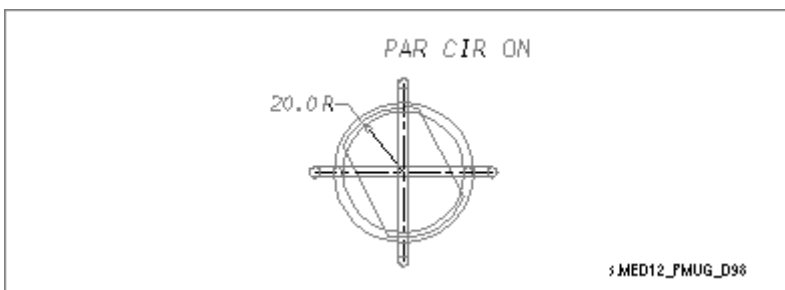


Figure 98 shows what happens when the grid is drawn after an in-sheet PAR CIR ON command has been added. A complete circular grid line is generated supporting the whole circle, and not just the dimensioned arc.

Figure 98 Effect of PAR CIR ON



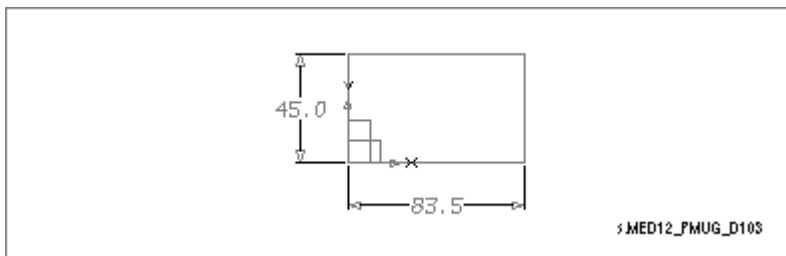
The PAR UND switch

This switch is used when simulating mechanical movement.

When PAR UND is ON, the original object definition is undrawn when you parameterize. This is done before drawing the new object, so the old component disappears before the new one is drawn. This is the default setting.

With PAR UND set to OFF, the original definition remains visible until you refresh the screen. When you parameterize with PAR UND OFF, you can see the parameterized and unparameterized versions of the drawing simultaneously.

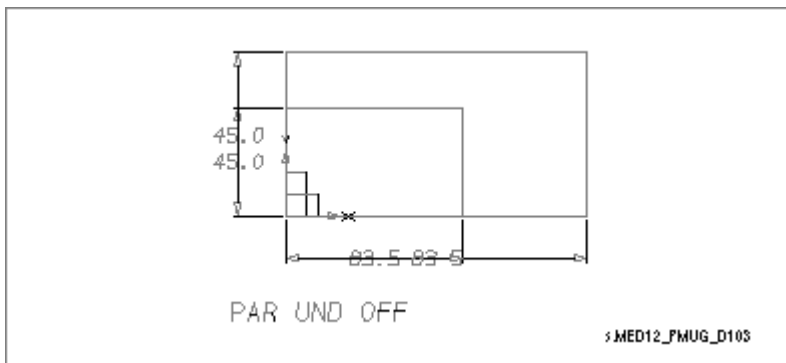
Figure 99 Original Geometry With New Parameters



The object shown in Figure 99 has been prepared for parameterization. The original dimension values have been replaced with new parameters.

Figure 100 shows the result of a parameterization. Both the original and the parameterized versions of the object (complete with dimensioning) can be seen. The original object definition can be removed by redrawing the graphical area.

Figure 100 Original and Parameterized Versions



TABLES

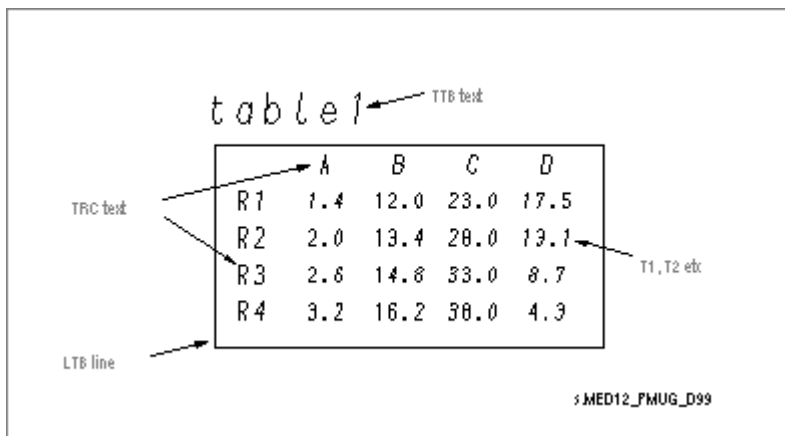
This chapter shows how you can use tables to store one or more groups of values on a sheet for subsequent during parameterization. A table is composed of a set of special elements.

- Structure of a Table 118
- Creating a Table 120
- Where to Place a Table 123
- Accessing Values From a Table 124
- Worked Example 126
- Parameterize Variant from a Table 131

Structure of a Table

Using a table you can assign values to several variables at once with a single TBL command. Specifying a table row or column has the same effect as giving a LET command for each entry in that row or column. Figure 101 shows the essential parts of a table.

Figure 101 The Parts of a Table



Description of Table Parts

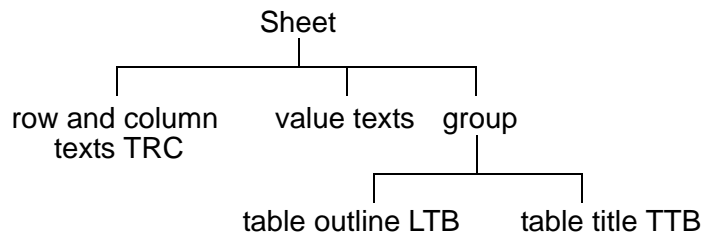
You must construct a table as a group.

Part	Figure 101	Element Type	Function
Title	table1	TTB text	Defines a name so that you can reference a particular table. You can have one or more tables on a sheet. The table name text is optional if you have only one table on the sheet. The TTB text must be in a group
Box		LTB line	Defines the table outline. The LTB line must be in the same group as the TTB text.
Row and column names	A B C D R1 R2 R3 R4	TRC text	Defines which variable names take values from the table. You can define up to 100 rows and 100 columns
Values	1.4 12.0... 2.0... 2.8... 3.2...	Any text type	Values can be any valid expression which evaluates to one or more numbers, or a string of characters. Place the text datums at the intersection of horizontal and vertical lines from the row and column text datums. Use either construction lines or a construction grid to place your values.

Structure of The Table Elements

Figure 102 shows the structure of table elements. The table outline and name text are part of a group. The group containing the table outline and name text is a sheet element, as are the table row and column texts and value texts.

Figure 102 The Structure of Table Elements



You also can put the value, row and column texts into the group with table title and outline in order to keep the table together.

Additional Table Elements

You can add lines to the table, for example to divide the rows and columns. These lines can be of any type and are ignored when you give a TBL command. However, if the table is inside a viewbox, you must make sure that any extra lines you create are placed on an untransformable layer. This prevents Parametric Design trying to parameterize them along with the object geometry.

Points to Remember When You Use Tables

When you use tables, remember the following:

- The table can have up to 100 rows and columns. This gives you the potential to set 99 variables at once by giving a single command.
- The values in a table can be vectors, text strings, or expressions.
- You can omit angled brackets (<... >) from expressions within a table.
- All the entries in a table must be present before you can access the values.
- If there is only one table on the sheet, the table name is optional.

Creating a Table

For creating tables MEDUSA provides the following toolset:

Figure 103 Toolset for Tables



The tools from left to right are:

Creates a parametric table boundary line

creates a line of style Parametric Table box line, type LTB. This line encloses a table in case that it is placed outside the viewbox of the geometry which shall be parameterized. After selecting this tool a new group is created automatically in which the line has to be.

Creates table name of type TTB text

is used after the tool Creates a parametric table boundary line is done. The created table boundary line has to be active in order to put the text in the same group as the boundary line, otherwise you get an error message. The text is of type TTB.

Creates table row & column text

is used for creating the table row and column text manually by placing every text individually on the sheet. This text has to be on sheet level and it is of type TRC.

Creates a parametric table

This tool combines all the tools explained above. For details on using this tool see [“Procedure - Method2” on page 121](#).

Procedure - Method1

Use the following procedure to create a table:

1. Create the table outline using the tool Creates a parametric table boundary line.
This line is automatically created inside a new group.
2. Name the table using the tool Creates table name of type TTB text.
Table name text must be also part of the new group.
3. Create table row and column texts, with the tool Creates table row & column text.
4. Choose a text tool and define and place text for the table values.
These texts can be of any type but they must be accurately aligned with the row and column texts.

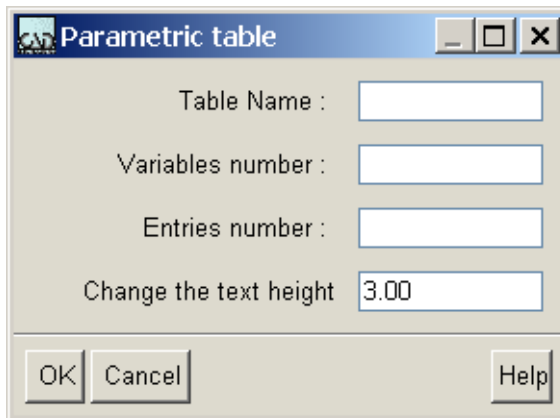
Please note: If you place the table inside the viewbox of the geometry which shall be parameterized with this table, do not forget to specify the table outline untransformable with a layer command text like `LAY 14 UNTRN` otherwise you get error messages when parameterizing later on.

Procedure - Method2

Use the following procedure to create a table:

1. Create the table using the tool *Creates a parametric table.*
The dialog *Parametric table* opens.

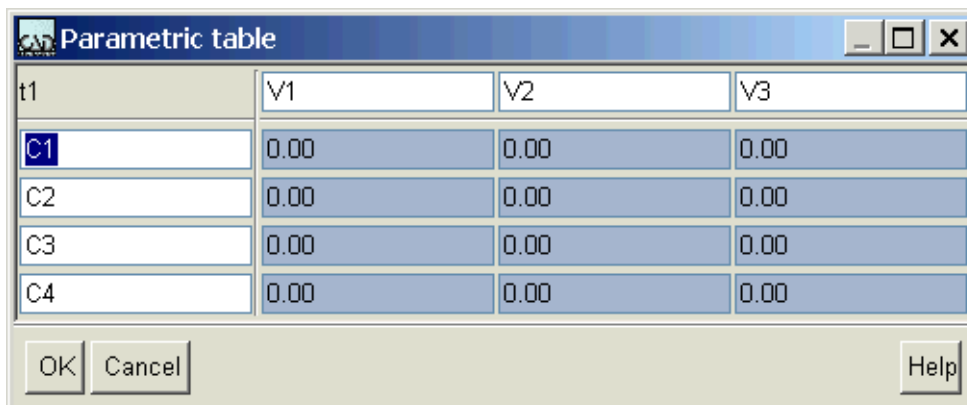
Figure 104 Dialog for Defining Tables



The following parameters can be given:

- *Table Name* is the table identifier and it will be the text of type TTB.
 - *Variables number* is the number of columns the table shall have.
 - *Entries number* is the number of rows the table shall have.
 - *Change the text height* defines the height of all table texts.
2. Type in the *Table Name*, the *Variables number* (columns of the table) and the *Entries number* (rows).
 3. *Click left* on *OK*.
The table opens with the defined number of rows and columns.

Figure 105 Example of a Table with 4 entries/rows and 3 variables



The first row entry is selected and you can type in the rows name.

4. Type in the table entries.

Use the TAB key of your keyboard for doing so. Every time you press the TAB key, the cursor jumps to the next table cell in the column. If the last table cell of a column is done, it jumps to the first cell of the next column, which is the name of the variable.

5. After having defined all table entries *click left* on the button OK.

Now the table is attached to the cursor and you can place it on the sheet.

Table Values

Table values can be of any text type. Use either a MEDUSA 2D grid or construction lines to place the table values in correct alignment with the row and column texts. If they are not aligned, it may not be possible to separate the row and column entries when the drawing is parameterized. If this happens, you will receive the following error message on the screen when you try to parameterize the drawing:

Cannot resolve table

Variables and Expressions in Tables

Table values can be variables or expressions as well as simple numbers. For example, the table in Figure 106 shows the values of variables B and C in row Row1 as expressions. Note that the sequence of variables and expressions in a table is important.

Figure 106 Table With Variables

	A	B	C
Row1	10	A+D	B/2
Row2	20	30	40

The value of each variable used in a table refers to its value before the execution of the TBL command and not to its value calculated during the execution of the TBL command.

Tables and Parametric Symbols

If you include a table in a parametric symbol definition, you must save the geometry, the table and the in-sheet TBL command together as part of the symbol definition. There is a worked example in "Symbols", "[Loading Parametric Symbols With Tables](#)" on page 144 showing how to load a parametric symbol definition using values from a table.

Where to Place a Table

When you create a table you place object points on the sheet. If the table is inside a parametric viewbox the points in the LTB line will be treated like any other point in the viewbox and Parametric Design will attempt to parameterize them. To prevent this, do one of the following:

- Place the table outside the viewbox
- Make the layer containing the table untransformable

You can place a table anywhere on the sheet. A table does not have to be in the same viewbox as the in-sheet command which refers to it.

Untransformable layers are not affected by parameterization. Therefore, you do not have to place elements on untransformable layers on the grid. When you start up MEDUSA, all layers are transformable.

To make a table untransformable, use the following procedure:

1. Find out what layer the table elements are on. If you created the table elements using the tools given in [“Creating a Table” on page 120](#), they will be on layer 14.
2. Create an in-sheet layer command making that layer untransformable, for example:

```
LAY 14 UNTRN
```

Make sure that there is nothing on this layer that you want to parameterize.

3. Place the in-sheet command text somewhere inside the viewbox.

If necessary, refer back to [“Layers and Parametric Switches”](#), [“Changing Layer Properties” on page 110](#) for more information about changing layer properties.

Accessing Values From a Table

You use the TBL command as an in-sheet command to access values in a table.

Please note: If you used “[Procedure - Method2](#)” on page 121 for creating tables, you already have the TBL command. It is at the upper right corner of the table.



How the TBL Command Works

Specifying a row or column of the table with a single TBL command is equivalent to giving a LET command for each entry in that row or column.

- When you specify a row name in the TBL command, the values for that row are assigned to the variables specified in the columns
- When you specify a column name in the TBL command, the values for that column are assigned to the variables specified in the rows

Procedure

These are the steps for accessing table values:

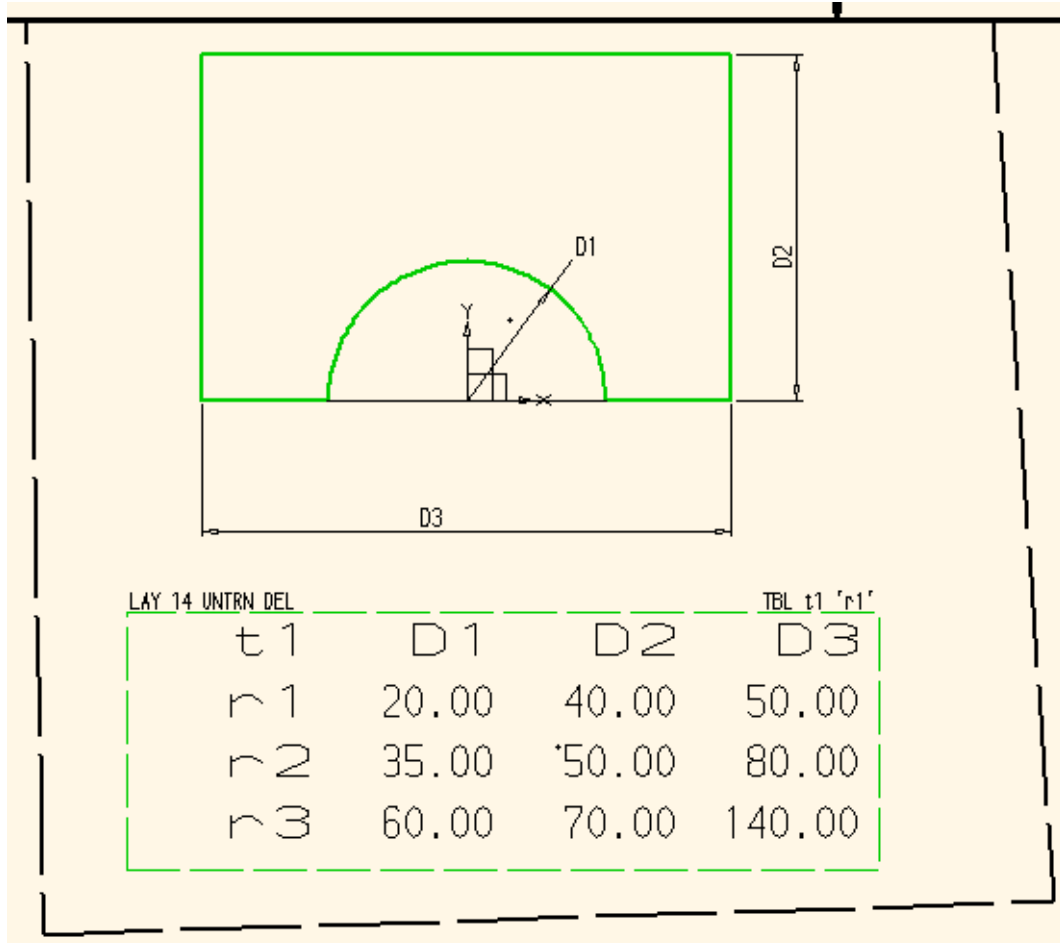
1. Choose the tool  Creates free parametric command text.
2. Inside the dashboard insert the text which calls the table and the row whose parameters you want to use, for example, `tbl table1 'row2'`.
3. Place the text inside the viewbox with the geometry which shall be parameterized.
4. Run parameterization using the tool . The geometry should be displayed according to the values defined in the given row.

Please note: If you copy and paste a table on a sheet ensure that you use [Recreate Structure](#) from the popup menu while pasting in order to get the pasted table into its own group. Otherwise the pasted table is in the same group as its original which causes an error message when you parameterize it.

Example

The sheet shown below in Figure 107 has an in-sheet TBL command. When the sheet is parameterized, the values from the first row of the table, r1, will be used for the dimension variables D1, D2 and D3.

Figure 107 Example of an In-sheet TBL Command



Worked Example

This example shows how you can use a table to produce three parameterized versions of an object.

Drawing the Object

Draw and dimension the object shown below in Figure 108 using the following guidelines:


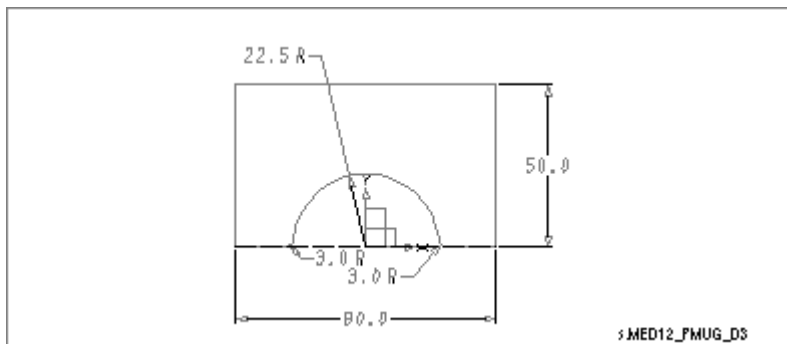
- The prim datum is positioned at the center of the arc
- The dashed line across the arc ensures that the two collinear horizontal line segments are supported on the grid. Create the line with the tool **Draws lines connecting all collinear straight lines** . Make sure that the connecting line is on an untransformable layer before parameterization.
- The horizontal 80.0 mm dimension must be a chain dimension in order to pick up the center support from the prim

Figure 108 Original Component




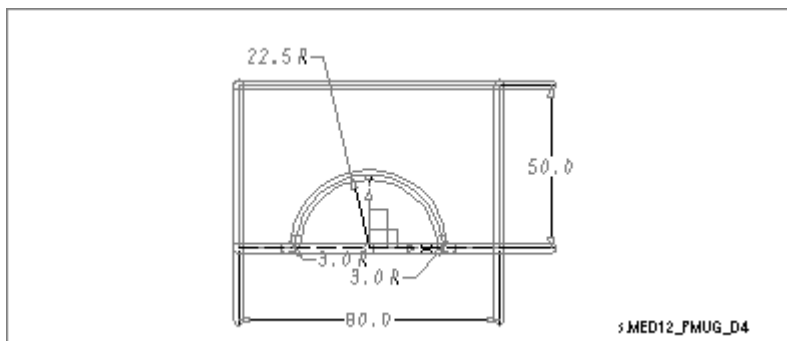


When you have dimensioned the object, use tool **Temporarily draws the grid corresponding to the current drawing**  to display the grid. The result is shown in Figure 109.

Figure 109 Dimensioned Component With Grid

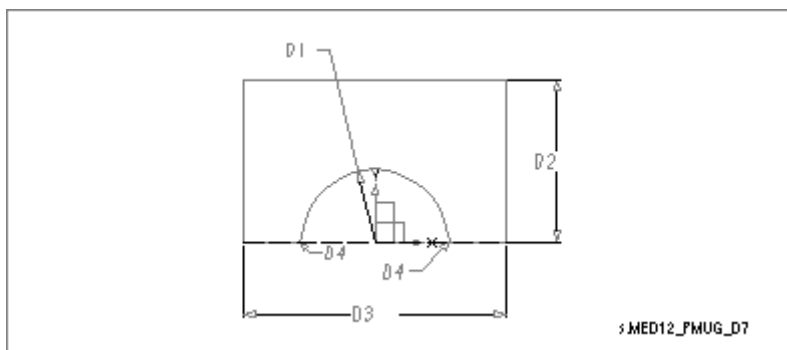


If there are no problems with the grid, test the drawing with the tool . Parameterizes the geometry then immediately undoes . Correct any errors before continuing.

Replacing Dimension Values

The next step is to change the dimension values to variables, as in Figure 110. It isn't necessary to change all dimension values to variables, as in this example. If you want some dimensions to remain constant then you can leave them as numeric values.

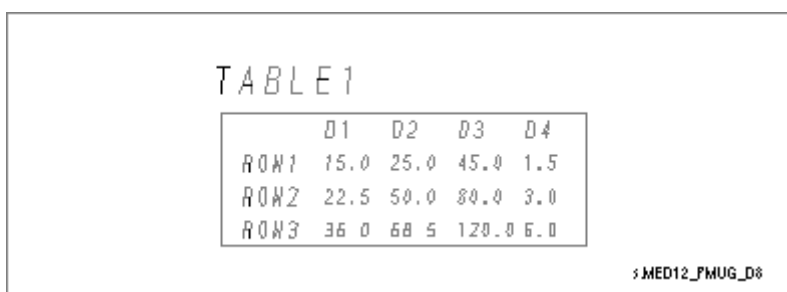
Figure 110 Changing Dimension Values to Variables



Creating the Table

Next, create the table shown in Figure 111 using the procedure described either in [“Procedure - Method1” on page 120](#) or in [“Procedure - Method2” on page 121](#). If you intend to use tables frequently, it is best to create a skeleton table and store it as a symbol. You can then load it into any master drawing that requires a table.

Figure 111 Table of Variable Values



	D1	D2	D3	D4
ROW1	15.0	25.0	45.0	1.5
ROW2	22.5	50.0	80.0	3.0
ROW3	36.0	68.5	120.0	5.0

The table is now complete, as shown in Figure 112. If you created the table using construction lines, delete them before continuing. In a moment you will be able to parameterize the compo-

ment by selecting a particular row of values from the table. Before that, you must add some in-sheet commands to the drawing.

Adding in-sheet Commands

Create the following in-sheet commands (text type TCO) and place them inside the viewbox:



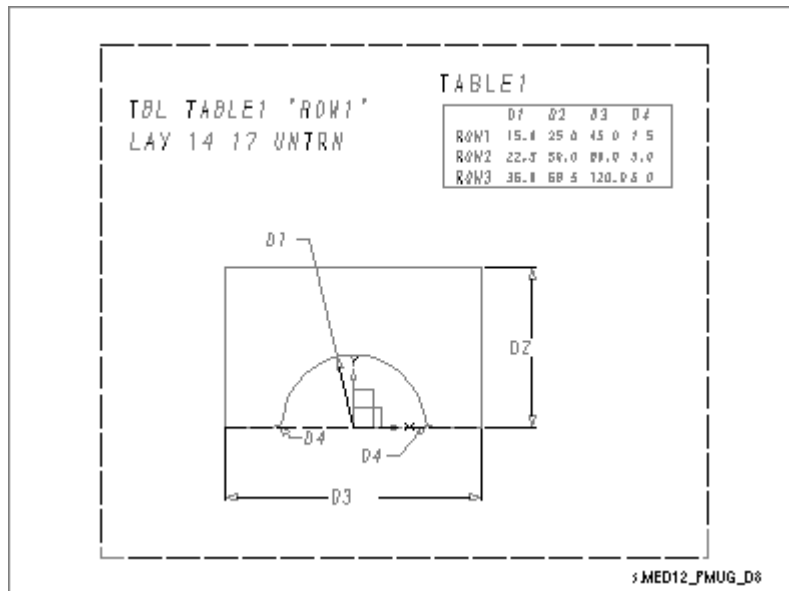
<pre>TBL table1 'ROW1'</pre>	<p>This tells Parametric Design which row of dimensions to use. When you run the tool Parameterizes the geometry then immediately undoes , the values from ROW1 of the table named TABLE1 will be used. Note that ROW1 must be enclosed in either double or single quotation marks.</p>
<pre>LAY 14 17 UNTRN</pre>	<p>This command prevents Parametric Design trying to parameterize the outline of your table by making the layer containing the table boundary line (layer 14) untransformable. Layer 17 also contains the line you added with the tool Draws lines connecting all collinear straight lines .</p>

Figure 112 Table and in-sheet commands



Parameterizing the Component Using Table Values


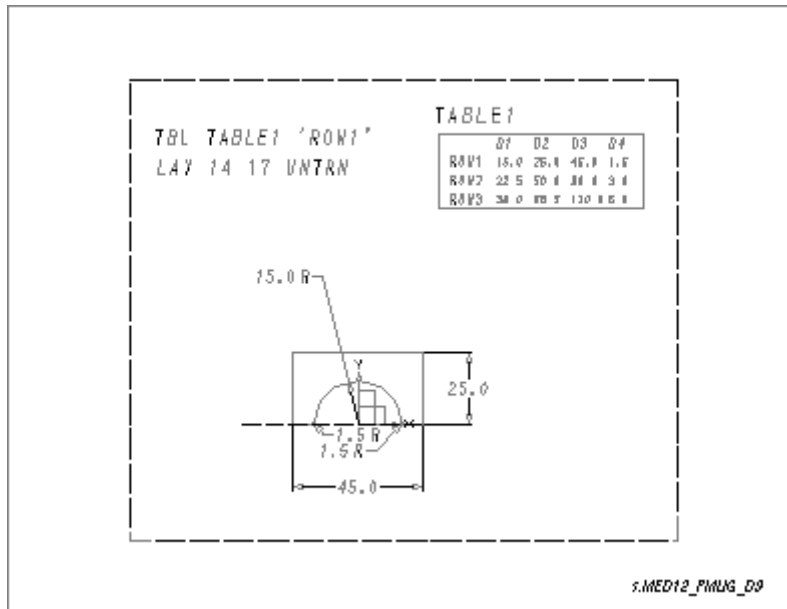

Parameterize the component using the tool Parameterizes the geometry then immediately undoes . The result is shown in Figure 113. The values from the first row, ROW1, of the table are used to calculate the new dimensions.

Figure 113 Effect of the temporary parameterization




Further Parameterization

Now parameterize the original drawing using the values in the second and third rows, ROW2 and ROW3. You can do this by changing the in-sheet command to `TBL table1 'ROW2'` and `TBL table1 'ROW3'` before running the tool Parameterizes the geometry then immediately undoes .

Deleting Elements During Parameterization

When you parameterize a drawing you can prepare the sheet so that unwanted elements are deleted during parameterization, making the final drawing clearer. For example, in the component you draw in the last example you could, if desired, delete the following elements during parameterization:


- All table elements, including table boundary line (LTB), name text (TTB), row and column text (TRC) and value texts
- In-sheet commands
- Extra lines you have added to develop the parametric grid

The existing in-sheet command on your sheet makes layer 14 untransformable. This ensures that Parametric Design does not try to parameterize the table outline when you run the tool Parameterizes the geometry then immediately undoes . To delete the whole table from the final drawing,

make the layer containing all of the table elements deleteable as well as untransformable. For example:

```
LAY 14 17 UNTRN DEL
```

Table values: Remember that table value texts, which can be any text type, will not be on the same layer as the other table elements by default. You must place them on the same layer. An easy way to do this is to place a group line around all the value texts then increment the layer number of the group to make it the same as the other table elements.

Extra lines: Layer 17 contains the extra line added with the tool *Draws lines connecting all collinear straight lines* . All extra lines must be made untransformable or Parametric Design will try to parameterize them. You can then choose whether or not to delete additional lines from the finished drawing along with the table elements and in-sheet commands.

Parameterize Variant from a Table

MEDUSA provides a tool for selecting a variant from a table and parameterize it.

Please note: Consider that variant parameterization is permanent.

These are the steps for using table values for parameterization:


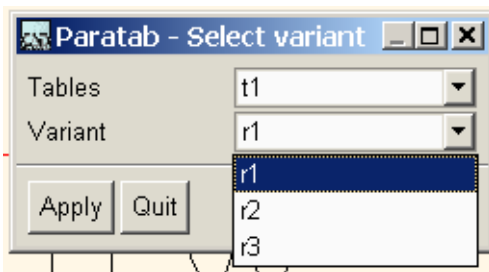
1. Load a sheet with tables used for parameterization.
The tables have to be inside the viewbox (Figure 115, “Example for Variant Parameterization, Input” on page 132) otherwise variant parameterization does not work.
2. Choose the tool  Selects a variant from a table and parameterizes it.
If there is no parametric table on the sheet an error message is given. Otherwise the following dialog opens:

Figure 114 Dialog for Selecting a Table Variant



The arrows behind the text fields open pulldown menus.

MEDUSA automatically zooms into the sheet displaying the first table available in the Tables pulldown menu.

The button **Quit** aborts the function.



3. Select the table whose entries you want to use for parameterization.
MEDUSA automatically zooms into the sheet displaying the selected table.
4. Select a Variant.
5. Choose **Apply** to run parameterization.
Parameterization is done immediately.
MEDUSA zooms out displaying the last view before choosing this tool.
The geometry associated with the selected table is parameterized permanently.
6. Go on doing one of the following:
 - If you are **not satisfied** with the result of parameterization, you can undo the parameterization using the tool **Undo**  from the toolbar.
 - If you are **satisfied** with the result of parameterization, save the sheet with a new name using the tool **Save as**  from the toolbar, in order to keep the input sheet.

Figure 115 Example for Variant Parameterization, Input

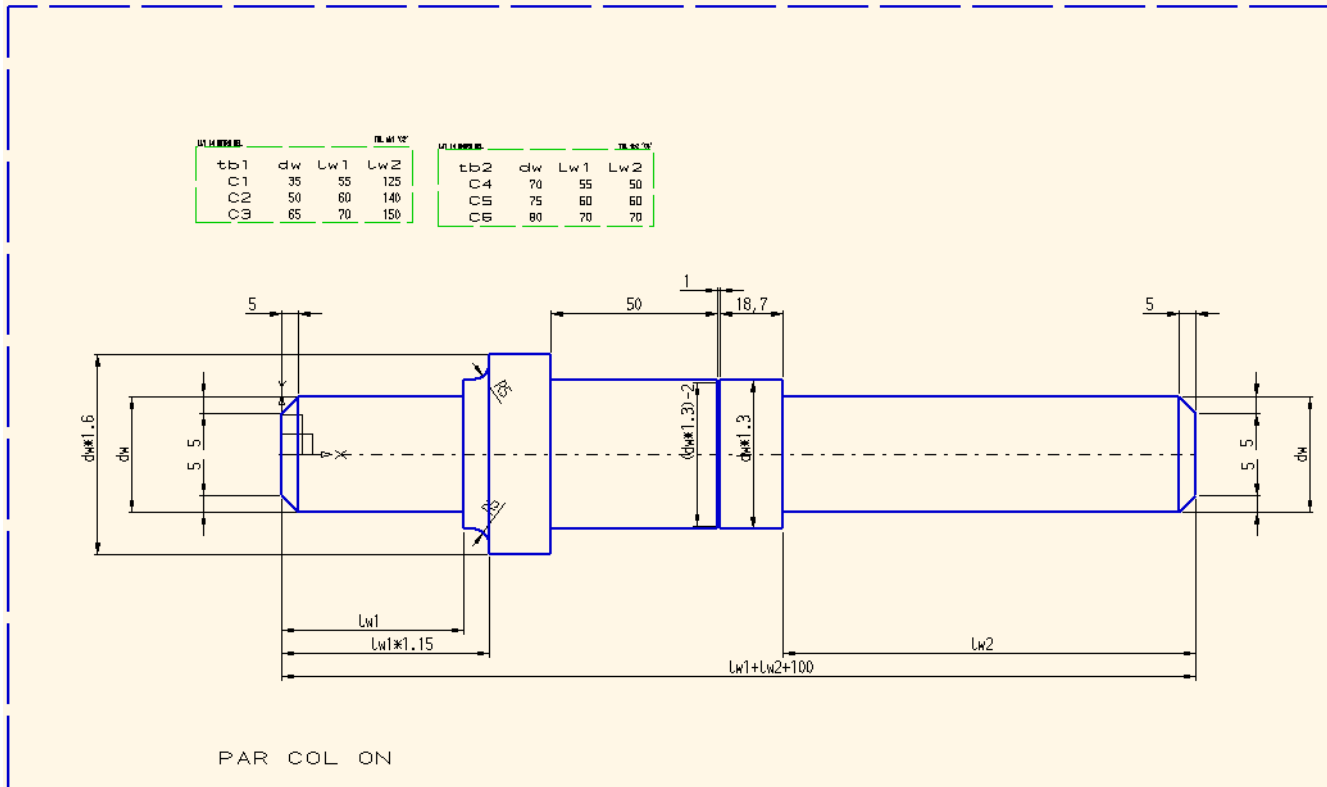
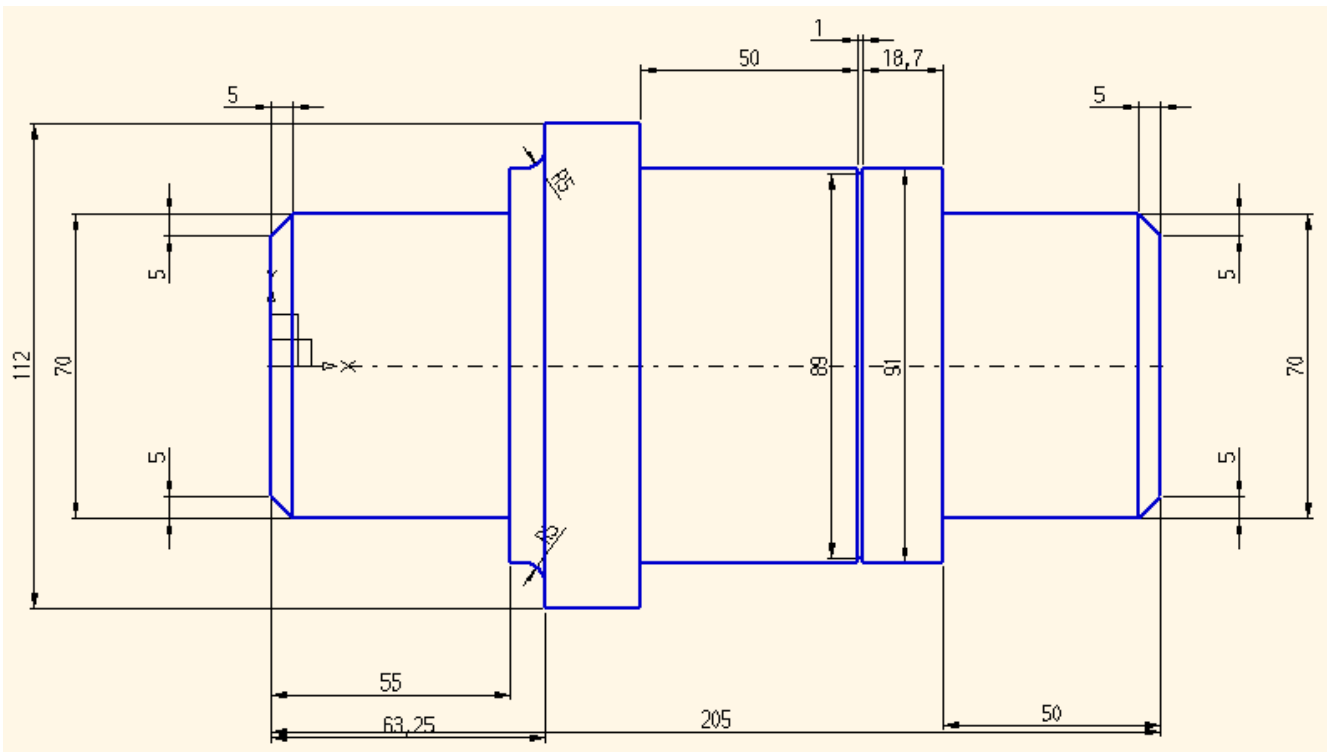


Figure 116 Example for Variant Parameterization, Result



SYMBOLS


This chapter describes how to use parametric symbols. A parametric symbol is a fully dimensioned drawing, complete with reference point, stored in a symbol file. You can parameterize it as you load it onto a MEDUSA sheet. You can load parametric symbols in one of two ways:

- Interactively, by loading each symbol one-by-one onto a sheet
- Automatically, by parameterizing a sheet containing CPI named groups. CPI named groups store information about parametric symbols. When you parameterize a sheet containing CPI named groups, Parametric Design automatically loads and parameterizes the required symbols.

- [Creating Parametric Symbols](#) 134
- [Loading Parametric Symbols](#) 138
- [Worked Example 1](#) 142
- [Loading Parametric Symbols With Tables](#) 144
- [Loading Symbols Using CPI Named Groups](#)..... 148
- [Worked Example 2 - CPI Named Groups](#) 150

Creating Parametric Symbols

You should already be familiar with storing and loading symbols in the 2D drafting system. Parametric symbols have to be prepared in a special way:

- Before you save the symbol you must add attachment points, text type ATP, which are used to position the symbol on the master drawing
- To load a parametric symbol onto the master drawing the tool `Load parametric symbol`  is available, which simultaneously loads and parameterizes a parametric symbol.

Attachment Points

So far, when preparing a drawing for parameterization, you have identified a datum or reference point by placing a prim or static baselines on the object geometry. When you create a parametric symbol definition instead of static baselines or a prim you must use one or more attachment points to specify a reference point.

An attachment point is a text of type ATP. This text consists of a variable name or an expression that evaluates to a set of X and Y sheet coordinates.

Positioning the symbol using attachment points: When you load a parametric symbol, the attachment points are used to position it on the new sheet. You can place more than one attachment point in a parametric symbol definition.

With one attachment point, the symbol will be loaded at the same orientation as the original. You can rotate it with the popup menu while loading, see [“Popup Menu while Loading a Symbol” on page 141](#).

If you use two attachment points, they can be moved relative to one another, for example to rotate the symbol as it is loaded. If you want to do so consider that the whole symbol and its dimensions have to be drawn with a orientation, which is neither horizontal nor vertical, otherwise the symbol can be placed horizontal or vertical only.

Grid lines: Attachment points generate grid lines in the same way as other reference points. Two grid lines are generated through the datum of the text, one horizontal and one vertical. When you display the grid for the symbol geometry, grid lines are drawn along any lines which pass horizontally or vertically through the attachment points.

If you have rotated the symbol and if you use more than one attachment point, grid lines are created parallel to the dimension lines.

Drawing the Symbol

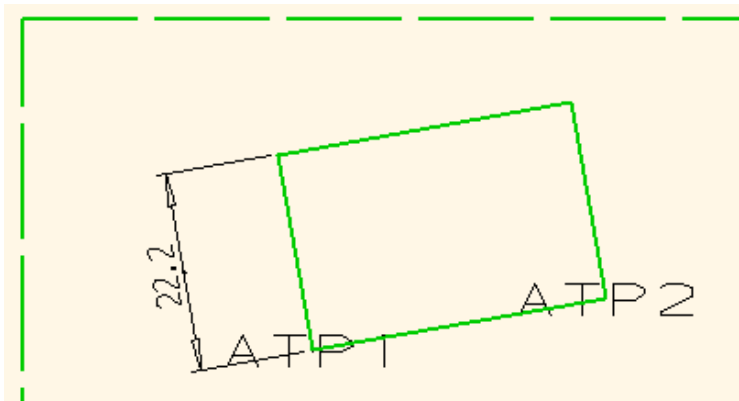
1. Draw a geometry, e.g. a rectangle.
2. Draw a view box.
3. Choose the tool *Creates parametric symbol attachment text*.
An attachment point is a text of type ATP. This text is a variable name or an expression that evaluates to a set of XY sheet coordinates.
4. Type the text, e.g. ATP1.
5. Place the text at any point of the geometry.
6. Type another text, e.g. ATP2, and place it also on the sheet.
Now the symbol is finished.

Preparing the Symbol

Once you have created the geometry and added attachment points, you must prepare the symbol definition before saving it into a symbol file.


7. To avoid the constraints caused by special cases, rotate the symbol complete with supporting lines **before** dimensioning. Make sure that no lines pass vertically or horizontally through the attachment points, or at angles of 30, 45, or 60 degrees.
8. Dimension the geometry using the perpendicular and parallel kinds of chain dimensioning in case you need to rotate the symbol when you load it.
Now your symbol should look like following:

Figure 117 Attachment Points



Testing the Drawing

For testing the symbol do the following:

9. Choose the tool *Load parametric symbol*  from the tooltray *Creation Tools*.
10. *Click left* on the view box of the symbol you just created.
11. Choose the button *Select* from the *Load parametric symbol* dialog.
The symbol is attached to the mouse pointer.

12. Define values for the variables if there are some and then *click left* on the sheet to place the first attachment point.
13. If you use the example of [Figure 117, "Attachment Points" on page 135](#), *click left* on the sheet to place the second attachment point.
The symbol is placed. If it is not, correct any errors before going on.


Some More Preparations Before Saving

14. Replace some or all of the dimension value texts with **variables**, e.g. replace the dimension value of the example with the letter *h*.
15. If desired, you can add **in-sheet commands** so that certain parts of the symbol are deleted as the symbol is loaded. You can delete attachment point texts, dimensioning and tables from the finished drawing using the layer command:

```
LAY 4 14 DEL.
```

Saving the Symbol

For saving the symbol,

16. Select the geometry and the attachment points of the symbol.
Consider that you also have to select dimensions and other elements inside the view box, which define the symbol. Do not include the viewbox for a parametric symbol!
17. Choose the tool **Save named symbol**  from the tooltray **Creation Tools**.
The selection is saved as symbol and you can use it on other sheets now (see ["Loading Parametric Symbols" on page 138](#)).

Using Tables With Parametric Symbols


Using a table you can load a parametric symbol at a number of positions on your sheet, each time using a different set of parameters from the table.


The procedure of creating a symbol with a table is as follows:

1. Draw the geometry and dimension it.
2. Replace the dimension texts with variables.
3. Create the table as explained in ["Tables" on page 117](#).
4. Create an in-sheet command for making the table outline, table elements and in-sheet commands associated with the symbol untransformable and for deleting them when you load the symbol:

```
LAY 14 UNTRN DEL
```

If you used the tool **Creates a parametric table** for creating the table, then this command is created automatically.

5. Choose the selection tool **Selects elements or groups at sheet level** .

6. Select the geometry, the dimensions, the table and the in-sheet command.
7. Save the selection as symbol using the tool Save named symbol  from the tooltray Creation Tools.

Now you can load this parametric symbol as explained in [“Loading Parametric Symbols With Tables”](#) on page 144.

Loading Parametric Symbols

If you want to place parametric symbols, you first have to define the values for the variables. Then, when placing the symbol, it is automatically parameterized.

Dialog


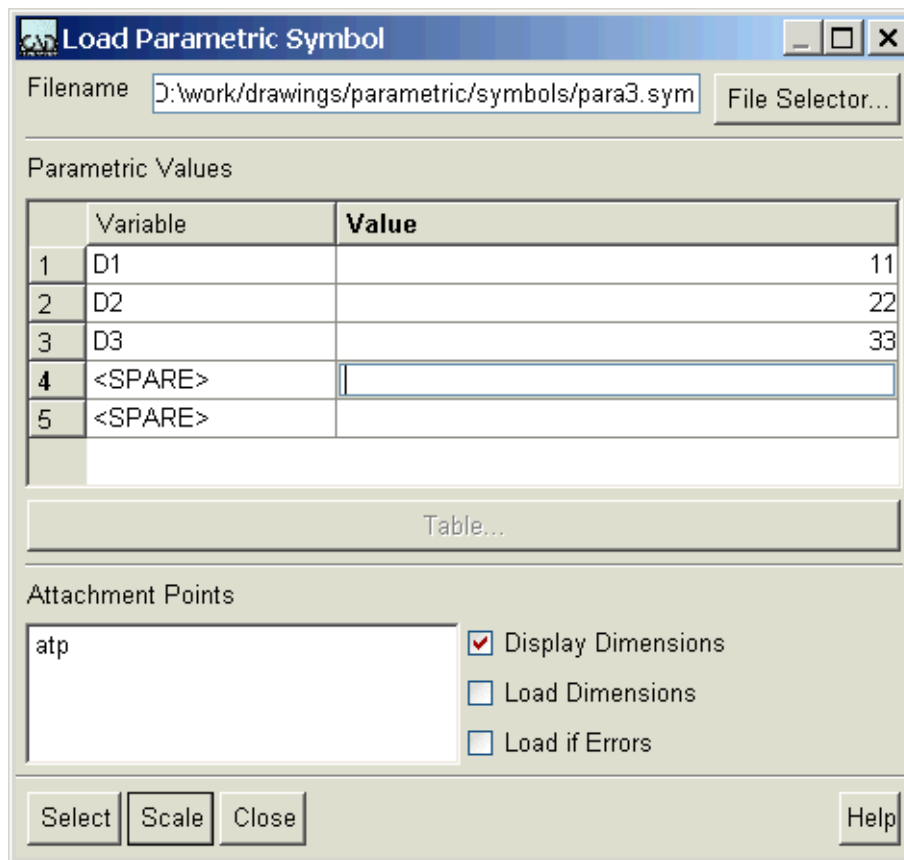
For loading a parametric symbol choose the tool Load parametric symbol  inside the tooltray Creation Tools. The following dialog opens:

Figure 118 Dialog for Loading Parametric Symbol



The parameters of the dialog are:

Filename

is the name of the parametric symbol with full path.

Parametric Values

displays a list of variables defined for the symbol. *Click* into an edit field, type the value and then *click* into another edit field to confirm the value.

Table

opens a further dialog in which you can select the row whose values shall be used for parameterization. Table is only activated if you selected a symbol file with table definition. For details see [“Loading Parametric Symbols With Tables”](#) on page 144.

Attachment Points

shows a list of attachment points defined for the symbol.

If you see only one attachment point, it is used for placing the symbol.

If you see more than one attachment point you can change the current point by *clicking* on a text inside the list. The selected attachment point is used for next probing on the sheet. After probing the next attachment point is displayed as selected inside the list and when probing next time the current attachment point is placed.

Options:

Display Dimensions

shows the symbol, which is attached to the cursor, with applied parametric values and its dimensions.

Load Dimensions

places the symbol with dimensions on the sheet.


Load if Errors

places the symbol on the sheet also in case that parameterization with the defined values fails. If this option is off, you can see error messages only at the dimensions of the symbol attached to the cursor.

The buttons are:

- Select chooses the selected symbol with name `Filename` and displays its variables inside the dialog. The chosen symbol is attached to the cursor drawn in the manner as it was constructed, Now you can set values for the variables.
- Scale applies the set values to the symbol attached to the cursor. By default the attached symbol is displayed as it was constructed.
- Close undisplay the dialog. The symbol can be placed until you choose `Exit Tool` from the popup menu.

Procedure

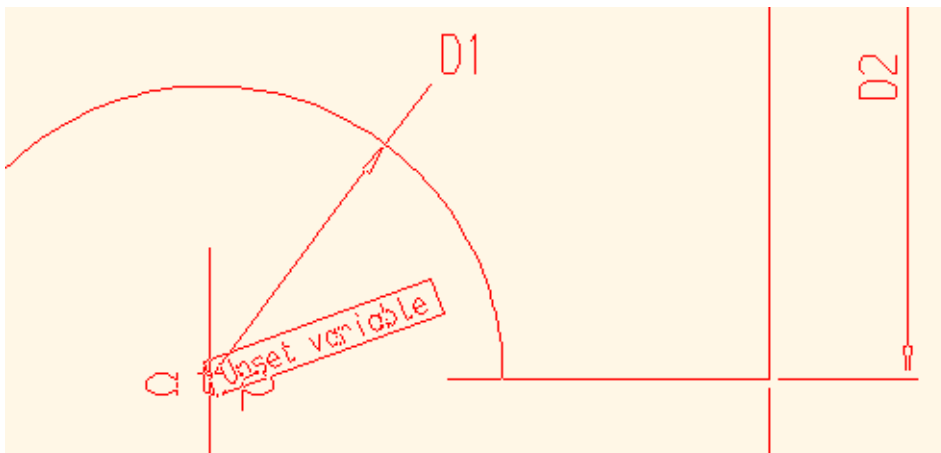
1. For loading a parametric symbol choose the tool Load parametric symbol  inside the tooltray Creation Tools.
The dialog Load Parametric Symbol opens.
2. Use the File Selector button to choose a symbol file.
3. *Click left* on the button Select.
The variables of the symbol and its attachment points are displayed.
4. Type in the values for the variables.

5. If you have more than one attachment point choose that point you want to place first.
At the cursor the chosen attachment point is displayed.
If you do not choose an attachment point the first one is used by default.
6. Choose the button *Scale*.
The symbol attached to the cursor changes according to the defined values.
7. If you want to rotate or mirror the symbol before placing it, use the popup menu (see [“Popup Menu while Loading a Symbol”](#) on page 141).
8. Place the symbol on the sheet.
You can place the symbol as often as you want. You also can change the values of the variables and *Scale* the symbol again.
9. If you finished placing parametric symbols choose *Exit Tool* from the popup menu.


Making Mistakes When Loading Symbols

Sometimes you may make a mistake when loading a parametric symbol. A common error is to forget to give a value to a variable. If any problems arise, the unparameterized symbol at the cursor gives the error messages.

Figure 119 Example for Error Messages while Loading a Parametric Symbol



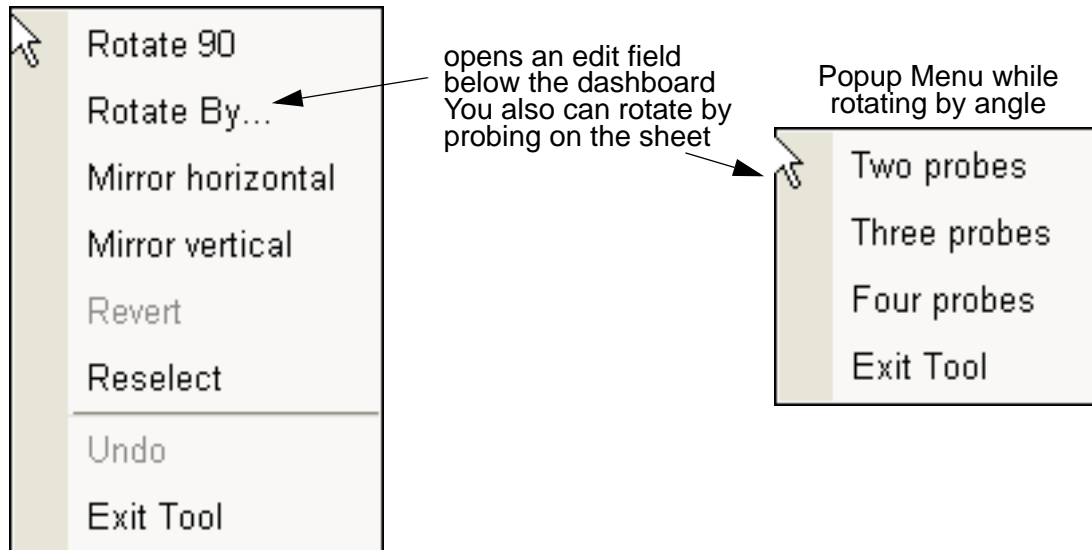
After you have fixed all errors, the variables are replaced by the values indicating that the symbol now can be placed.

If you suspect that there is an error in the original drawing you saved as a symbol, you can check this quite easily. Load the symbol into a blank area of your sheet using the tool *Load named symbol*  and investigate the problem.

Popup Menu while Loading a Symbol

While the symbol is attached to the cursor you can manipulate the symbol. You can rotate, mirror, revert rotation or mirroring and reselect for choosing a new symbol.


Figure 120 Popup while Loading Parametric Symbol



Revert resets rotation and mirroring.

Reselect activates the dialog for loading parametric symbols for choosing another symbol with File Selector. All entries for variables and attachment points are empty after clicking on Reselect.

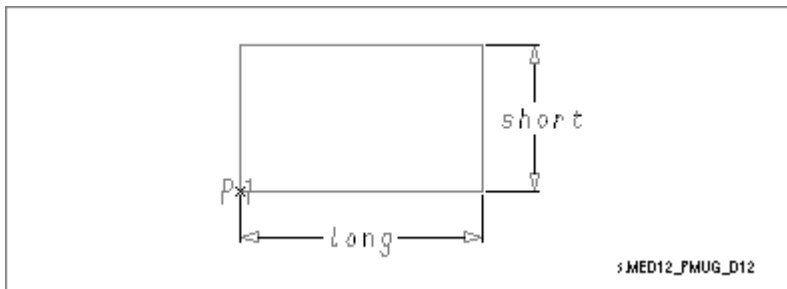
Worked Example 1


This example shows how to load a parametric symbol using the tool Load parametric symbol .


Creating the Symbol Definition

Draw and dimension a rectangle, as in Figure 121. Notice that the reference point is marked with an attachment point text, type ATP, which contains the variable name P1. Do not replace the dimension texts with variables yet.

Figure 121 Symbol Definition




Before you can use the tool Parameterizes the geometry then immediately undoes  to check that you have drawn and dimensioned the rectangle accurately, you must give the attachment point variable a value.

1. Choose the tool Creates free parametric command text .
2. Type the following command in the text edit field below the dashboard:

```
LET P1 = 100 100
```


With this the rectangle is placed at x=100 and y=100 on the sheet.

3. Place the text anywhere on the sheet but inside the viewbox of the drawn geometry.

If you do not give the attachment point variable a value before running the tool Parameterizes the geometry then immediately undoes , the error message Unset variable will be written onto the screen near the variable.

When you display the grid you will see that the attachment point supports grid lines in the same way as a prim. Grid lines are drawn horizontally and vertically through the attachment point.

Saving the Parametric Symbol

4. When you have replaced the dimension texts with variables, save the parametric symbol definition using the tool Save named symbol  from the tooltray Creation Tools. Be careful not to save the parametric viewbox with the symbol geometry. Remember that the position of the probe as you save the group is not significant as the attachment point and not the probed point is used to position the symbol when you load it onto the new sheet.

Loading the Parametric Symbol

You are now ready to load the symbol onto a new sheet.


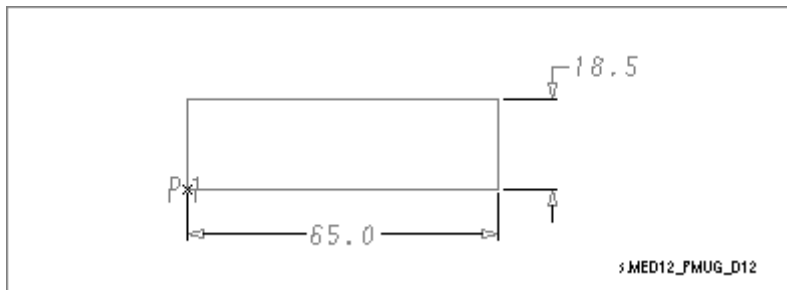
5. Choose the tool **Load parametric symbol**  inside the tooltray **Creation Tools**.
The dialog **Load parametric symbol** opens.
6. Choose **File Selector** from the dialog.
Another dialog opens for selecting the symbol file to load. After selecting a file name the dialog closes and the selected file is displayed in the line **Filename** of the dialog **Load parametric symbol**.
7. Choose the button **Select**.
The dialog updates showing the variables of the selected symbol. In the bottom the attachment points are given. The symbol is attached to the cursor still showing the variable names.
8. Specify the values for the variables.
9. Choose **Scale**.
The symbol attached to the cursor changes and displays the defined values now.
10. Place the symbol on the sheet.

Figure 122 Parametric Symbol Loaded Onto Sheet



11. Now try loading the rectangle at different places on the sheet by moving the cursor to new locations.
You can change the values but consider to choose **Scale** to apply the new values to the symbol before placement.
If you want to rotate or mirror the symbol, use the popup menu.


Loading Parametric Symbols With Tables

For parametric symbol definitions which may have several sets of variables, you can include a table of values in the symbol definition. When you include a table in a parametric symbol definition, make sure that you:

- Include an in-sheet TBL command in the symbol definition
- Make the table (frame and text) untransformable
- Save the table and all in-sheet commands with the symbol geometry

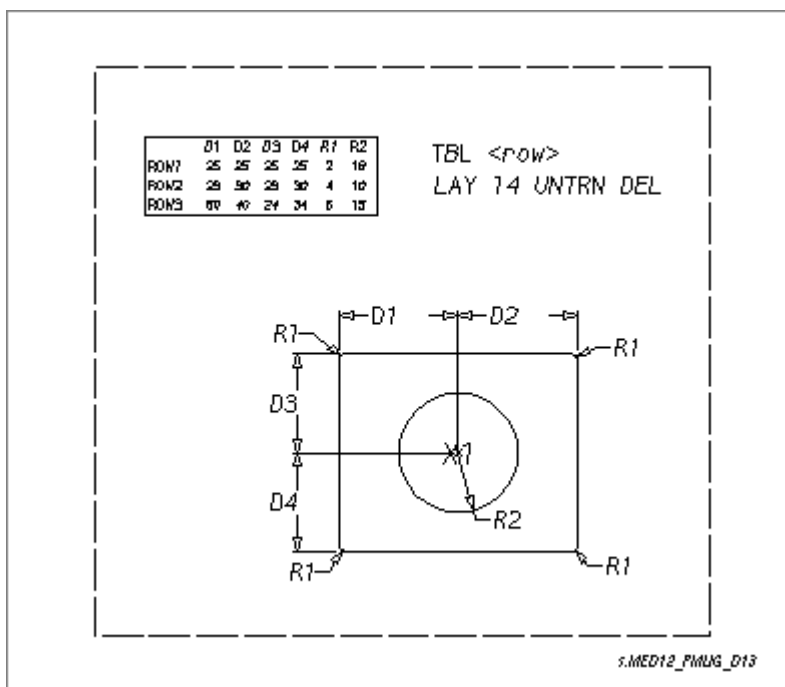
You can delete the table and any in-sheet commands included in the symbol definition when you load the symbol onto the new sheet.

Example

Using the tool  Load parametric symbol, you can load the symbol in Figure 123 using the values in the table. The only arguments you need to give are a datum position on the new sheet, and the name of the row of values you wish to use.

First draw and dimension the object shown below in Figure 123. Change some or all dimensions to variables. The original dimension values are not important as you will use the dimension values in the table when you load the symbol.

Figure 123 Table as Part of a Parametric Symbol



Creating the Table

When you have finished drawing the object, create the table of values. The table is detailed in Figure 124. Refer back to "Tables", "Creating a Table" on page 120 to see how to create a table.

Figure 124 Table

	D1	D2	D3	D4	R1	R2
ROW1	25	25	25	25	2	78
ROW2	23	30	23	30	4	10
ROW3	60	40	24	34	6	15

Adding in-sheet Commands

Finally, you need to add some in-sheet commands.

In-sheet TBL command: In "Tables" on page 117, you saw how to use an in-sheet TBL command such as `TBL table1 'R1'` to select a row from a table during parameterization. Every time you load a symbol using an in-sheet command that refers to an explicit row in a table, the new object dimensions will be taken from the same row. Replacing the row name in the TBL command with a variable name enables you to select a different set of dimensions each time you load the symbol.

When you want to place a row variable in a TBL command instead of specifying a row name, you must place the variable name inside angled brackets (<... >). For example:

```
TBL <row>
```


Using a variable is only needed when loading the symbol with `Basis1` commands. In the graphical user interface the command can remain unchanged because the rows selected in a dialog when loading the symbol. You do not need to specify a table name in the TBL command in this example because this is the only table on the sheet.

In-sheet layer command: You also need to add an in-sheet layer command to the symbol definition to do the following:

- Makes the table outline untransformable
- Deletes the table elements and in-sheet commands associated with the symbol when you load the symbol onto the new sheet

For example, `LAY 14 UNTRN DEL`. See "Tables", "Deleting Elements During Parameterization" on page 129 for more information about deleting table elements during parameterization.

Saving the Symbol

When you have drawn the object and the table and added the in-sheet commands, you are ready to save them all as a symbol. Make sure that you save the object, table, and in-sheet commands but not the viewbox. For selecting use the tool **Selects elements or groups at sheet level**  otherwise you get errors when trying to load the symbol.

Loading the Parametric Symbol

Now load the symbol. The following sequence shows how you might do this:


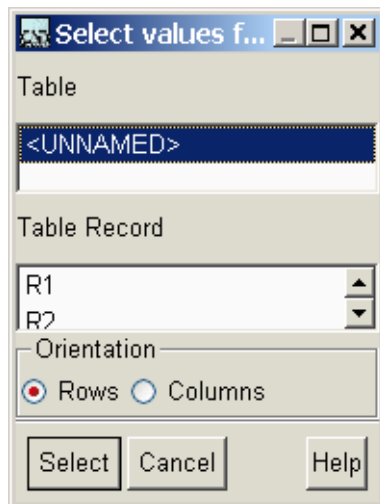
1. Choose the tool **Load parametric symbol**  inside the tooltray **Creation Tools**.
The dialog **Load parametric symbol** opens.
2. Choose **File Selector** from the dialog.
Another dialog opens for selecting the symbol file to load. After selecting a file name the dialog closes and the selected file is displayed in the line **Filename** of the dialog **Load parametric symbol**.
3. Choose the button **Select**.
The dialog updates showing the variables of the selected symbol. In the bottom the attachment points are given. The symbol is attached to the cursor still showing the variable names.
4. Choose the button **Table** to specify the values for the variables from the table saved with the symbol.
A further dialog opens providing the **Table** and the **Table Records**. With the options in **Orientation** you can display the specified values for rows or columns.

Figure 125 Dialog **Select values from Table**

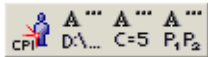


Loading Symbols Using CPI Named Groups

A CPI named group (CPI=Clump Parametrics Instance) is a special group that instructs Parametric Design to load and parameterize a parametric symbol at a specified point. By using CPI named groups you can load and parameterize automatically several instances of the same symbol on the sheet.

Toolset

Figure 127 Toolset for CPI Named Groups



The tools from left to right are:

Creates a CPI named group for loading parametric symbols

If you open the structure tree after choosing this tool, you can see the new group. For renaming it use the *right mouse button* while the cursor is upon the tree. This opens the popup menu. Then choose *Properties* to open the dialog *Named group properties*, in which you can change the *Name* of the group.

Creates SPS text within CPI named group

creates a text of style *SPS text*. This text has to be the file name of the parametric symbol with full path.

Creates SCO text within CPI named group




creates a text of style *SCO text*. This text holds a variable definition, e.g. `ARG var=10`. If the symbol has more than one variable definition you have to define several of these texts.



Creates SAT text within CPI named group

creates a text of style *SAT text*. This text holds the attachment point text as defined inside the symbol. If the symbol has more than one attachment point you have to define several of these texts. Ensure that these texts are placed carefully in order to create no undefined situation when running parameterization.

Creating a CPI Named Group

Use the toolset in the succession the tools are provided:

1. First open a new group of type CPI with the tool *Creates a CPI named group for loading parametric symbols* .
2. Create symbol name text using the tool *Creates SPS text within CPI named group*  and place it on the sheet using a probe.
3. Create command text of text type SCO using the tool *Creates SCO text within CPI named group*  to assign a value to a variable in the symbol definition and place it on the sheet. Repeat this step for each variable of the symbol.

4. Create attachment point text of type SAT using the tool  Creates SAT text within CPI named group  and place it on the sheet with a probe. Repeat this step for each attachment point text of the symbol.
5. Move all CPI named group elements on an untransformable layer.

Please note: For each symbol you want to reference a separate CPI group has to exist. Do not forget `arg` in front of variable assignments.

The following example shows a reference with one attachment point and one variable.



Figure 128 Example for a CPI named group

```
| pa      arg long = 44      |  
| e: \work\drawings\parametric\symbols\para2.sym |
```


Preparing the Drawing

You must place a CPI named group at all points where you want to load a symbol. Each CPI named group contains a number of texts, specifying the name, size, and position of each symbol that is to be loaded. You need to create a separate CPI named group for each symbol instance required. Each CPI named group contains all the information that is required to automatically load a symbol.

Dimensioning Attachment Points


You must dimension the symbol attachment points in each CPI named group to place them onto grid intersections. Dimensioning the attachment points is critical, as the attachment point datums must lie at grid intersections when the drawing is parameterized. Use the tool  Parameterizes the geometry then immediately undoes  to check that attachment points lie at grid intersections after you have dimensioned them.


Worked Example 2 - CPI Named Groups

This section takes you step-by-step through an example which uses CPI named groups to load three separate instances of a symbol onto a master drawing. You begin by drawing a gearbox cover and then create a symbol representing a tab. After adding CPI named groups, you can load and parameterize the tabs automatically. When you run the tool *Parameterizes the geometry* then immediately undoes it , the gearbox cover itself will be parameterized, and then each tab will be parameterized separately as it is loaded onto the master drawing.

Drawing the Component

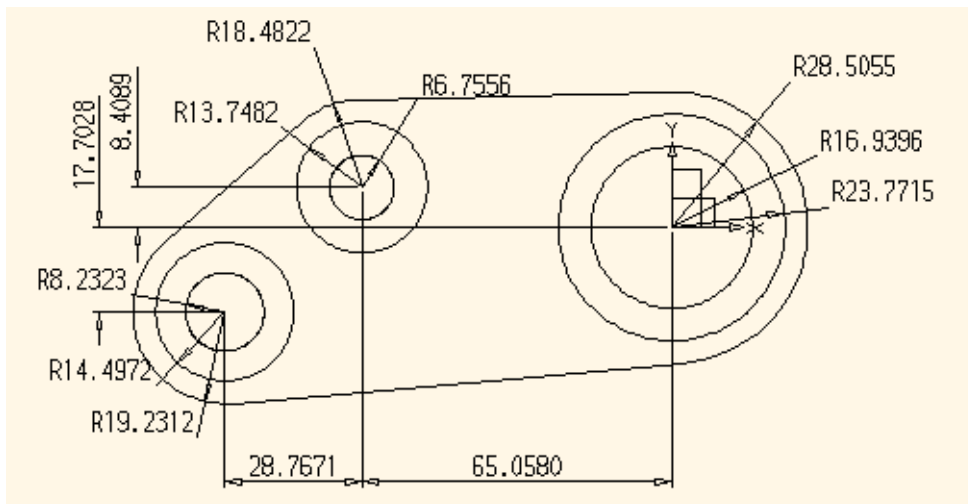
Draw and dimension the gearbox cover shown in Figure 129 using a view prim as a reference point. The base of the gearbox cover is horizontal, which prevents the base from being rotated during parameterization. However, it is still possible to move other parts of the drawing.

When you have finished drawing the component, use the tool *Draws the grid corresponding to the original drawing* and undoes it  to display the grid.

Then test the drawing with the tool *Parameterizes the geometry* then immediately undoes it .

Do not proceed unless the drawing can be parameterized.

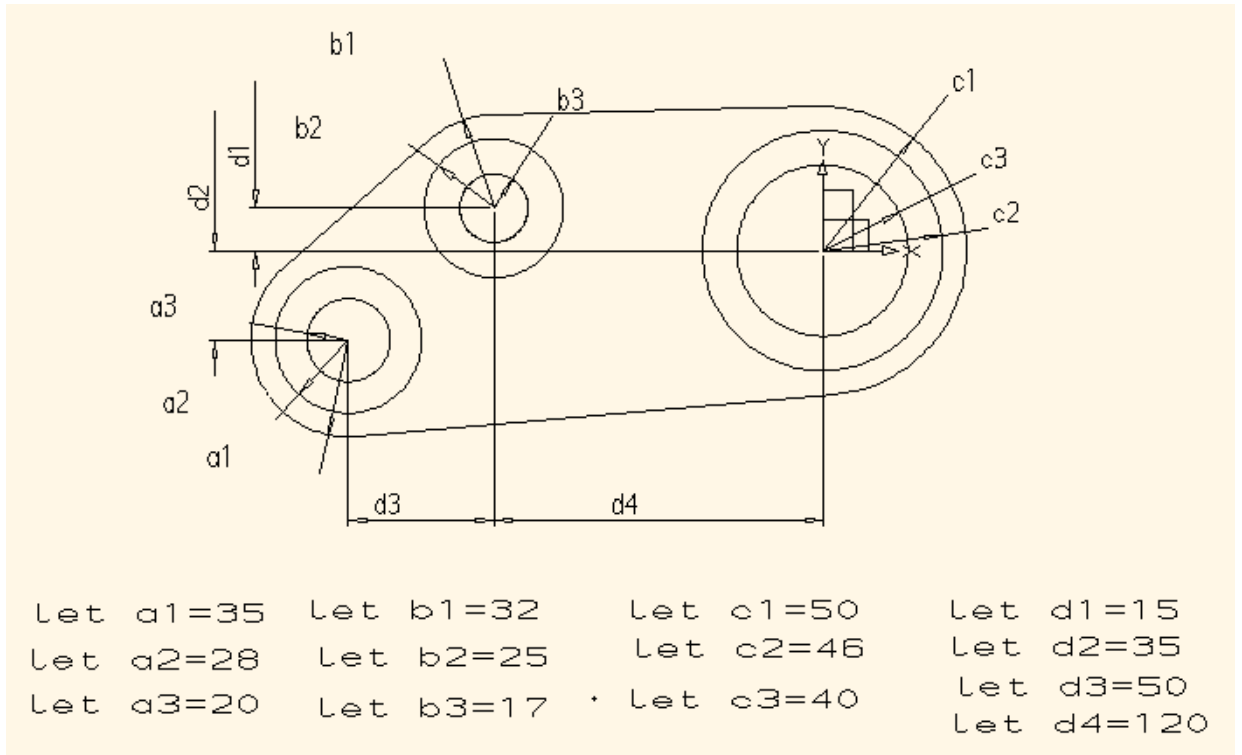
Figure 129 Dimensioned Component



Preparing the Component

Change the dimensions to variables as shown in Figure 130. All the variables in this example have been assigned values using in-sheet commands though you can use a table if you prefer. See “Tables” on page 117, for information on how to create a table.

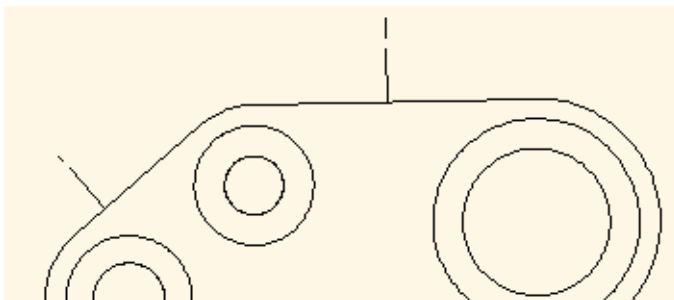
Figure 130 Changing Dimensions to Variables



Please note: In all subsequent illustrations, the dimensioning and in-sheet commands shown in Figure 130 will be omitted. This is for the sake of clarity.

Add lines to the drawing to mark the positions where the tabs will be placed, as in Figure 131.

Figure 131 Marking Positions of Tabs



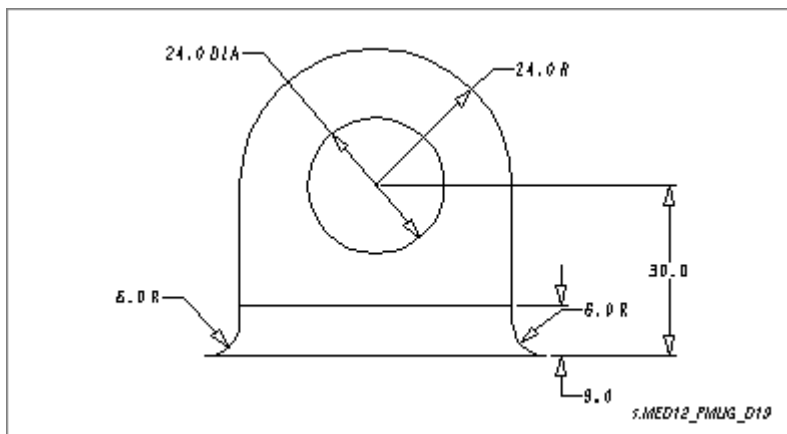
The lines used here are type L6, but you can use any line type. Each line is drawn perpendicular to the outline of the gearbox cover. This is so that when the tabs are added, they will always be perpendicular to the sides of the component. The length of each line determines the length of the parameterized symbol. The lines can be any length that you want and placed anywhere on one of the straight edges of the gearbox cover. In this example the dashed lines are drawn approximately at the center of each straight edge.


It is useful to place these lines on the same layer as the CPI named group elements, so that they can all be deleted together during parameterization.

Creating the Parametric Symbol

The next step is to create the symbol for the tabs. Draw and dimension the tab shown below in Figure 132. Remember to use the parallel or perpendicular kinds of chain dimensioning, rather than the horizontal or vertical kinds. This allows the tab to be loaded at any orientation.

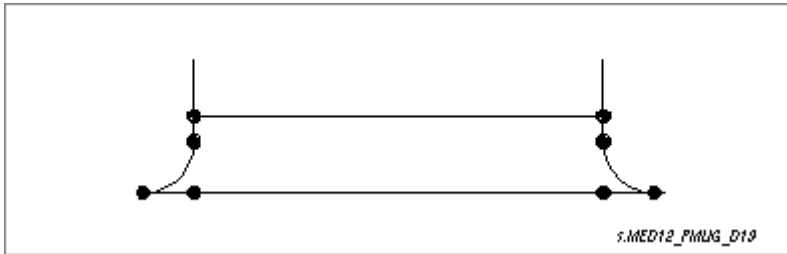
Figure 132 Dimensioned Tab



Creating the fillets: Add the fillets after you have completed the outline of the component. The fillets are tangent point fillets, created with the tool  from the Lines+Edit tooltray. The size of the fillets is not really important because they will be changed to whatever radius is required as you load them.

The important thing is to make sure that there are no coincident points in the drawing. If the end point of a fillet coincides with another point, the two points will be locked together during parameterization. Sometimes you may want certain points to remain locked together. However, it is usually better to create a master drawing that has no coincident points. Figure 133 shows where to create the individual points for the fillets.

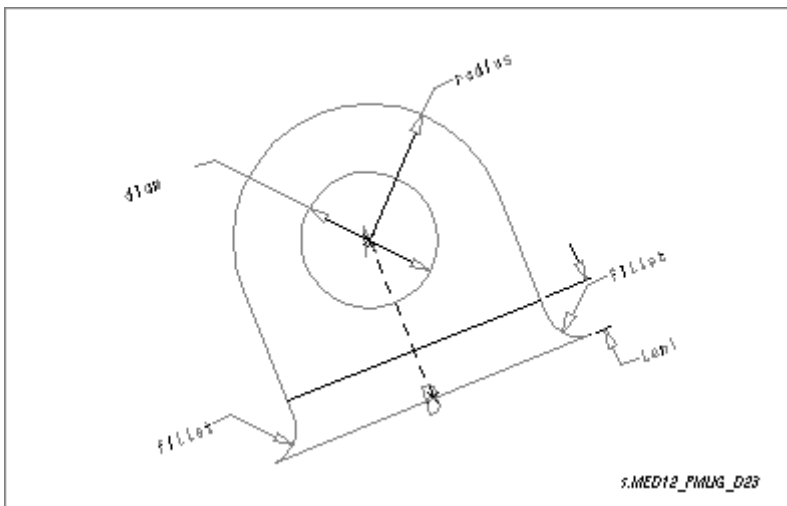
Figure 133 Individual Points on the Fillets





Adding attachment points: Add two attachment points variable texts, A and B as shown in Figure 134 below. These are texts of type ATP and must be linked by a line of any line type. Place the attachment points and the line linking them on the same layer so that you can delete them easily when you load the symbol.

Adding an in-sheet command: Add an in-sheet command to delete unwanted elements during parameterization, for example, `LAY 4 14 DEL`. This deletes the layers containing the attachment point texts, linking line, and dimensioning.

Figure 134 Tab Prepared for Saving



Testing the drawing: When you have drawn and dimensioned the symbol, test it with the tool . Parameterizes the geometry then immediately undoes . If this is successful, change the dimensions to variables and then rotate the whole symbol through an arbitrary angle, as in Figure 134.

You are almost ready to save the symbol. If you save the in-sheet command with the symbol, ensure that the layer containing in-sheet commands is deleted during parameterization.

Creating CPI Named Groups

Having created the tab symbol, you must now create a CPI named group at every position where the tab is to be loaded as shown in Figure 135. The sequence required to create one CPI named group is given below. Figure 136 shows one of the CPI named groups in detail.

Figure 135 CPI named Groups in Position

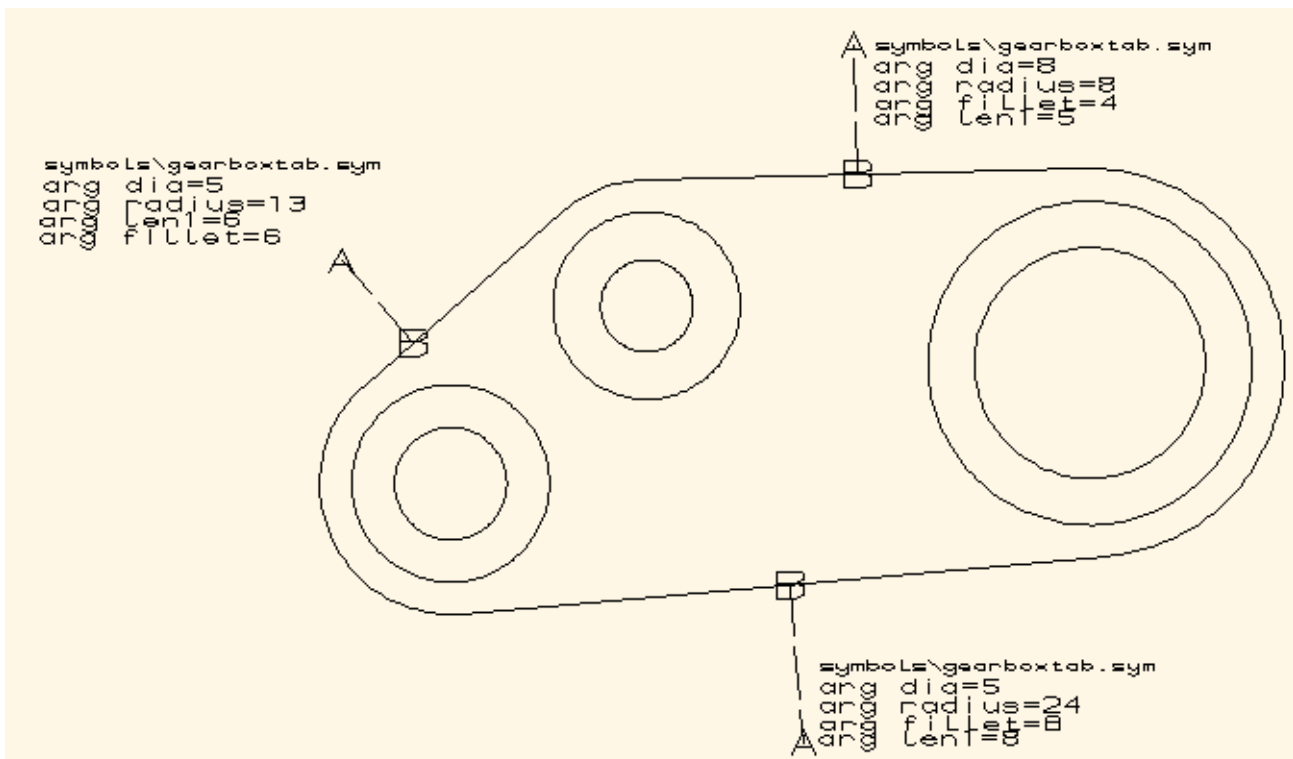
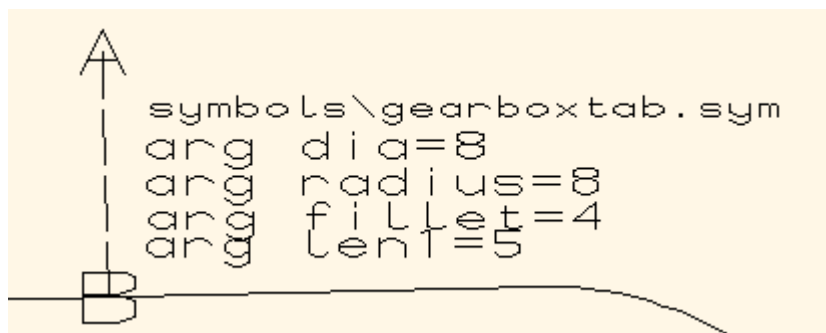




Figure 136 Detail of CPI named Group





Creating a CPI group: First, open a new CPI group using the tool  Creates a CPI named group for loading parametric symbols.

Creating SPS text: Then create a new text of type SPS using the tool  Creates SPS text within CPI named group. The SPS text must be part of the CPI group. Enter the symbol file name into the

text edit field below the dashboard and then place the text on the sheet with a probe. For clarity, it is best to position the text near to where the symbol will be loaded, although the exact position of the SPS text is not important. If the symbol is in the same directory as the sheet, no path need to be given. If you have a sub-directory for symbols, only the sub-path needs to be specified, but you also can define the complete path. For example:

```
symbols\gearboxtab.sym
```

Creating SAT texts: Now create two SAT texts for the attachment points A and B using the tool . Make sure that these texts are part of the CPI group by making the SPS text created previously current before you create the new texts. Type the attachment point variable names into the text edit field and then place them at the ends of the dashed line (see [Figure 136, "Detail of CPI named Group" on page 154](#)).

Creating SCO texts: Finally, create four SCO texts using the tool . These texts contain ARG commands which assign values to the symbol variables. The exact position of these texts is not important, but they must be part of the CPI group. Make sure that you end the group when you have finished. For example:

```
ARG diam = 8  
ARG radius = 8  
ARG fillet = 4  
ARG len1 = 5
```

Checking CPI named groups: You can verify that all texts are contained within the CPI group by using the structure tree and *clicking left* on a CPI group. If you do so, all elements within the chosen CPI group become selected and they are displayed highlighted. This verifies that they are associated together in the same group.

Create the other CPI named groups: Once you have selected a CPI group, you can copy and paste it easily to define the other CPI groups. You then have to rearrange only the attachment points and adjust the values for the variables according to the current position of the tab. Create CPI named groups at all the other places where a symbol is to be loaded as shown in [Figure 135, "CPI named Groups in Position" on page 154](#). Use the variable values shown in [Figure 135](#).

Dimensioning the CPI named Groups


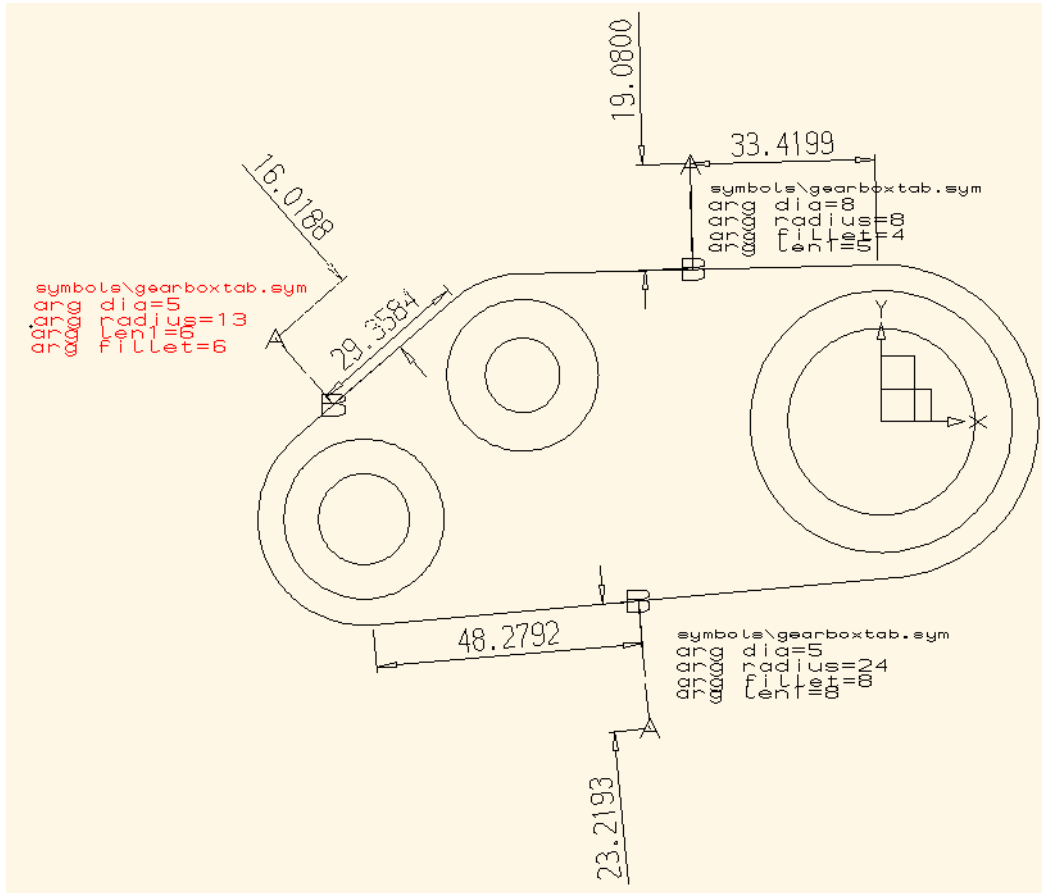


The symbol attachment points must be dimensioned so that they lie at grid intersections. [Figure 137](#) shows how to do this. Remember that the main dimensions have been omitted for clarity. The way that the attachment points are dimensioned is critical. In this example they are dimensioned from the tangent points of the arcs, easy to probe using the Near probe . They can be dimensioned in any way that you want, provided that the attachment point datums lie at grid intersections when the drawing is parameterized.

Figure 137 Dimensioning the Attachment Points



Check that attachment points lie at grid intersections with the tool . Draws the grid corresponding to the original drawing and undoes it . From now on, whenever the shape of the gearbox cover is changed, the attachment points will also move. This enables the symbols to adjust to the shape of the component.

Adding in-sheet Commands

Add an in-sheet command to delete all the dimensioning, symbol attachment points, supporting lines and in-sheet commands on the master drawing during loading parameterization. For example, `LAY 4 13 14 DEL.`

Parameterizing the Drawing


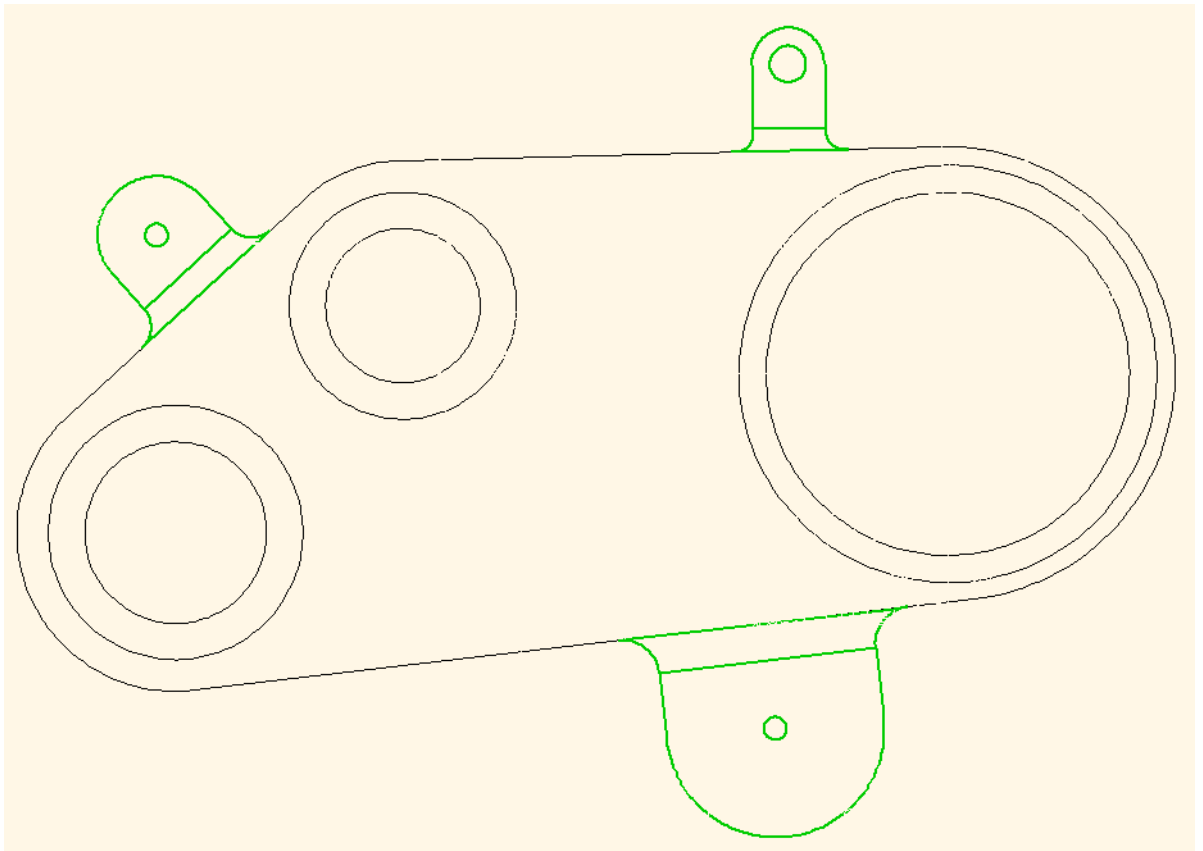
Finally, parameterize the drawing using the tool *Parameterizes the geometry then immediately undoes* . The result is shown in Figure 138. The gearbox cover itself is not parameterized because its dimensions have not been changed. The tab symbol is parameterized and loaded at the three prepared locations. The tabs are aligned with the attachment points and are drawn using the dimensions specified in the CPI named groups.

Figure 138 Result of temporary parameterization



Errors during Parameterization

If there is a problem during parameterization, an error message is written next to the SPS text of the CPI named group that is causing the problem. An error may be due to a problem with the CPI named group or with the symbol that you are trying to load.

Further Parameterization


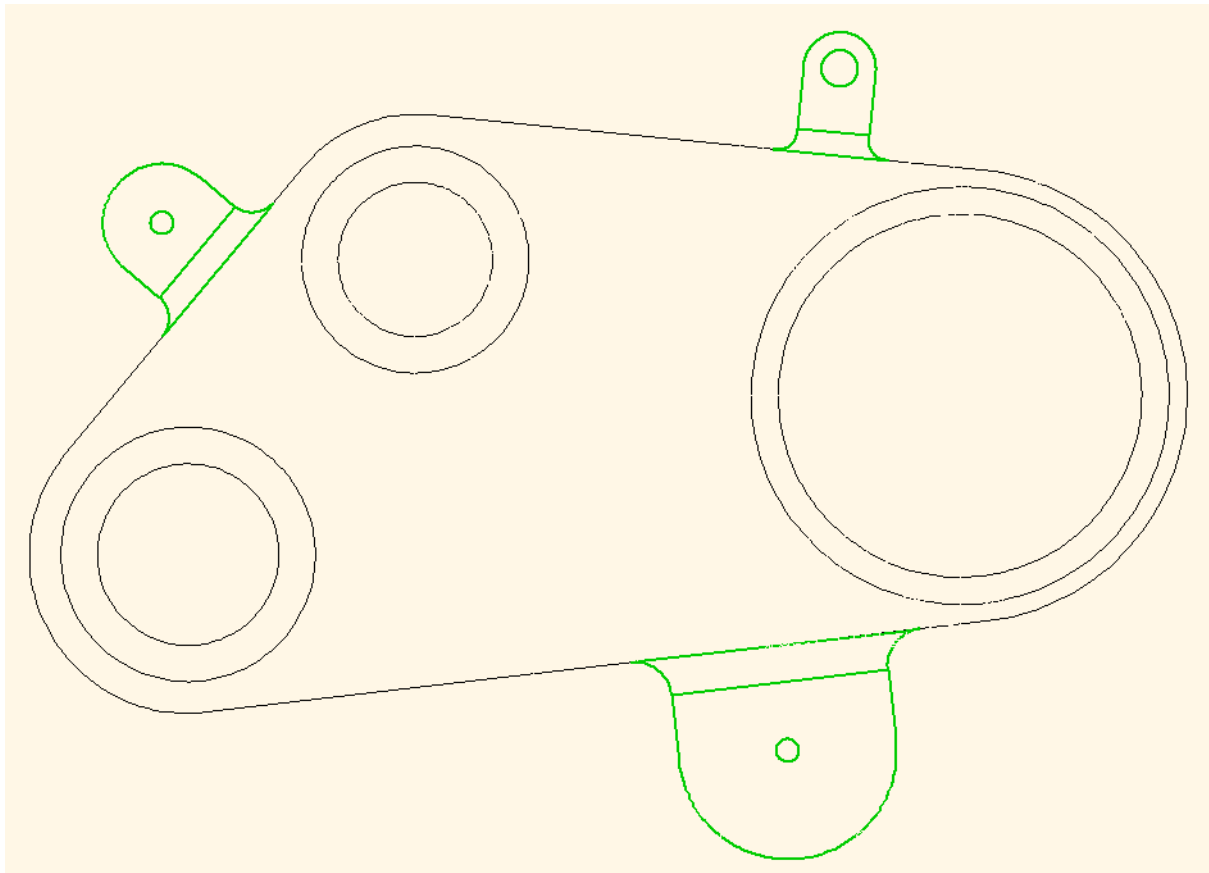
Refresh the screen with Refresh Graphics  to restore the original drawing. Now change the variable $d1$ on the gearbox cover to 30.0 mm instead of 15.0 mm and parameterize the component again. Figure 139 shows the result. This time the gearbox cover itself is parameterized, then the parameterized tabs are added to the sides. Notice that the tabs have moved with the sides of the gearbox cover.

Figure 139 Alternative Parameterization



PARAMETRIC GROUPS

This chapter shows how you can use parametric groups to isolate geometry where dimensions are to remain unchanged. This enables you to exclude some parts of geometry from parameterization.

- Introduction 160
- Creating a Parametric Group 162
- Example 1: Static Groups 163
- Example 2: Dynamic Group with One Prim 165
- Example 3: Dynamic Group with Two Prims 167
- Example 4: Rotating Parametric Groups 171
- Example 5: Dynamic Group with Three Prims 172

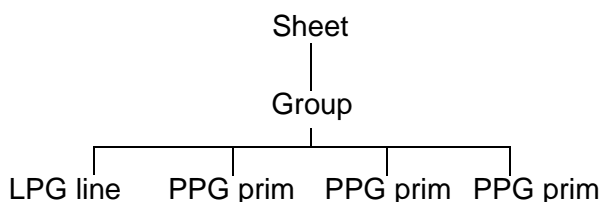
Introduction

Parametric groups provide a useful way of moving and scaling parts of a drawing without having to dimension every detail. The detail enclosed in a parametric group is either scaled or ignored during parameterization depending on the number of prims in the group.

Parametric Group Elements

A parametric group consists of an parametric group line (type LPG) and 0, 1, 2 or 3 parametric group prims (type PPG).

Figure 140 Structure of a Parametric Group



Static and Dynamic Groups

The simplest form of parametric group is a static group. This consists of a closed line of type LPG in a group, and does not include any parametric group prims. Undimensioned points inside a static group remain in their original position on the sheet during parameterization.

A parametric group which contains an LPG line and one or more PPG prims is known as a dynamic group. Points within a dynamic group can be moved, scaled, rotated, or differentially scaled depending on how many PPG prims the group contains.

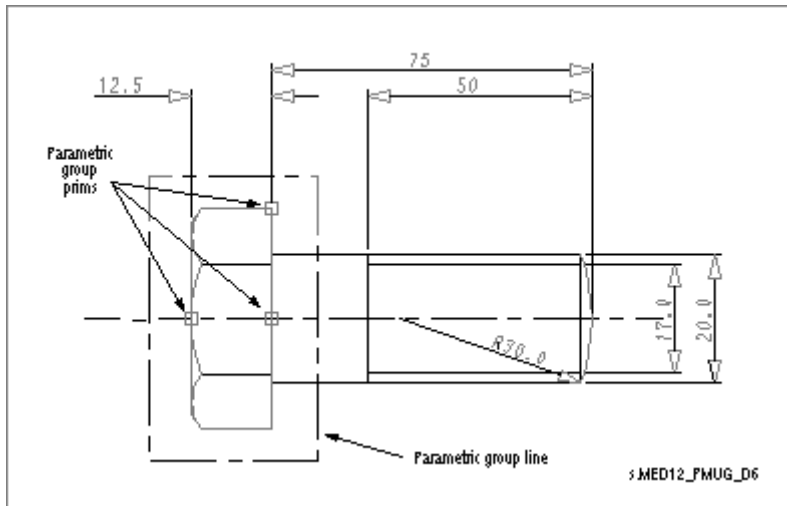
Points Within Parametric Groups

Normally when you prepare a drawing for parameterization, you must dimension every single point inside the parametric viewbox. So long as they are not required to move during parameterization, points within a parametric group do not have to be fully dimensioned. This can come in useful when you are parameterizing drawings containing detailed representations of springs, knurling, bolts, and so on. Points inside a parametric group that are dimensioned move as required during parameterization.

Example

The example in Figure 141 shows a bolt. Because the bolt head is included in a parametric group, the bolt head chamfer does not have to be dimensioned. When the drawing is parameterized, the bolt head will not be parameterized but scaled. The three small squares are parametric group prims. These are placed on the grid by dimensioning. During parameterization these prims will move to a position on the new grid: the rest of the points in the parametric group move will move in relation to the way the prims move.

Figure 141 Parametric Group With Three PPG Prims



Creating a Parametric Group

MEDUSA provides the following tools to create a parametric group:

Figure 142 Toolset for Groups



The tools from left to right are:

Creates parametric group lines

creates a line of style Parametric group line, type LPG.

Creates parametric group boxes

creates a box of style Parametric group line, type LPG.

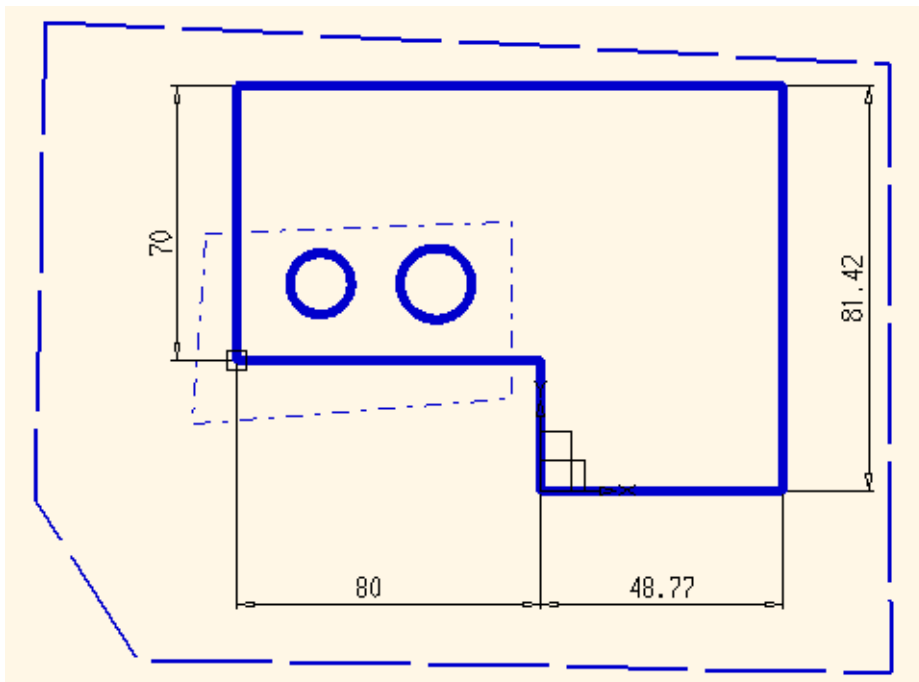
PPG Prim to create and control dynamic groups

creates a prim of style Para group datum, type PPG. Such a prim has to be a member of a parametric group. Up to three prims can be placed for one group.

To create a group:

1. Open a new group.
2. Draw a parametric group line (type LPG) around the detailed parts that you do not wish to parameterize.
3. If necessary, add up to three prims (type PPG) to the parametric group.
The following example shows a geometry with one group line and one prim. The prim determines that the holes keep their position relative to the segments close to them.

Figure 143 Example of Preventing Parameterization with Group Line and Prim



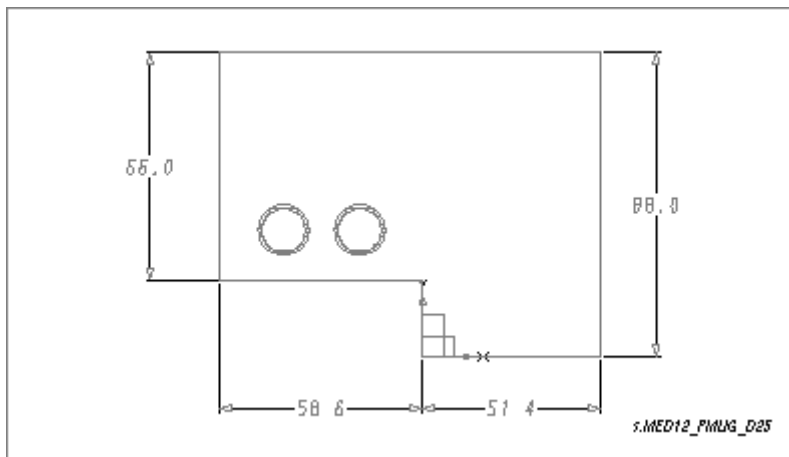
Example 1: Static Groups

In this example, the size of the component changes, but the size of the bolts remains constant. Figure 144 shows a component which has two bolts. You may want the bolt heads to remain completely static or require them to move together with the main component when the drawing is parameterized. You can do this without having to dimension the bolts in any detail by using a parametric group.

Creating the Component

Draw the outline of the component in Figure 144 and then draw the bolt heads and washers. Use a prim for the reference point. When you have finished drawing the component, dimension it.

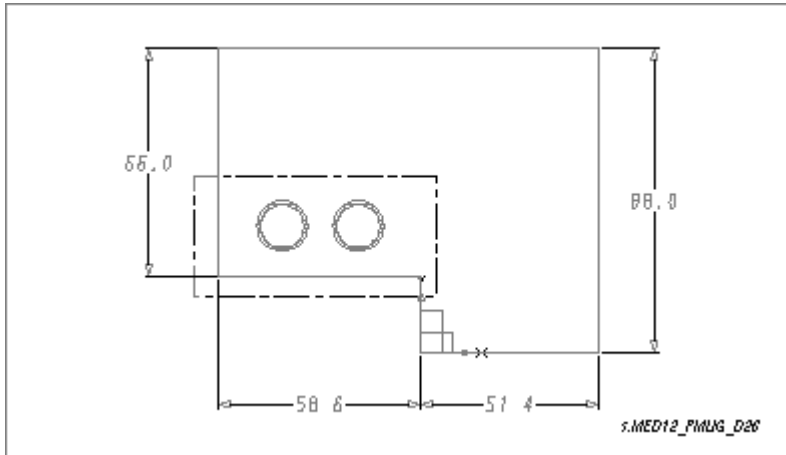
Figure 144 Component with Bolts



Creating a Parametric Group

Begin by opening a new group. This group will contain the parametric group line, a closed line of type LPG. The line must be part of the group. Draw a closed LPG line around the bolts, as shown in, Figure 145 and then end the group.

Figure 145 Adding a Parametric Group Line



Parameterizing the Component


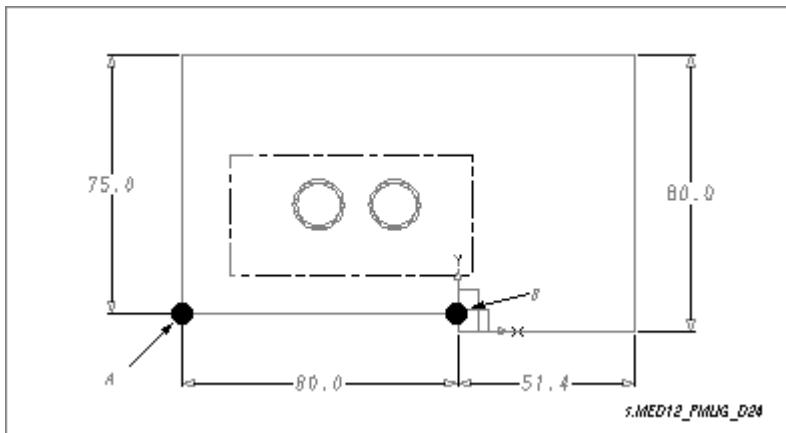
Change some of the dimensions of the component and then parameterize the drawing using the tool . Figure 146 shows how the outline of the component is changed according to the new dimensions but the bolts remain in their original position.

Figure 146 Result of Parameterization



Note that the bolts are neither parameterized nor repositioned. However the outline is changed, all undimensioned points in the group remain stationary. The points in the group that are dimensioned (the two points A and B forming part of the outline of the main component) are parameterized normally.

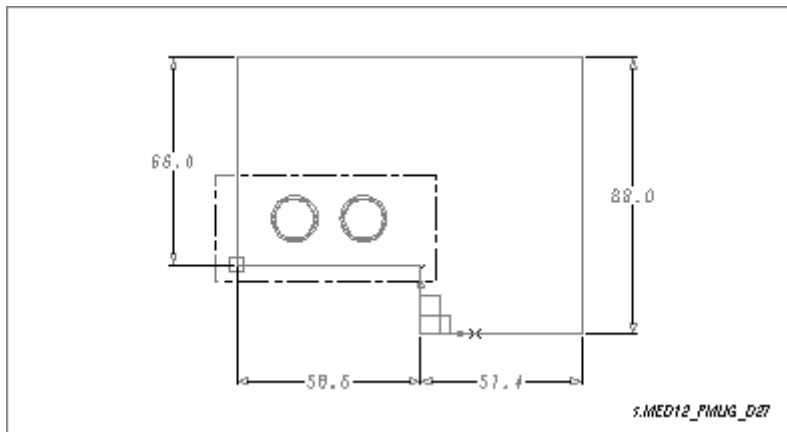
Example 2: Dynamic Group with One Prim

In this example, a PPG prim is used with a parametric group line to move the undimensioned points in the group with the component as it is parameterized. The prim links the objects inside the group line to the parametric grid. By placing the PPG prim at a grid intersection, all undimensioned points inside the group will now move in exactly the same way as the prim during parameterization, although they are neither scaled nor parameterized.

Change the Drawing

Redraw the sheet and restore the original dimensions, as shown in [Figure 144, “Component with Bolts” on page 163](#). Now make the parametric group line current and then create a prim of type PPG. The prim looks like a small square. By making the group line current before creating the prim, you ensure that the prim is part of the same group as the line. Place the prim inside the parametric group at the point shown below.

Figure 147 Prim Placed Inside Parametric Group



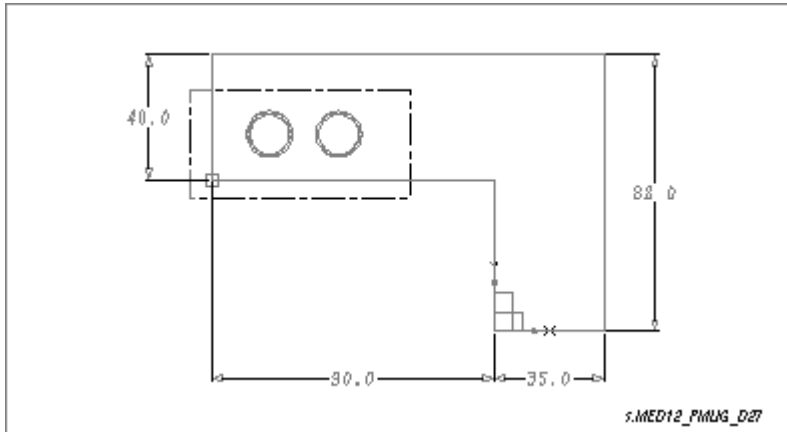
The prim must be placed at an intersection of grid lines, as in Figure 147, otherwise Parametric Design returns the following error message when you try to parameterize the drawing:

```
Point not dimensioned
```

Parameterizing the Component

Now replace the original dimension text with the new parameters shown in Figure 148 and then parameterize the component. This time the bolts move in exactly the same way as the prim. The PPG prim has moved up and left. So have the bolts. Again notice that, although the bolts have moved, they have not been changed in size.

Figure 148 Change of Dimensions after parameterization



Experiment by changing the dimensions and then parameterizing the drawing again. The group always moves in unison with the prim.

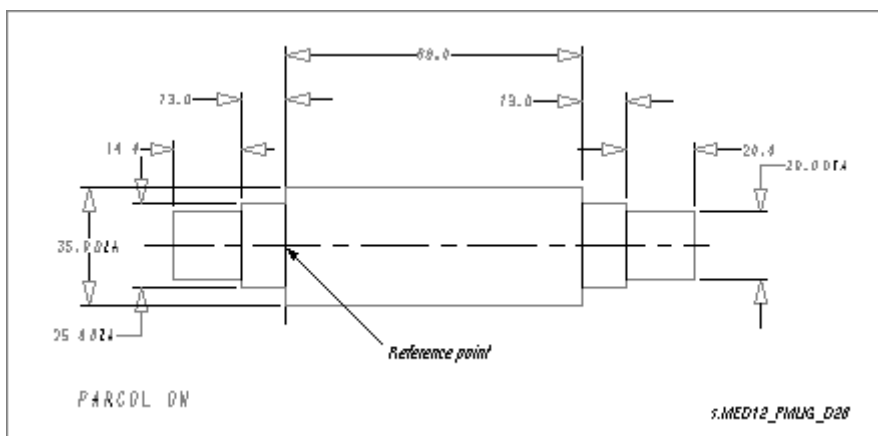
Example 3: Dynamic Group with Two Prims


In this example you will see how two prims can be used with a parametric group to scale the undimensioned points in the group.

Drawing the Component

Draw and dimension the component shown in Figure 149. Create the horizontal 20.0 mm dimension on the right side of the shaft separately from the other horizontal dimensioning. This allows it to be removed in a later operation.

Figure 149 Dimensioned Component

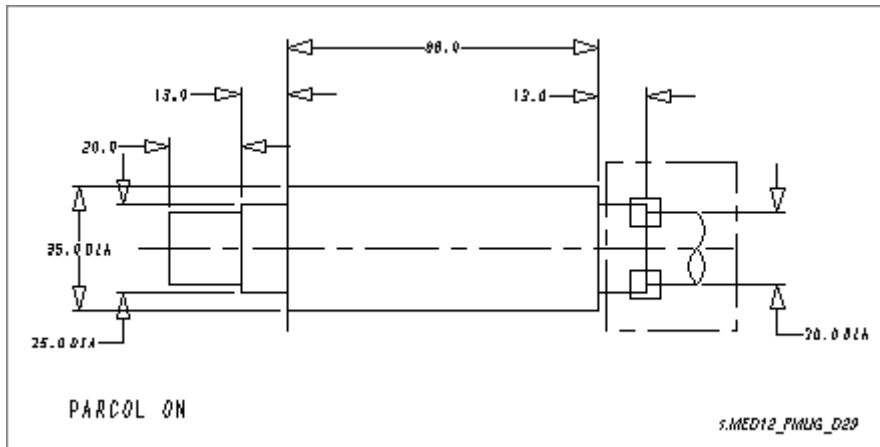


Use intersecting static baselines to specify the reference point. Add the in-sheet command PAR COL ON. Remember that the PAR COL switch that can be turned on so that grid lines are drawn along lines which are collinear but which do not overlap (refer back to "Layers and Parametric Switches", "The PAR COL switch" on page 114 for a description of the PAR COL switch). Using PAR COL here saves having to dimension the 20.0 mm shaft diameter twice, once on each side of the component. Draw the grid and then test the drawing with the tool Parameterizes the geometry then immediately undoes .

Changing the Geometry

Now change the right-hand end of the shaft so that it represents an interrupted view, as shown in Figure 150. The end of the shaft is constructed by drawing three elliptical arcs. Remove the 20.0 mm horizontal chain dimension on the right side of the component.

Figure 152 Addition of Parametric Group Line



In Figure 152 you can see that the shaft on the right is dimensioned. This means that the shaft will be parameterized according to how the 20.0 mm diameter dimension changes. The only undimensioned points in the parametric group are the points that form the elliptical arcs. During parameterization, these points will move in the same way as the prims move. They will be scaled, but not parameterized.

Parameterize the Component


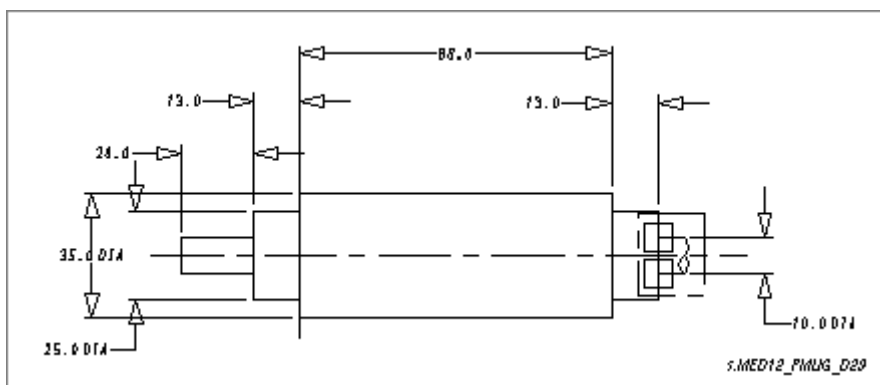
Now change the 20.0 mm dimension text on the interrupted shaft dimension to 10.0 mm and then run the tool . The result is shown in Figure 153.

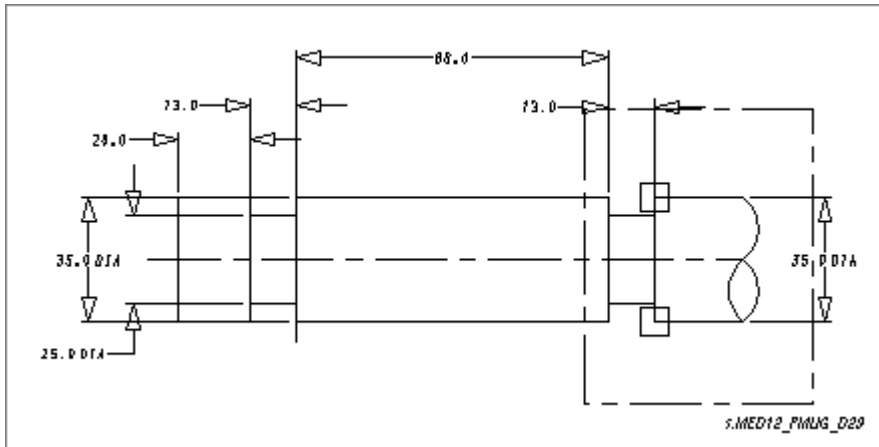
Figure 153 Parameterized Component



As expected, the diameter of the shaft changes according to the new dimension. If you look closely, however, you can see that the end of the shaft has been shortened, and that the elliptical arcs have been scaled down in proportion. It is important to grasp that the diameter of the shaft is the only part of it that has been parameterized. The rest of it has been scaled to match the new positions of the prims. The prims, which were 20.0 mm apart, are now 10.0 mm apart. This means that all undimensioned parts of the group have been scaled by a factor of 10/20. In other words, the parametric group has been scaled down to half size.

Now change the diameter of the shaft to 35.0 mm and parameterize it again. Figure 154 shows the result. The shaft has been scaled by a factor of 35/20. The length of the shaft is changed by the same ratio.

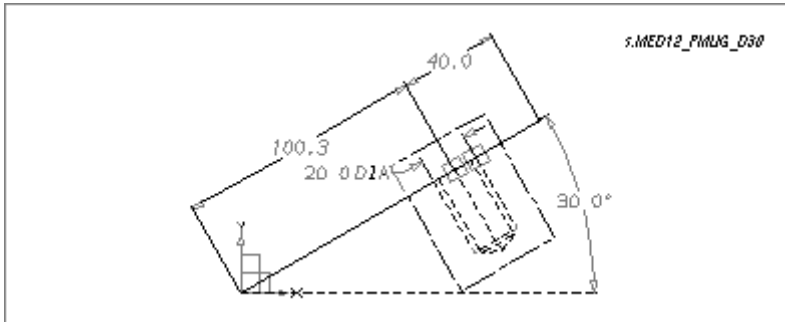
Figure 154 Parameterized Component



Example 4: Rotating Parametric Groups

Using two prims, you can ensure that a parametric group rotates with the rest of the geometry if necessary. This example illustrates how a bolt hole can be rotated in this way. In the component shown in Figure 155, one prim lies on the center line of the bolt hole and the other is positioned so that it marks the width of the hole.

Figure 155 Bolt Hole With Parametric Group

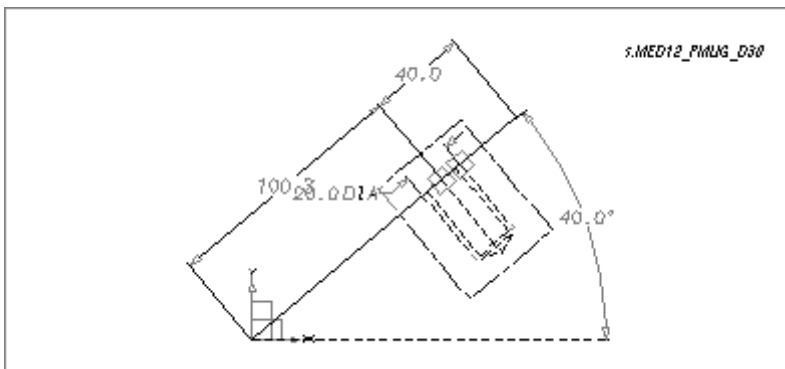


The second prim ensures that the parametric group rotates if the main component rotates. The 20.0 mm hole dimension serves two purposes:

- It allows the diameter of the hole to be changed during parameterization.
- It places the second prim at a grid intersection, allowing the bolt hole to be rotated.

The undimensioned points in the group will move in the same way as these prims move in relation to each other, as shown in Figure 156:

Figure 156 Rotating a Parametric Group



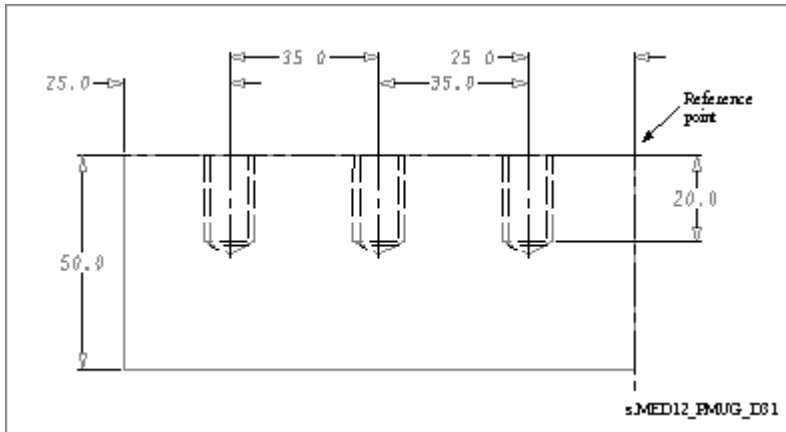
When the 20.0 mm dimension is changed, the length of the bolt hole is changed as well as the width. For example, if the width is changed from 20.0 mm to 30.0 mm, the length of the bolt hole is scaled accordingly. Its new length is 30/20 of its original length. You can set the length of the bolt hole independently from its width by using a third prim. This is explained in the next section.

Example 5: Dynamic Group with Three Prims

In this example you will see how to parameterize a more complex component without having to dimension all the intricate details, such as bolt holes.

Figure 157 shows a component that contains three bolt holes. Using three prims, you can differentially scale the parametric group and set the length and width of each bolt hole independently.

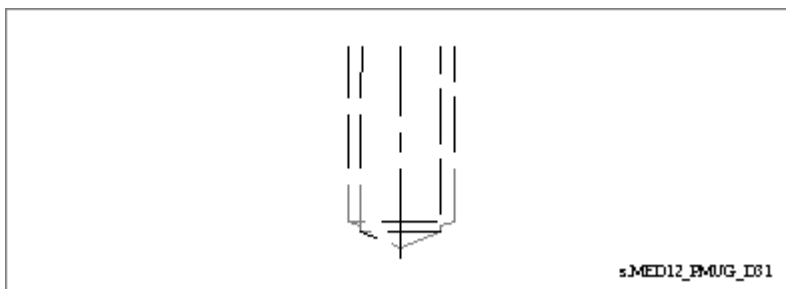
Figure 157 Component With Bolt Holes



Drawing the Component

Begin by drawing and dimensioning the component in Figure 157. The reference point is fixed by two static baselines that intersect at the top right corner of the component. The dashed vertical line through the center of each bolt hole (shown more clearly in Figure 158) enables you to dimension the holes from the reference point.

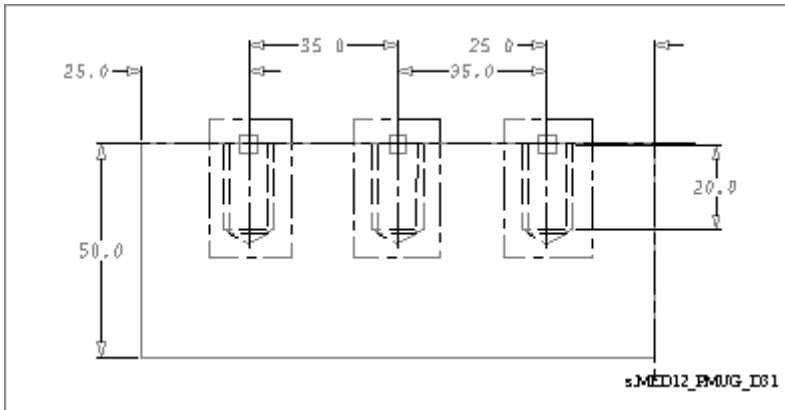
Figure 158 Detail of Bolt Hole from Figure 157



Creating Parametric Groups

When you have drawn and dimensioned the component, draw a parametric group line around each hole, as shown in Figure 159. Remember to create a new group for each parametric group. At this stage add a single PPG prim to each group.

Figure 159 Parametric Group Line

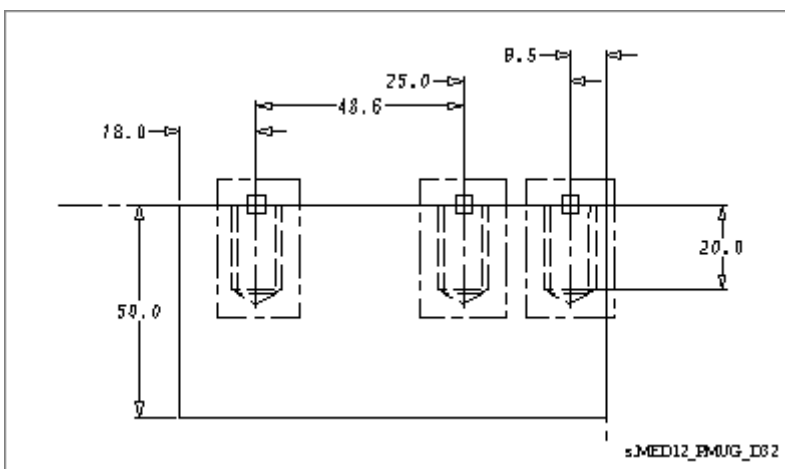


Test the drawing to make sure that it will parameterize. If there are no problems replace some of the dimensions with new parameters.

Parameterizing

Parameterize the drawing. Figure 160 shows one parameterized version. Notice how the bolt holes have moved.

Figure 160 Parameterized Drawing



Adding More Prims

The next step is to add two more prims to each parametric group, as shown in Figure 161. Remember to open the group before creating new prims for each group. Add the 12.0 mm dimensions. This enables you to set the diameter of each hole when the drawing is parameterized. Finally, add an in-sheet command PAR COL ON. This ensures that the grid line created by the 20.0 mm dimension at the right side of the component extends to the bottom of the other bolt holes (which are collinear). You can now change the depth of each hole during parameterization.

Figure 161 Adding More Prims

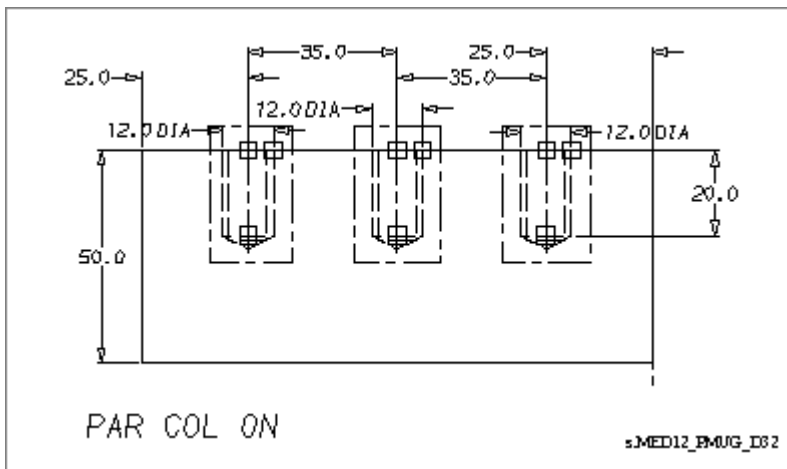
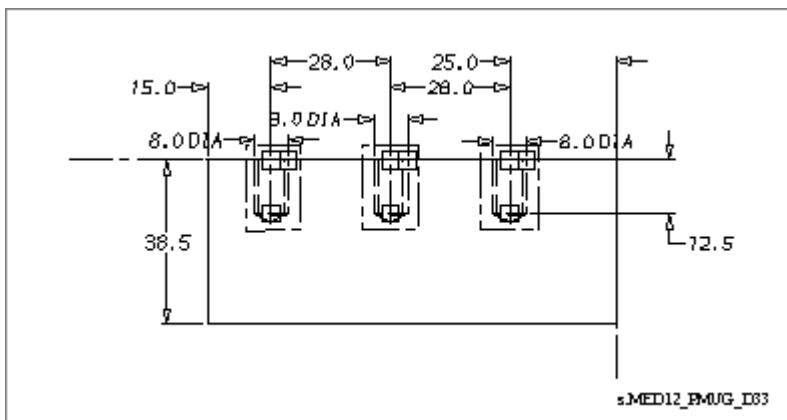


Figure 162 shows the result of parameterization. Note that the depth, diameter, and positions of the holes have been changed.

Figure 162 Parameterized Drawing



POST-PARAMETERIZATION

This chapter shows the functionality of 2D Parametrics which cannot be done with parameterization itself but afterwards therefore it is called Post-Parameterization.

- [Overview](#) 176
- [Dialog.....](#) 177
- [Creating Post-Parametric Definition Sheets](#) 180
- [Callbacks](#) 182
- [Example](#) 186
- [Demo Sheet.....](#) 188

Overview

The purpose of the post-parametric module is to create families of parts by applying user parameters to procedures attached to MEDUSA graphics groups.

The callback functionality may be used on its own or as an extension to the MEDUSA Parametric module, for example to arrange multiple copies of a parametric part into an array pattern.

Callback definitions are saved with the sheet.

Precondition for post-parameterization is that geometries are assembled in groups.

Dialog


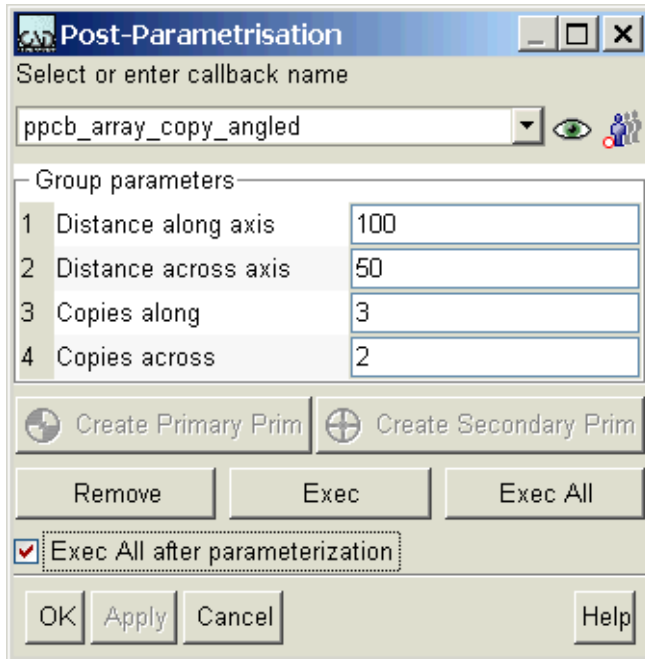

To bring up the Post-Parametrization dialog press the button Display Post Param Dialog  in the Parametrics tooltray.

Figure 163 Dialog Post-Parameterization



Consider the following things for this dialog:

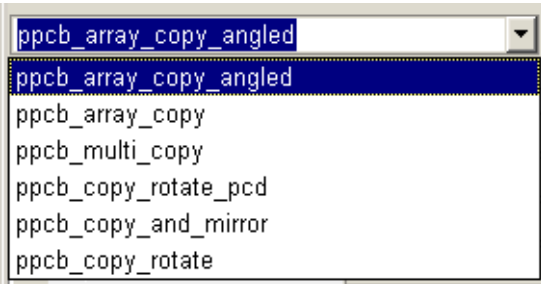
- For this dialog groups need to be selected as a whole and during partial selection the dialog will behave as if there is nothing selected.
- When the dialog is first displayed the current selection is auto-escalated to a sheet level group.
- Also note that during its life the dialog modifies the current selection tool to Selects sheet-level named groups .
- If the currently selected elements belongs to a group, the group becomes selected automatically and the first callback for this group is displayed


The options and buttons on the dialog are:

Select or enter callback name


In the pulldown menu below this entry you see a list of internal graphics-modifying procedures (callbacks) available to assign them to groups. For details see [“Callbacks” on page 182](#).

Figure 164 Callbacks available for Post-Parameterization



Highlight all groups on sheet with this callback 

Press this button next to the pulldown list to highlight all groups on the current sheet for which the shown callback is defined.

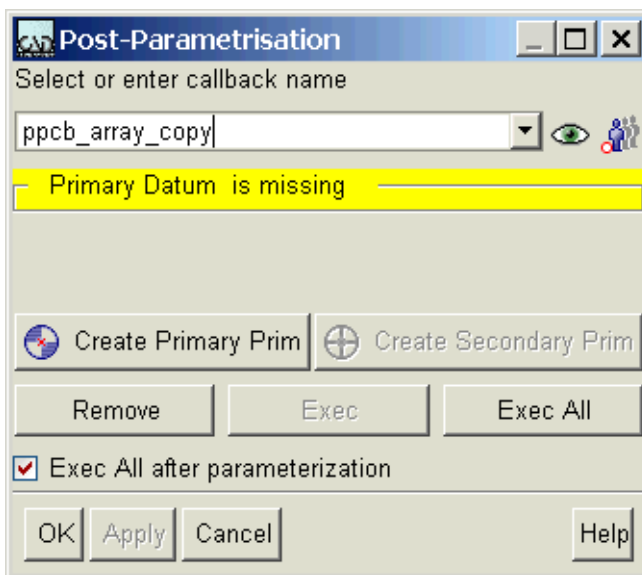
Selects sheet-level named groups 

If the current selection tool is not Selects sheet-level named groups, press this button to switch back to the tool Selects sheet-level named groups.

Create Primary/Secondary Prim

Some of the callbacks may require up to two datum prim. Use these buttons to create these prim. After the required prim have been created the tools are disabled. If a datum prim is missing an error message with yellow background appears in the dialog.

Figure 165 Message for missing primary datum prim



Remove

deletes the callback stored on the selected group. After executing callbacks they are removed automatically from the group in order to prevent executing them several times.

Exec

executes the parametric procedure stored on the current group. If you changed parameters this button is disabled until you Apply the new settings.

Exec All

executes the procedures stored on all groups in the current sheet.

Exec All after parametrization

If this option is switched on, after parameterization all post-parametric procedures will be run automatically.


OK, Apply

These buttons apply the settings of the current callback. If you use `Apply` the button will be disabled and the dialog remains open. `OK` closes the dialog.

Creating Post-Parametric Definition Sheets

Before defining a callback function for a certain geometry group, we recommend to look through the available callbacks. If no group is selected, inside the Post-Parameterization dialog the list Group parameters is disabled. Use this mode to browse through the available callbacks. The browse text of the selected callback show you a short description.

Procedure

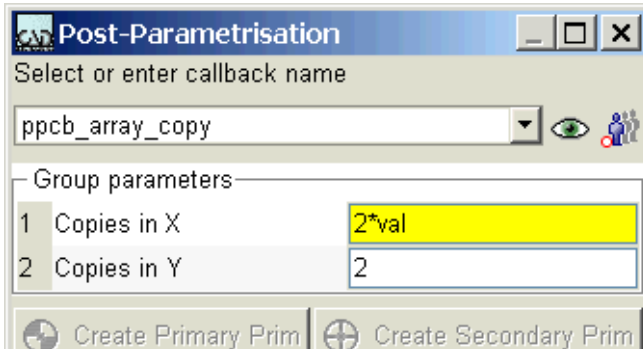
1. Open the Post-Parameterization dialog with the button Display Post Param Dialog  in the Parametrics tooltray.
You can keep the Post-Parameterization Dialog open while switching between sheets or changing the sheet selection.
2. Select a sheet level group.
3. Pick a callback.
Instead of selecting from the list you can also type the name of the callback.
4. Enter its parameters and press Apply.
Parameters can be certain values, variables and expressions.
5. After the parameters have been assigned you can press Exec to test-run the callback.
6. Undo the execution because otherwise the callback definitions cannot be stored with the sheet.
7. Repeat the procedure for any group of geometry which needs to get a callback definition.
8. If you are finished, save the sheet to store the callback definitions with the sheet information.

Please note: Consider that a group can get only one callback.

Possible Errors

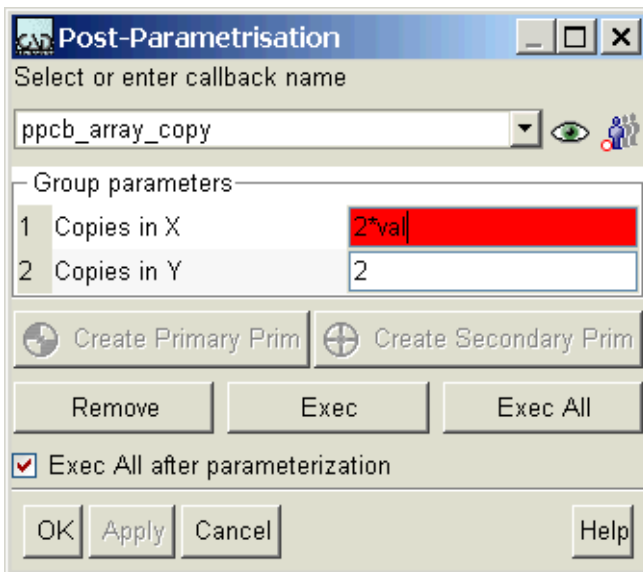
All expressions are evaluated at the time of entry and invalid ones are shown in yellow. The browse text of a parameter expression will show you its current value. If the expression cannot be evaluated the browse text displays Can't Evaluate.

Figure 166 Example for an invalid parameter after entering



You should note that at the time of execution a once-valid expression could turn out to be invalid. In this case the offending expression will be shown in red.

Figure 167 Example for an invalid parameter after running callback



Callbacks

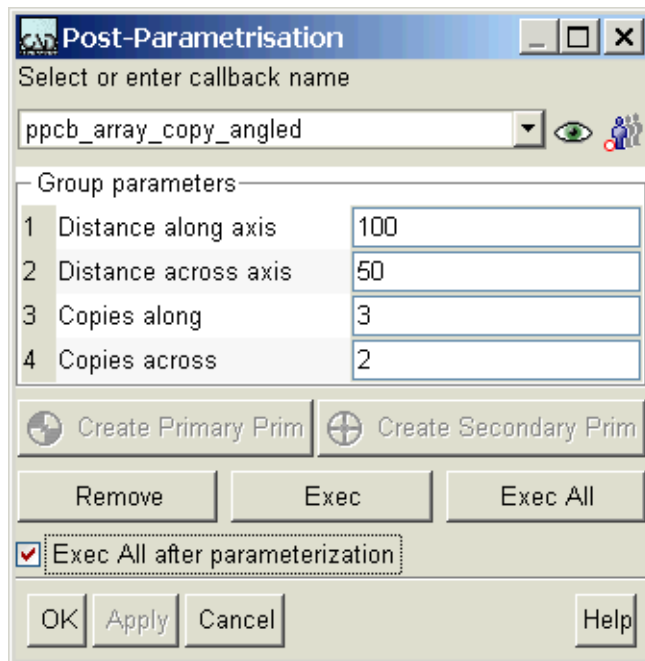
This section gives all the callbacks distributed with the MEDUSA 2D Parametrics module.

The function of a callback is explained in a browse text, which appears after selecting a callback and keeping the cursor on the edit field for a moment. The following sections give you some more information. In addition see the section “[Dialog](#)” on page 177.

ppcb_array_copy_angled

This callback copies the geometry of the current group into the direction from first to second prim (in the dialog it is called *Distance along axis*) and perpendicular (*Distance across axis*) to this direction. The distance and number of copies is defined by the specified values for *Copies along* and *Copies across the axis*.

Figure 168 ppcb_array_copy_angled

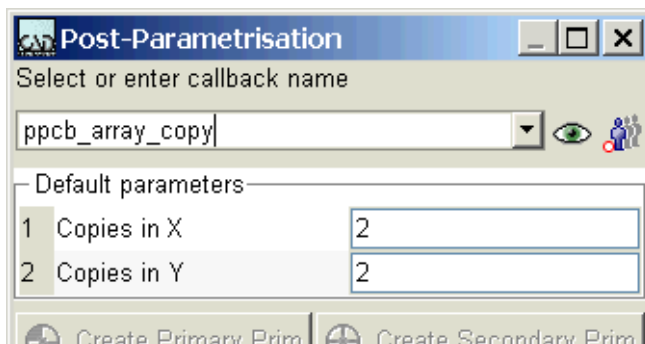


ppcb_array_copy

This callback copies the geometry of the current group in x- and y-direction. The virtual line defined by the prims specifies the direction of copying as well as the spacing between the copies in horizontal and vertical direction. The number of copies is defined by the user and it can be different in x- and y-direction. See also “[Example](#)” on page 186.

Please note: The values defined for Copies in X and Copies in Y have to be increased by one. So the final number of geometries in the array is: $(\text{Copies in X} + 1) \times (\text{Copies in Y} + 1)$. For example, if both values are two, the number of geometries in the array is nine.

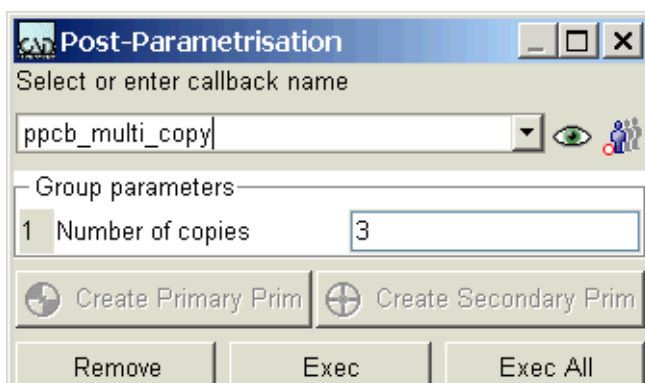
Figure 169 ppcb_array_copy



ppcb_multi_copy

This callback copies the geometry of the current group into the direction from first to second prim. The spacing between the copies is defined by the distance between the prims. The user obly defines the Number of copies in the dialog.

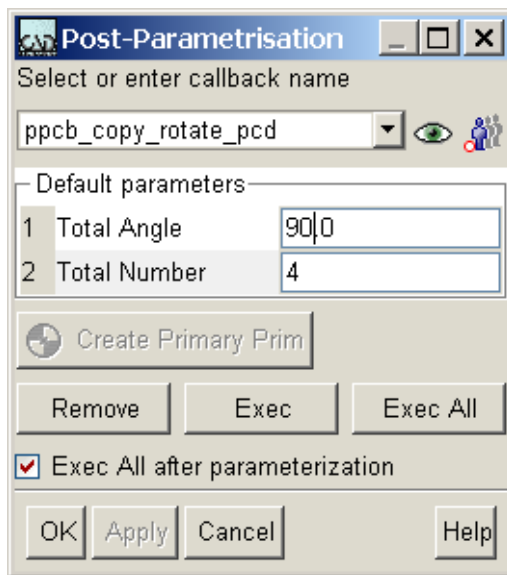
Figure 170 ppcb_multi_copy



ppcb_copy_rotate_pcd

This callback copies the selected geometry *Total Number* times on a circular line of *Total Angle* degrees whose center is defined by the prim. The geometries are placed even on the circular line.

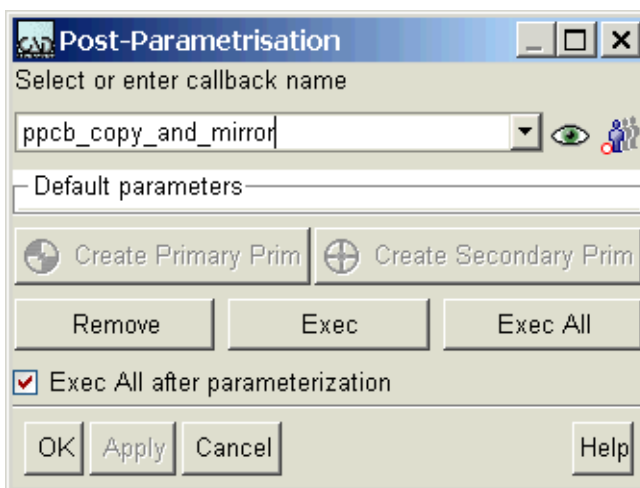
Figure 171 ppcb_copy_rotate_pcd



ppcb_copy_and_mirror

This callback mirrors the geometry at the virtual line defined by the two prim.

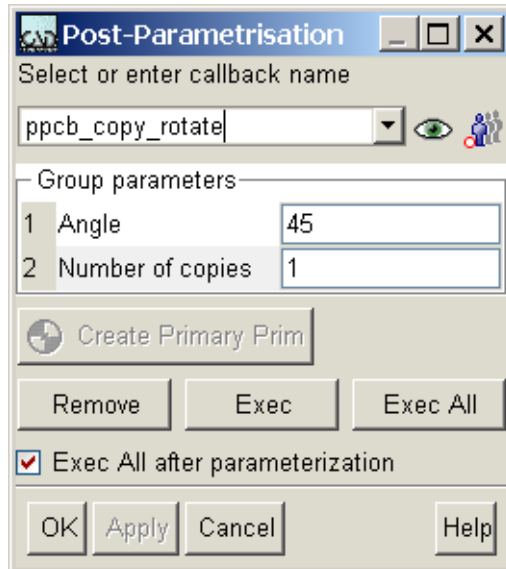
Figure 172 ppcb_copy_and_mirror



ppcb_copy_rotate

This callback copies the selected geometry *Number of copies* times on a circular line whose center is defined by the prim. The geometries are placed every *Angle* degrees on the circular line.

Figure 173 ppcb_copy_rotate



Example

The following images illustrate the effect of the procedure `ppcb_array_copy` applied with parameters `Copies in X = 4`, `Copies in Y = 2`. The original graphics will be copied 5 times in the X direction and 3 times in the Y direction using the displacement between the 2 prims as spatial increment.

Figure 174 Example: Parameter Settings in the Dialog Post-Parameterization

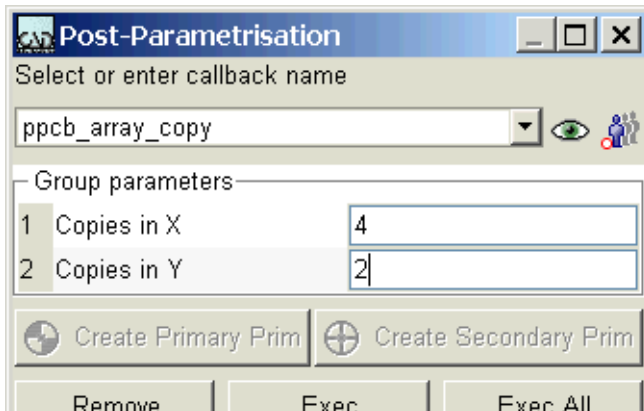


Figure 175 Example: Before Parameterization

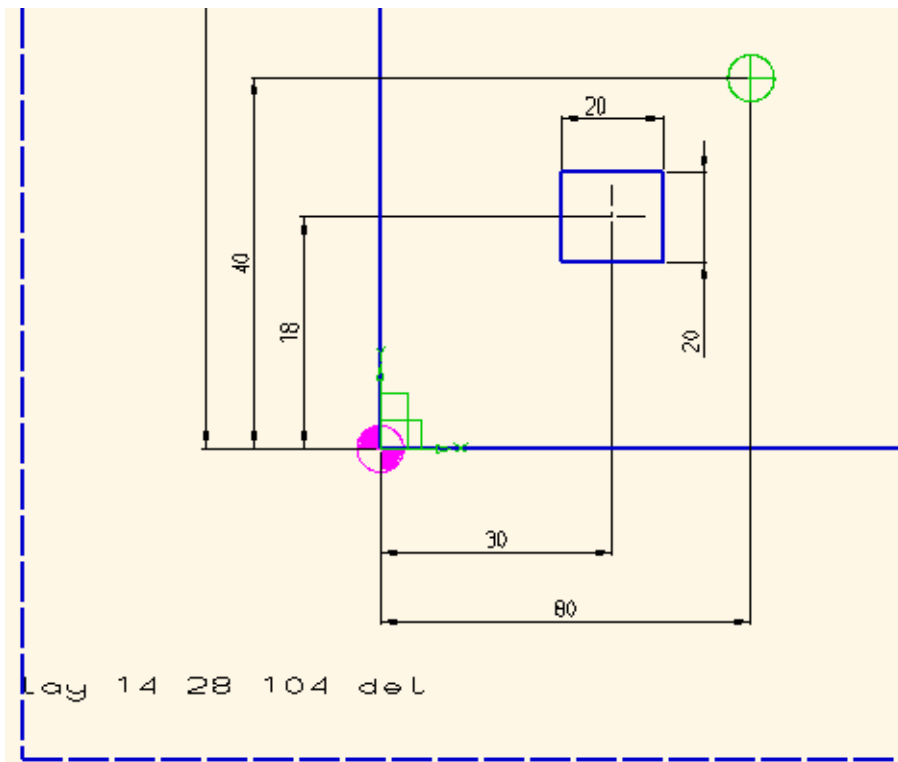


Figure 176 Example: After Parameterization

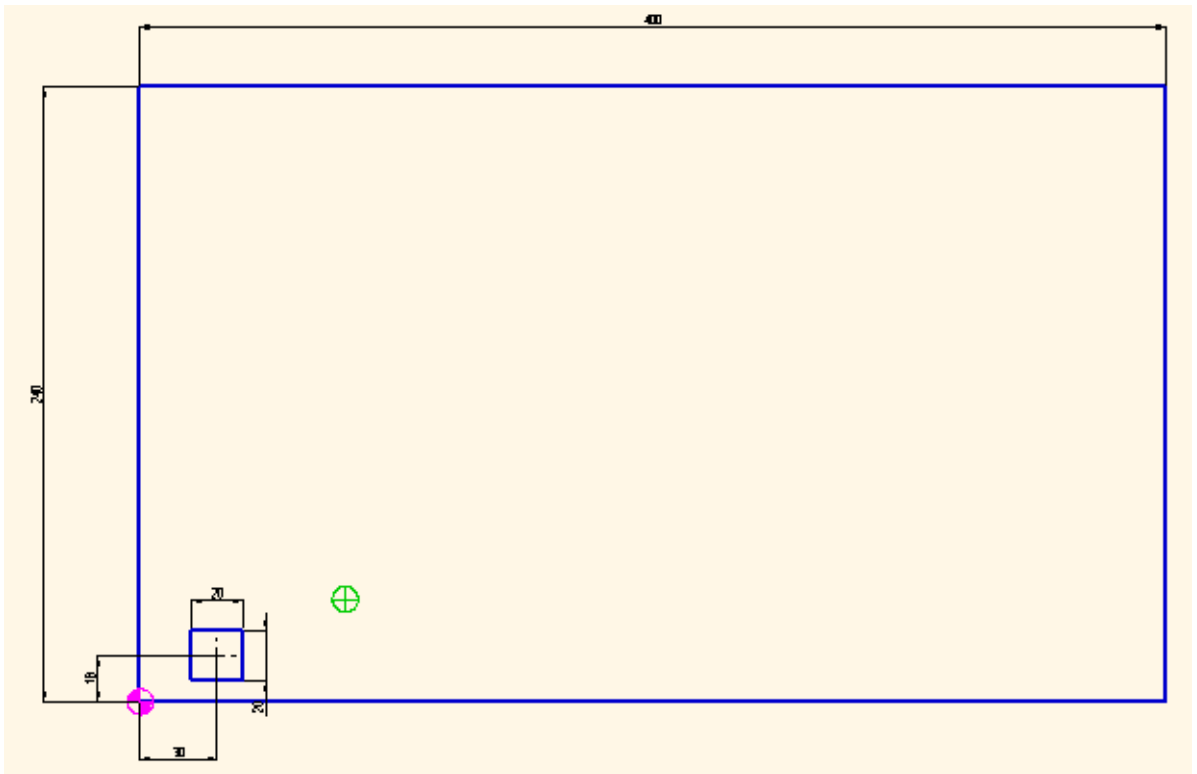
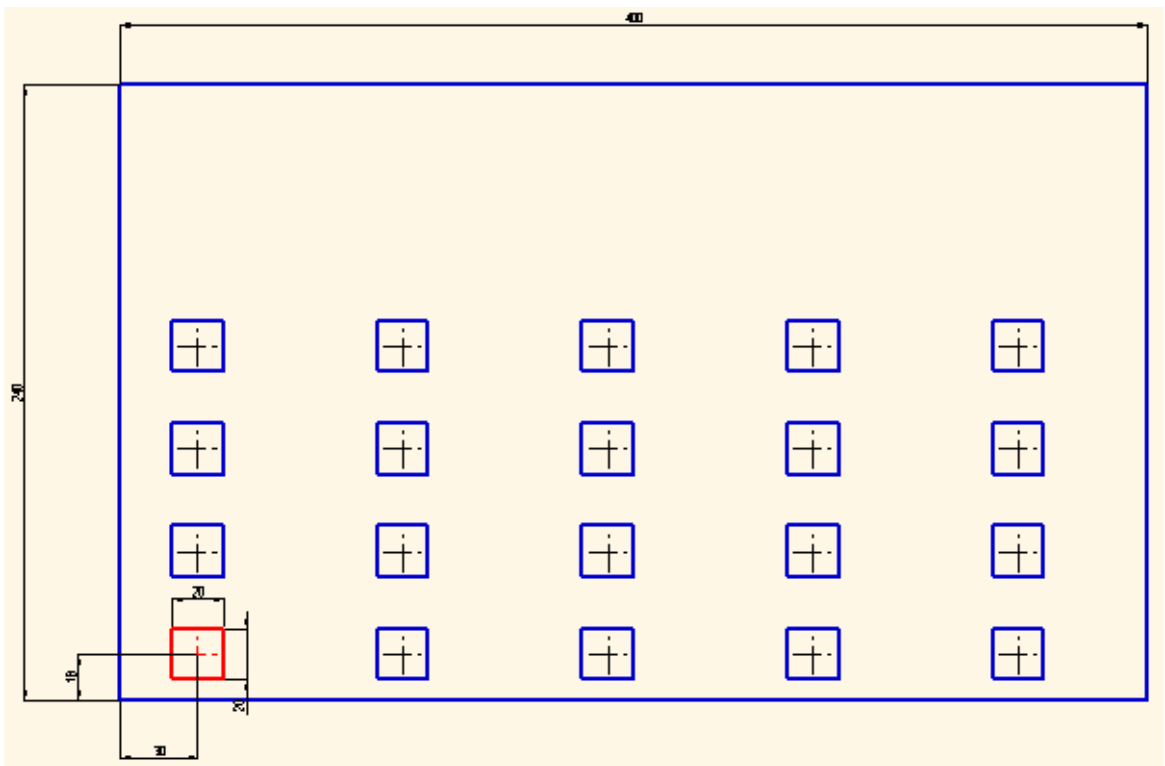


Figure 177 Example: After Running Post-Parameterization



Demo Sheet

To test the Post-Parametric functionality you can use the demo sheet *postparam.she*. This sheet is stored in *<installation directory of MEDUSA>\MEDPARA\M2D\DEMOS*. It contains several groups of geometry and each of them runs with a callback procedure.

After loading the sheet, open the Post-Parametric Dialog and press `Exec All`. Full Undo/Redo is available.

SIMULATING MECHANISMS

Using MEDUSA Parametric Design, you can simulate the movement of parts or objects by repeatedly parameterizing selected dimensions. A mechanism is redrawn on the screen at regular stages, enabling you to investigate potential clashes between different parts of the mechanism. This technique relies upon repeating a sequence of commands which increase or decrease the value of variables between repeated parameterization operations.

This chapter contains examples for you to try out.

- Introduction 190
- Dialog..... 191
- Example 1: Repeated Parameterization 193
- Example 2: Simulating Linear Motion 195
- Example 3: Simulating Rotary Motion..... 197
- Example 4: Using a Program to Simulate Motion 200
- Example 5: Simulating a Working Mechanism..... 202
- Plotting 204

Introduction

To simulate the movement of a mechanism, you draw the geometry and set up a parametric grid using a prim or intersecting baselines and valid dimensions as if you were preparing for normal parameterization. Enclose parts of the object that will not move within a parametric group, or leave their dimensions unchanged. Where dimensions will change, replace the text with a variable.

Please note: Remember to make your viewbox large enough to allow the movement you expect to see.

Repeated Parameterization

By repeating a sequence of commands which increment a controlling variable and parameterize the object temporary, you can produce several parameterized versions of one drawing. To execute a sequence of commands to parameterize an object repeatedly, use one of the following:

- The graphical user interface, Mechanisms dialog, described in [“Dialog” on page 191](#).
- A program. Using programs to simulate movement, described in [“Example 4: Using a Program to Simulate Motion” on page 200](#).

Points to Consider

When you use Parametric Design to simulate mechanical movement, ensure that:

- All parts of the mechanism are inside the parametric viewbox at all times
- The mechanism is adequately dimensioned and tested before attempting to simulate movement
- Dimensions which are to change are represented by variables or expressions containing functions of variables
- Variables are given an initial value before any parameterization begins
- Variables are incremented each time the drawing is parameterized
- For Bacis1 programs consider that the PARS command is cancelled each time the drawing is parameterized

Dialog


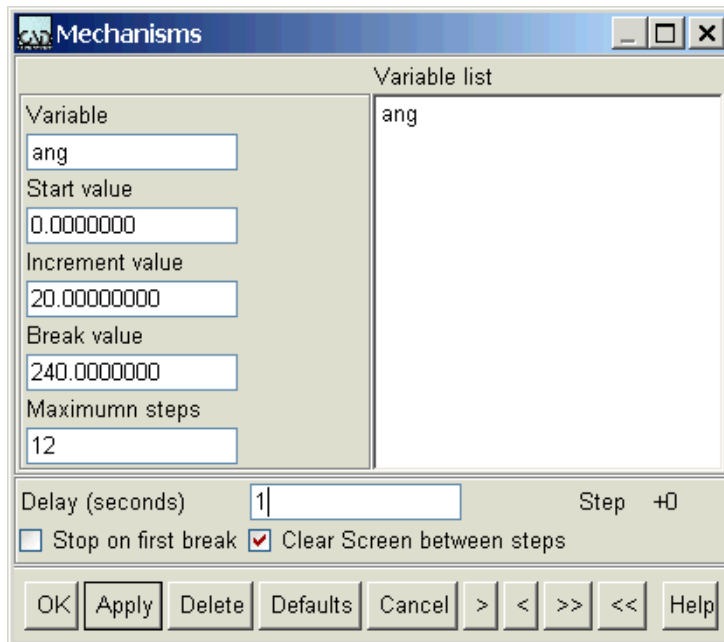
For simulating a mechanism MEDUSA provides the dialog *Mechanisms* which can be opened with the tool *Simulates mechanism motion*  available inside the toolset for parameterization.

Figure 178 Dialog for Running Mechanisms



The **variable definitions** are:

Variable

is the name of a variable defined for the geometry you want to simulate. For setting the values for a variable type its name, define the values and then choose the button *Apply*.

Start value

is the first value to which the variable is set.

Increment value

is the value by which the current value is increased.

Break value

is the value at which simulation stops.

Maximum steps

is the number of parameterizations which are done per simulation.

Variable list

gives all variables you defined for simulating the geometry. The values on the left are given for the selected variable.

The **options** are:

Delay (seconds)

is the time between the single steps. By default the edit field is empty.

Step is the current step number (for information only).

Stop on first break stops simulation if an error occurs. By default this option is off.

Clear screen between steps
If this option is off (default), the result of each parameterization is drawn on the screen. so you get parts of the parameterized geometry one over each other. To clear the screen after each parameterization, switch on this option.

The **buttons** are:


OK closes the dialog. The current settings are stored for later usage.

Apply uses the current settings. After each run of simulation you have to Apply the settings again in order to rewind the sequence to its beginning (see currently displayed Step number)

Delete removes the currently selected variable from the Variable list.

Defaults replaces the values of the current Variable with the default values.


Cancel closes the dialog.

Buttons for running simulation 

The left buttons run one step of the simulation, forward or backward.




The right buttons run the whole simulation, forward or backward. until the number of Maximum steps or until the first break point.

The buttons for running simulation backward do work as recently as simulation was done forward.

Each parameterization step accords with the singular usage of the tool Parameterizes the geometry then immediately undoes . So, if you redraw the graphics the parametrization result disappears and the original drawing is shown again.

Help opens the appropriate page in the online documentation.

The following steps show you how to run a simulation:

1. Draw the geometry and dimension it.
2. While dimensioning use the tool Draws the grid corresponding to the original drawing and undoes it  for checking the grid.
3. If the geometry is finished replace dimension values with variables or expressions.
4. Choose the tool Simulates mechanism motion  for opening the Mechanisms dialog.
5. Define variables and insert their values. Specify options if you want to use them.
6. Apply your settings.
7. Run simulation with the buttons .

Please note: Simulation settings are valid for the current MEDUSA session only.

Example 1: Repeated Parameterization

This example shows how to use a text variable command to repeatedly parameterize an object. No movement is simulated in this example but in the next example you will use the same technique to move an object across the screen.


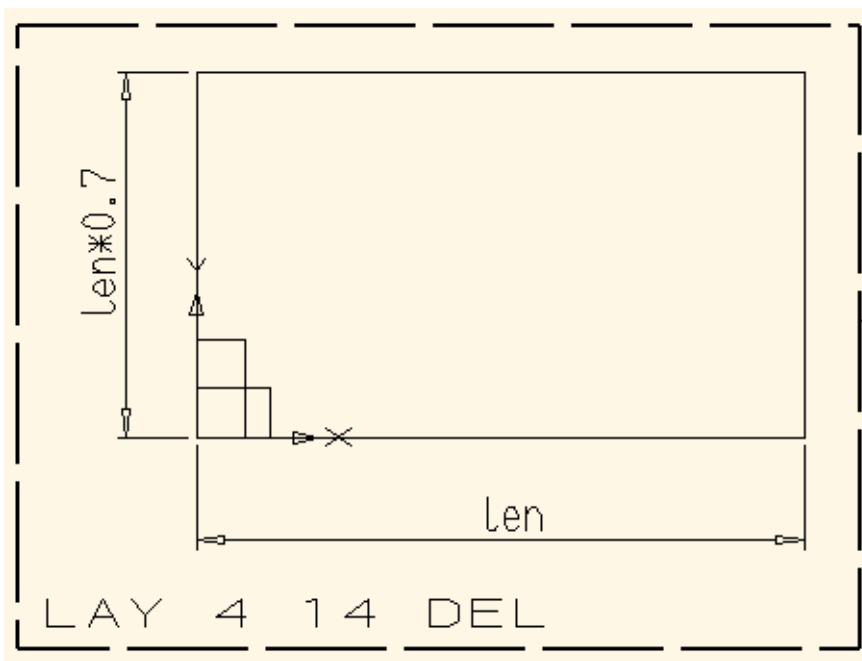
Draw and dimension a rectangle, as in Figure 179. The actual width and height are not important. Before you replace the dimension texts with variables, add the prim and then test to see if the drawing can be parameterized using the tool . Then change the dimension values to variables, as shown.

Figure 179 Drawing to be Parameterized



Adding in-sheet Command

To ensure that dimensioning and in-sheet command text do not appear on each parameterized drawing, add the command `LAY 4 14 DEL`.

Setting Values


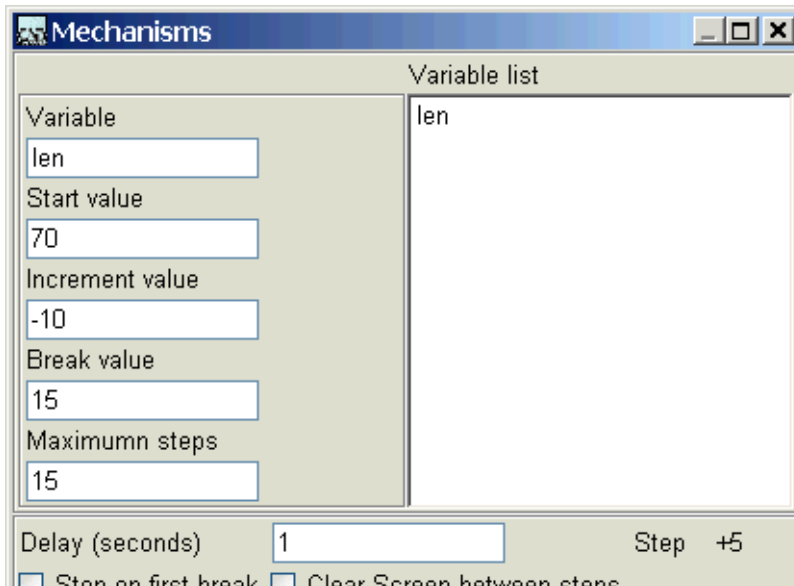
Now open the Mechanisms dialog with the tool  and define the parameters as given in the following figure.

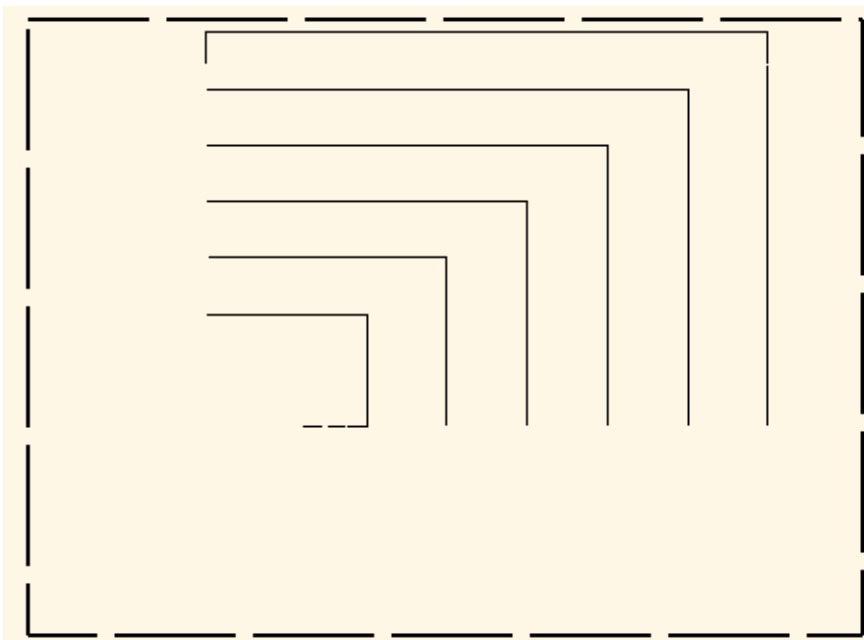
Figure 180 Settings for repeated parameterization



Running Simulation

You can see the result in Figure 181. The rectangle is repeatedly parameterized by changing the value of the variable `len`. The variable `len` was increased by 5 mm for each parameterization.

Figure 181 Result of Repeated Parameterization

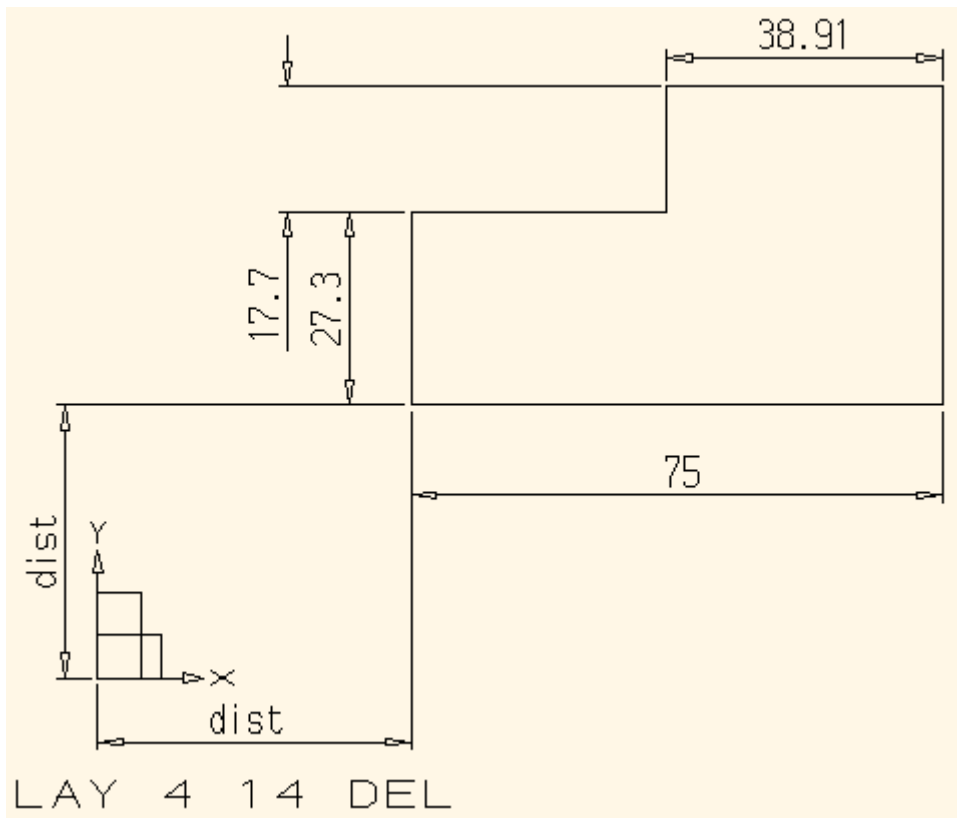


Example 2: Simulating Linear Motion

This is a simple example showing how to simulate linear motion. You will parameterize an object so that it moves across the screen, 5 mm further at each step.

Begin by creating the drawing shown in Figure 182. Use any convenient dimensions. Note the position of the prim. When you have drawn and dimensioned the object, add the variable parameter `dist`, as shown in Figure 182. This enables you to move the object away from the prim.

Figure 182 Object Definition Drawing



Adding in-sheet Commands

To ensure that dimensioning (layer 4) and in-sheet command text (layer 14) do not appear on each parameterized drawing, add the in-sheet command `LAY 4 14 DEL`.

Setting Values


Now open the Mechanisms dialog with the tool *Simulates mechanism motion*  and define the parameters as given in the following figure.

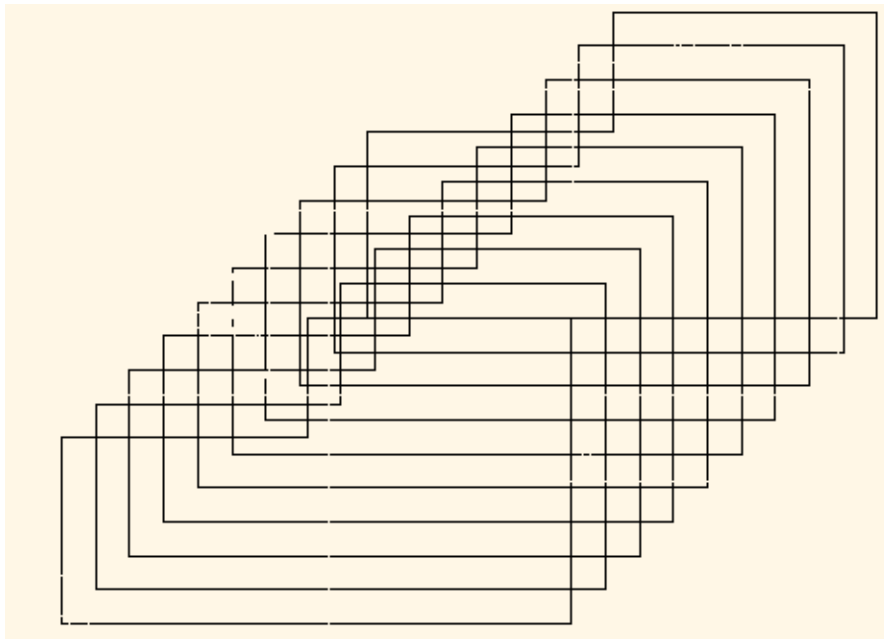
Figure 183 Settings for linear motion

Variable	<input type="text" value="dist"/>
Start value	<input type="text" value="50"/>
Increment value	<input type="text" value="-5"/>
Break value	<input type="text" value="5"/>
Maximumn steps	<input type="text" value="10"/>

Running Simulation

Figure 184 shows the result after running the simulation forward. For displaying only the last parameterization switch on the option *Clear screen between steps*.

Figure 184 Result of Simulating a Motion



Example 3: Simulating Rotary Motion

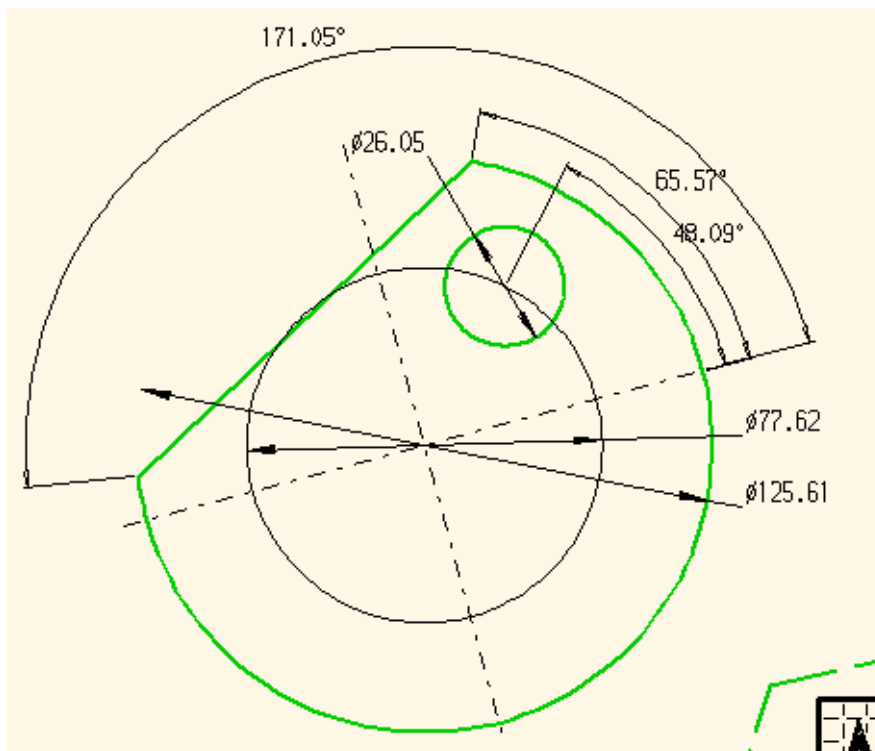
This example shows how you can use Parametric Design to simulate the rotary movement of a simple component which turns around its center. Later, two reciprocating rods will be added to the component to form a working mechanism.

Drawing and Dimensioning the Component

Draw the component shown in Figure 185 at a horizontal orientation and then rotate it so that it is not constrained to remain horizontal during parameterization.

Consider that the position where static baselines cross is the center point for both big circles. The center of the small circle is on the line of the inner big circle. Each circle is dimensioned with diameter dimensioning. The angle dimension is between the center of the small circle and the 15 degree baseline

Figure 185 **Component Definition**



Dimension the component once you have rotated it, as in Figure 185. Test the drawing with the tool . Do not proceed unless it can be parameterized.

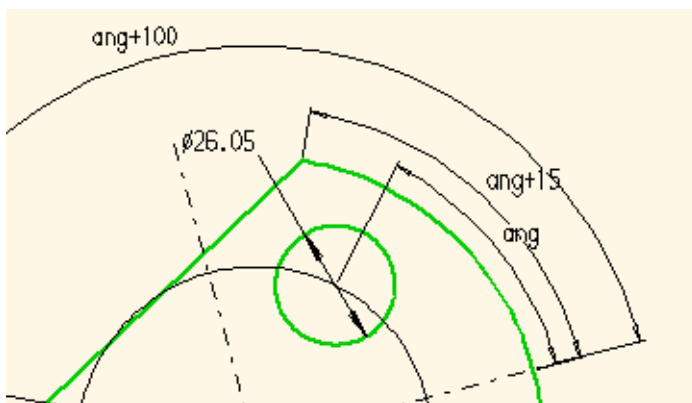
Adding in-sheet Commands

To ensure that dimensioning (layer 4) and in-sheet command text (layer 14) do not appear on each parameterized drawing, add the in-sheet command `LAY 4 14 DEL`.


Changing Dimension Values to Variables

Change the angular dimension values to variables, as in Figure 186. Save the sheet before continuing. You need this sheet for the next example.

Figure 186 Replaced angular dimension values by variables



Setting Values for the Variable

Now open the Mechanisms dialog with the tool Simulates mechanism motion  and define the parameters:

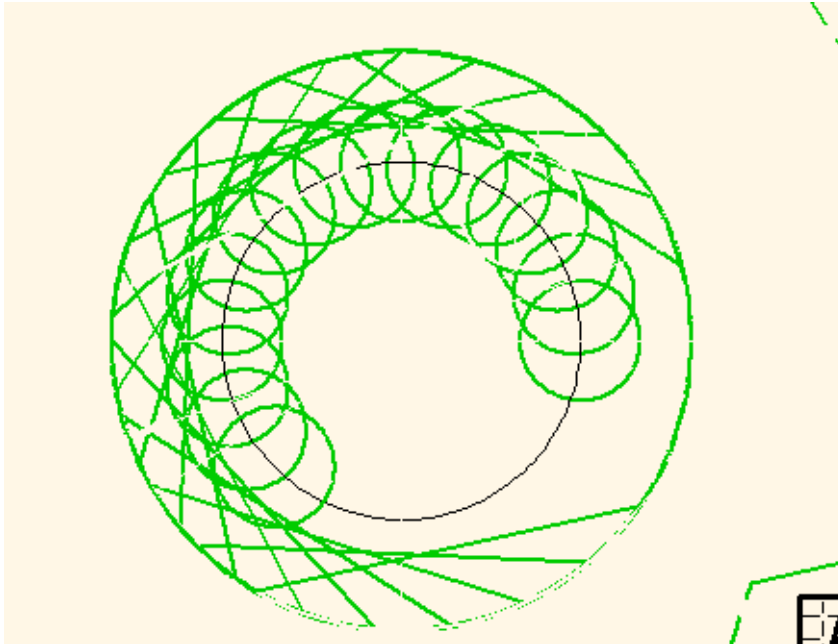
- Start Value = 0
- Increment Value = 15
- Maximum Steps = 15
- Delay = 0.5 seconds

Apply the settings and then use the buttons for simulation.

Result of Simulation

Figure 187 shows the result after running the whole simulation forward. For displaying only the last parameterization switch on the option Clear screen between steps.

Figure 187 Result of Simulating a Rotation



Example 4: Using a Program to Simulate Motion

Besides running simulation in the graphical user interface using the dialog *Mechanisms*, you also can use a *Bacis1* program. A program is a file containing a number of MEDUSA commands and it can be used to repeat a sequence of commands.

Creating the Component

This example uses the component you created in the last example, [“Example 3: Simulating Rotary Motion”](#) on page 197.

Writing the Program

You can create a program in two ways:

- Using your operating system text editor
- Using MEDUSA program mode

Details of how to create programs are given in the manual *MEDUSA Bacis1 Guide*.

Using a text editor

The example below shows a simple program which parameterizes the component you have drawn. To begin with, ANG is set to 40 degrees. The PARS CAN command is placed inside a loop which parameterizes the component repeatedly.

```
10 LET ANG = 40
20 LOOP
30     PARSCAN
40     BREAK IF (ANG.GT.100)
50     LET ANG = (ANG + 30)
60 ENDLLOOP
70 ENDRUN
```

With each loop, the program does the following:

- Parameterizes the component temporarily with the PARS CAN command
- Uses a logical operator to test if the value of ANG is less than 100 degrees: if this evaluates to true, then the program increments the value of ANG by 30 degrees
- If ANG is greater than 100 degrees, then the program stops

Please note: It is not possible to use PARS instead of PARS CAN in command loop. You will

generate error if you do use PARS because the drawing resulting from the first PARS command will be used as input for the next. This causes problems as there is no guarantee that a drawing produced by parameterization is itself capable of being parameterized. A drawing that is the result of parameterization may contain geometric constraints, such as coincident points or horizontal and vertical lines, which prevent it from being parameterized

Program mode

The following example shows how to create the simple program shown above using MEDUSA program mode. In this example, the file containing the parameterization program is called *MECH1.PRG*. User input is shown in bold text.

```

*PROGRAM
Program>AUTO 10 10
10 LET ANG = 40
20 LOOP
30 PARSCAN
40 BREAK IF (ANG.GT.100)
50 LET ANG = (ANG + 30)
60 ENDLOOP
70 ENDRUN
80 PROGRAM
Program>SAVE MECH1.PRG
PROGRAM saved on file: MECH1.PRG
Program>COMMAND

```

Running a Program

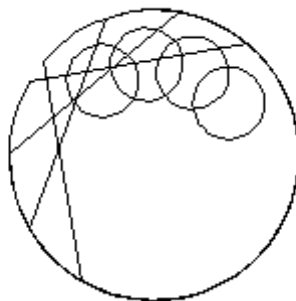
When you run the program, first clear the screen with the CLE command and then type RUN followed by the name of the file containing the program:

```

*CLE
*RUN MECH1.PRG

```

Figure 188 Result of Repeated Parameterization



Example 5: Simulating a Working Mechanism

In this example you combine rotary motion with two reciprocating rods to produce a working mechanism.

Take the drawing you created in “[Example 3: Simulating Rotary Motion](#)” on page 197.


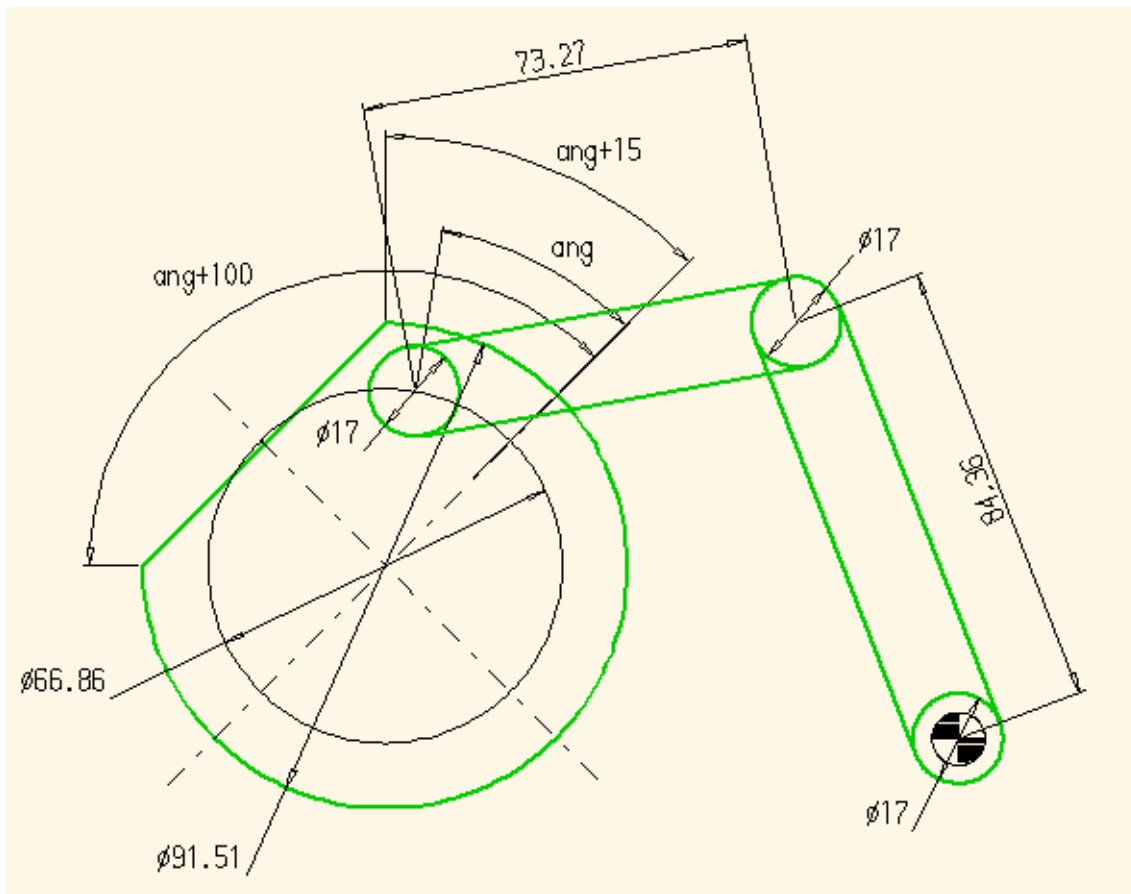
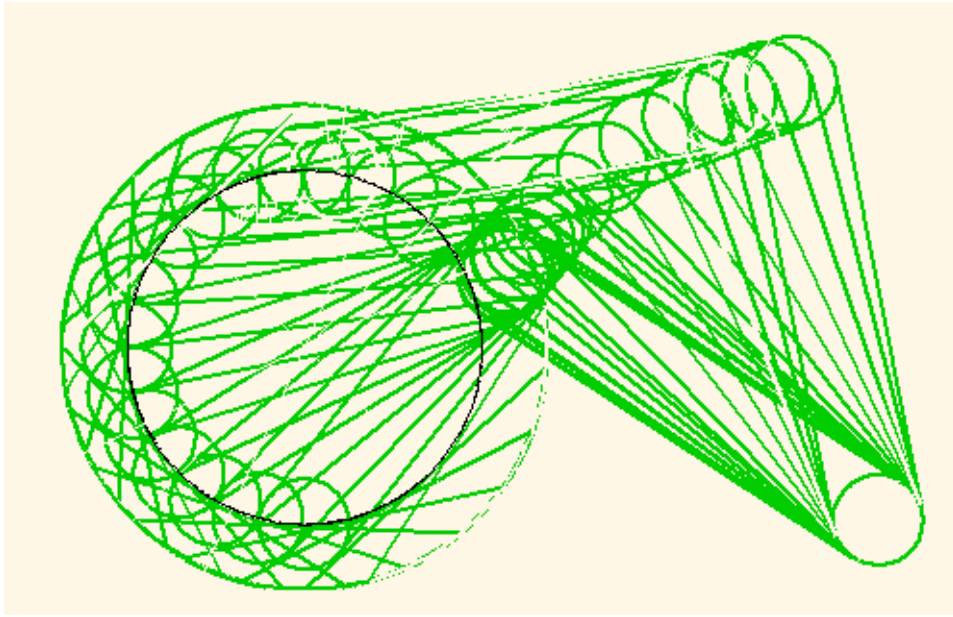
1. Draw two circles near to it, as illustrated in Figure 189. The new circles act as pivots for a rod.
2. Draw tangential lines between the new circles to create two rods. Use the tool *Creates construction lines through tangential chord of circles* from the tooltray *Creation tools* for designing them.
3. Dimension the new geometry. After each dimension use the tool *Draws the grid corresponding to the original drawing and undoes it*  for checking whether the grid is correct.
4. To complete the definition drawing, place a PVG prim at the center of the lower circle. The prim creates grid lines in the same way as the baselines at the other end of the mechanism. More important, the prim locks the circle into its current position. When the component rotates, the two rods will move, but the circle marked with the prim will remain fixed.

Figure 189 Adding Two Circles and Two Rods



Use the same settings for the variable `ang` as defined in “[Example 3: Simulating Rotary Motion](#)” on page 197. Apply the settings and then use the buttons for simulation. Figure 190 shows the result after running the whole simulation forward. For displaying only the last parameterization switch on the option `Clear screen between steps`.

Figure 190 Simulated Mechanical Movement



Plotting

It is not possible to plot the screen display to show all the movement of the mechanism. You can only plot the master drawing that is used for each parameterization.

The reason for this is that you parameterize the drawing temporary, not permanent. The master drawing is redrawn with new dimensions, but parameterization is immediately cancelled. When you plot a sheet, you get the information in the master drawing, not simply what you see on the screen.

Getting a Screen Dump

One way to get around this problem is to get a screen dump.

- On UNIX machines you can only obtain a screen dump if there is a hardcopy device attached to your workstation.
- Under Windows you can use the print screen button on your keyboard. Then paste the contents of the clipboard into an image editing program and print it.

Creating a Composite Drawing

If you cannot get a screen dump, you can create a composite drawing of the mechanism using the following procedure.

1. Increment the variable in the master drawing.
2. Parameterize the drawing permanently.
3. Save the sheet as a symbol.
4. Increment the variable on the master drawing by a further step.
5. Parameterize the drawing again.
6. Save the sheet as a symbol, using a new filename.

Continue this procedure until you have a symbol representing each stage in the movement of the mechanism. Then load all the symbols onto the same sheet to form a composite drawing which can be plotted.

OTHER APPLICATIONS

The Parametric Design system can be integrated with other MEDUSA software. This chapter gives some examples where Parametric Design has been applied to the MEDUSA 3D Design system and the MEDUSA Sheet Metal Design system.

The purpose of this chapter is to show what is possible. It is not intended to be a detailed reference.

- [MEDUSA Applications](#) 206
- [Parametric Design and 3D Modelling](#) 207
- [Parametric Design and Sheet Metal Design](#) 212

MEDUSA Applications

The name MEDUSA covers a variety of applications software including the following:

- MEDUSA 2D Design system
- MEDUSA Parametric Design
- MEDUSA 3D Design system
- MEDUSA Sheet Metal Design System
- NC definitions

2D Design System

The basic system is the 2D drafting system. This must be installed before any of the other packages can be used. The other applications can be added separately, as and when required.

3D Design System

The 3D design system takes a specially prepared 2D drawing and then turns it into a 3D model.

Sheet Metal Design System

The Sheet Metal Design system takes a specially prepared 2D drawing (or a 3D model) of an object and then calculates the true shape of the unfolded metal required to make it.

NC Definitions

Information from a specially prepared 2D sheet is used to define machining operations such as turning and milling.

Parametric Design and MEDUSA Applications

The common factor running through all these applications is that they take their information from a 2D drawing. The heart of MEDUSA is the 2D product. Any 2D drawing can be parameterized, so it is a simple step to parameterize a drawing and then use the drawing as input to one of the other MEDUSA applications.

Parametric Design and 3D Modelling

To parameterize a 3D model, you begin by drawing a 2D representation of a component on a 3D definition sheet. This section gives the procedure for an example consisting of a profile which has a hole of certain shape.

First draw the hole:


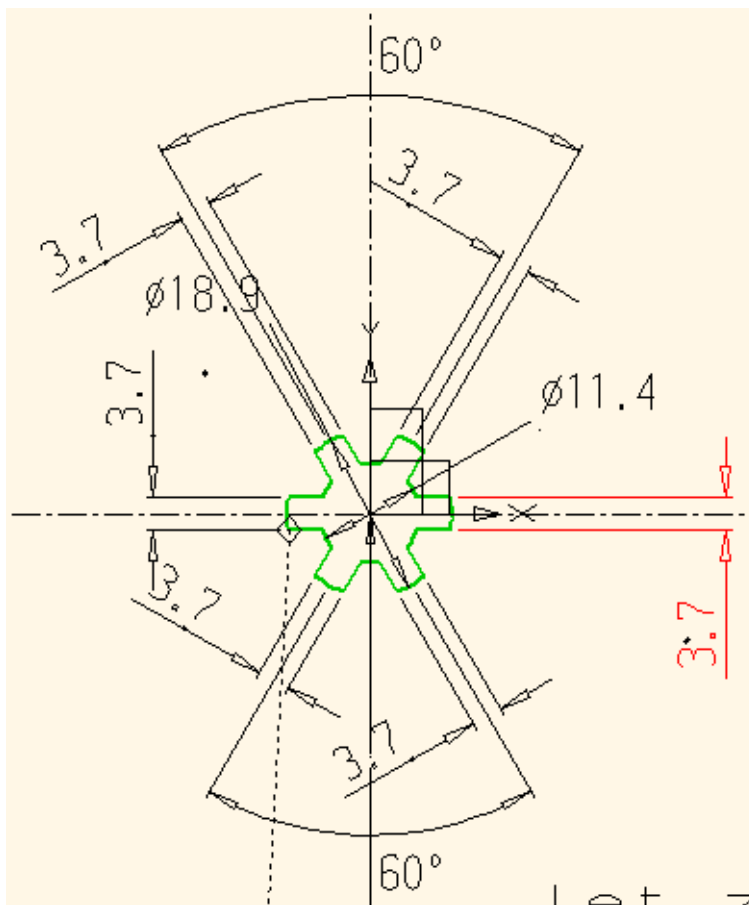

1. Open a 3D sheet and go to the XY view.
2. Draw the geometry with a profile line as given in the figure below (green circle with six pivots). Ensure that the center is identical with the datum point of the 3D DXY prim.
3. Dimension the diameter of inner and outer circle and the width of each pivot.
4. Dimension the angles between the upper and lower pivots (other angles need no dimension because of the 3D center line). Use the tool *Creates construction lines at 30, 60, 120 and 150 degrees*  to place the dimension lines properly on the center of the pivots area.

Figure 191 Example 3D Modelling, Hole

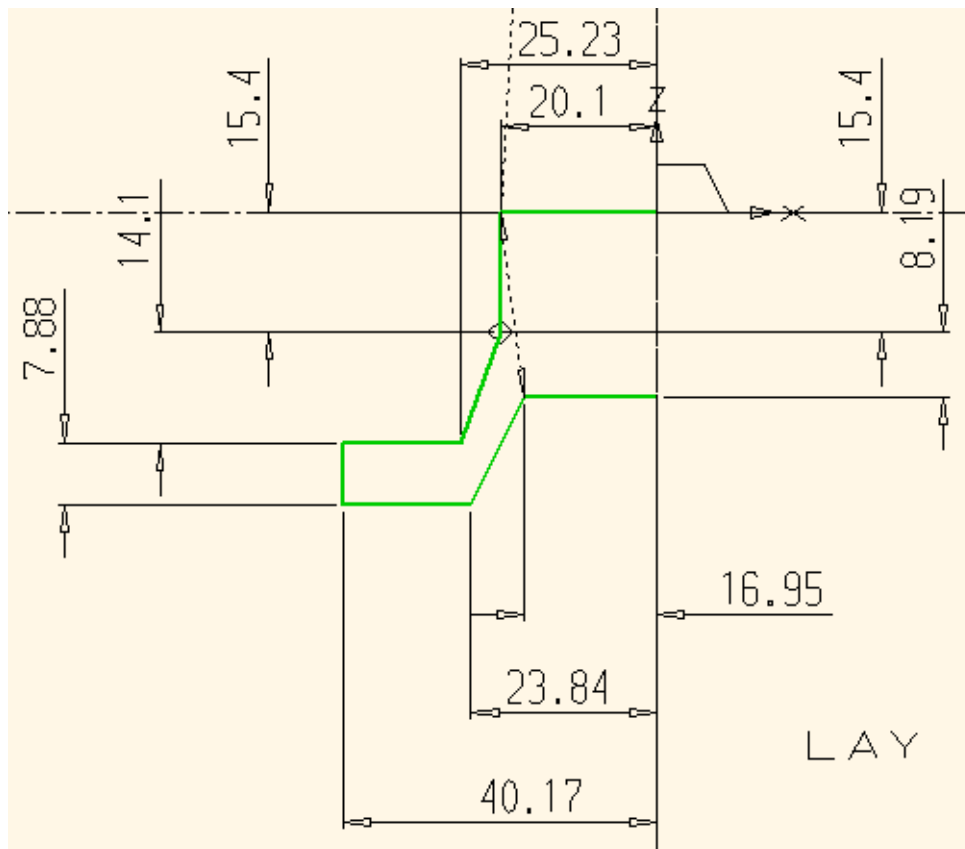




5. Choose the tool *Draws the grid corresponding to the original drawing and undoes it*  for checking the grid lines.

Now draw the plate:



6. Go to the XZ view.
7. Draw the profile as given by the green outline in the figure below. A center line need not to be drawn because it is already on the 3D sheet
8. Dimension the profile.

Figure 192 Example 3D Modelling, Plate




9. Choose the tool  Draws the grid corresponding to the original drawing and undoes it  for checking the grid lines.


Prepare the drawing for parameterization:


10. Define a variable for the width of the pivots of the hole, e.g. zw .
11. Define a variable for the outer diameter, e.g. $hd1$ and use it for the inner, e.g. $hd1-7$.
12. Define a variable and replace some dimension values of the plate with expressions using the specified variable.
See [Figure 193, "Example 3D Modelling, Prepared for Parameterization and Modelling"](#) on page 210 for examples.
13. Choose the tool  Creates free parametric command text  to assign values to the variables, e.g. `let zw=10, let hd1=40 and let hd2=20.`

14. Use the same tool to place the parametric command Text LAY 7 28 UNTRN in one of the view boxes in order to prevent 3D center and link lines used for parameterization.


Prepare for 3D modeling:

15. Draw a link line of style Slab Generator (tool Link line for linear sweeping ) from the hole to the plate starting at any point of the hole. The next two points define the height of the hole and also they have to be on points of the plate geometry.


16. Draw a link line of style Volume Revolution (tool Link line for rotational sweeping ) from the plate to the hole starting at any point of the plate which is not part of a horizontal segment. The next point is in horizontal direction (to ensure this, use the protractor) on the center line and it finishes at the datum point of the XY prim in the view with the hole.

Please note: If a point of the Link line for rotational sweeping  does not coincide with a point of the geometry, parameterization fails because it detects such point of the link line as an undimensioned point.

17. Name the link lines:

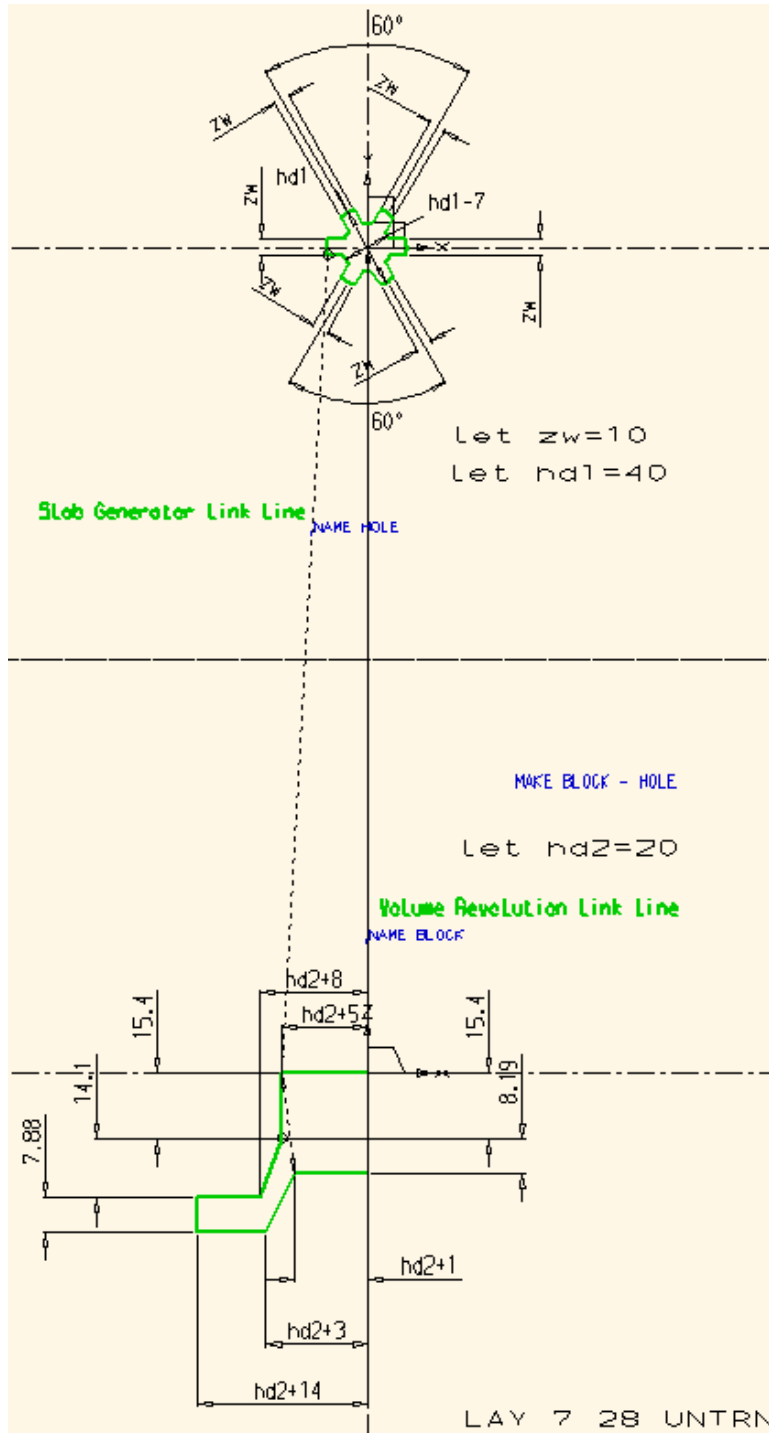
- a. Choose the tool Creates and manipulates 3D texts  from the 3D tooltray.
The dialog 3D Texts Controls - Line mode opens.
- b. Inside the drawing area *click left* on the link line you want to name.
The dialog 3D Texts Controls - Line mode lists all texts you can define now.
- c. Choose the text MODEL: NAME A and add it to the right list Actual Commands.
- d. Select the added text in the right list and name the link line by typing inside the field Values, e.g. Hole.
- e. Choose the button Edit for using the new name (the right list is updated).
- f. Choose the button Create to attach the text to the cursor.
- g. *Click left* close to the link line to place the name on the link line.
- h. Repeat the steps for the other link line.

18. Now you have to define that one geometry will be the hole in the other geometry:

- a. Again choose the tool Creates and manipulates 3D texts .
- b. *Click left* on the sheet but not inside any view box.
In the dialog 3D Texts Controls - Sheet mode you see all Possible Commands on the left side.
- c. Scroll the list until you find the entry MODEL: MAKE A - B and add it to the right list.
- d. Select the added text in the right list and replace A by the name of the plate and B by the name of the hole. Now you should see something like that:
MODEL: MAKE Plate - Hole
Ensure that there is a space between names and minus sign. If you run modelling later one, this command works like subtracting the Hole from the rotated profile Plate.
- e. Choose the button Edit for applying the command and then Create for attaching it to the cursor.

- f. Click left on the sheet but not inside any view box to place the command.
You should have now a drawing like in the figure below (consider that the MAKE command was moved into the XZ view only to show it, it has to be outside of any view!).

Figure 193 Example 3D Modelling, Prepared for Parameterization and Modelling



Running parameterization and 3D modelling:





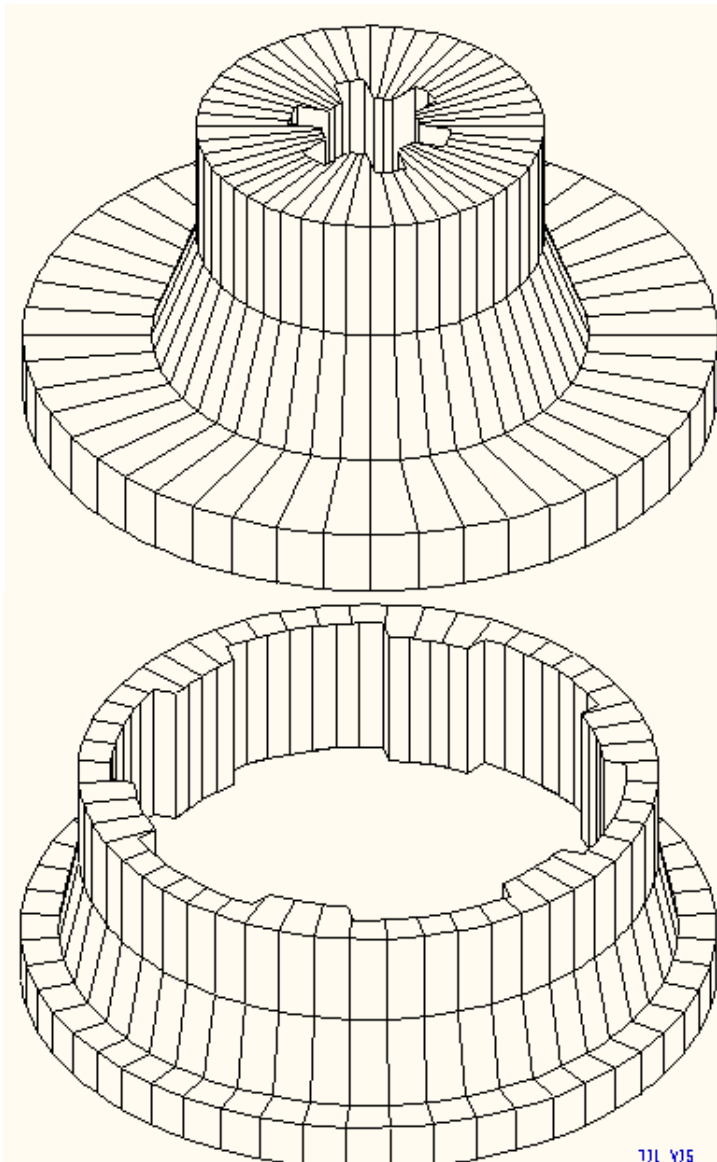
19. Assuming that the grid is already checked, choose the tool  to parameterize the drawing temporary.
20. If everything is fine, run the tool  to parameterize permanently.
21. Save the sheet under a new name in order to keep the input sheet for further parameterizations.
22. Run 3D Modelling with the tool . According to the variables you used for parameterization the model is drawn. Additionally a model file is written using the sheet name with extension `mod`.
23. Choose the tool  to display the model.

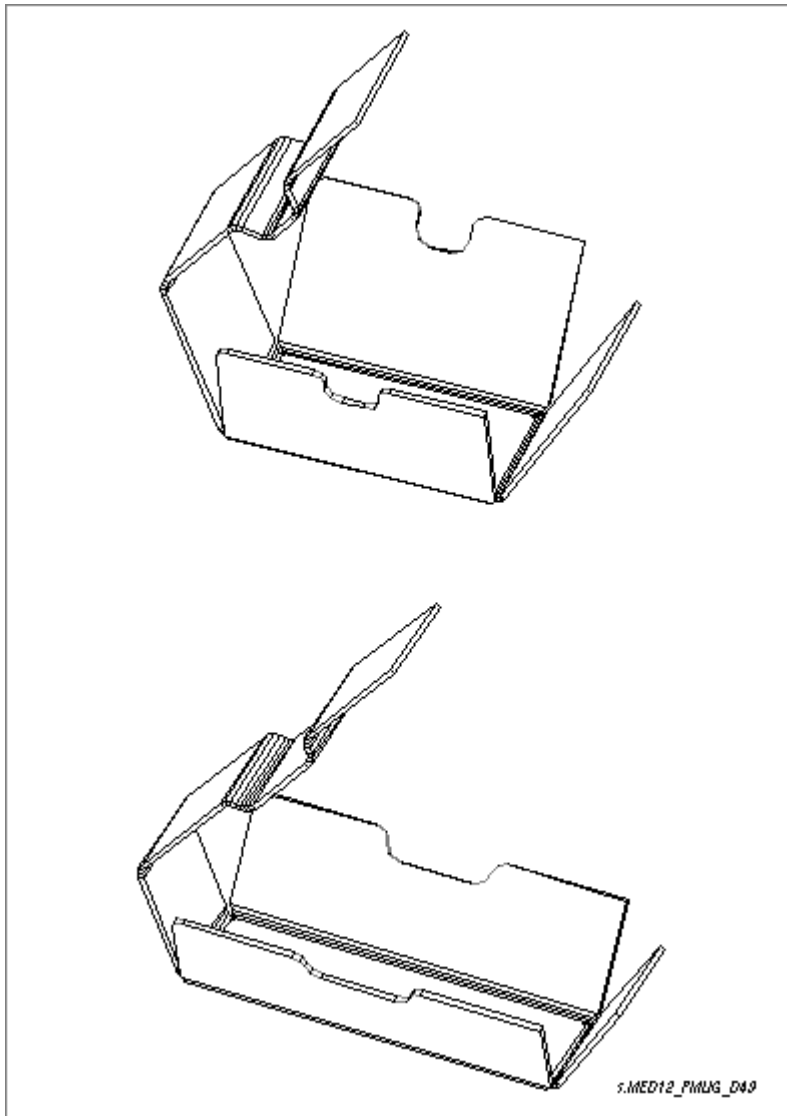
Figure 194 Example 3D Modelling, Possible Results



Parametric Design and Sheet Metal Design

You can parameterize an initial definition sheet and produce a whole range of differently folded components from a single drawing. Figure 195 shows two parameterized versions of a box.

Figure 195 Parametric Design and MEDUSA Sheet Metal Design



APPENDIX A PARAMETRIC DESIGN ELEMENT DEFAULTS

This appendix gives a brief summary of the element type and layer defaults for elements used in Parametric Design.

- Basic Parametric Elements 214
- Tables, Symbols, Groups and Error Messages 215
- Graphical User Interface 216

Basic Parametric Elements

The following table shows the type and layer defaults for some basic Parametric Design elements. You will use some of these elements whenever you use Parametric Design.

Element	Element type	Layer
Parametric viewbox	LPV line	28
Static baselines	LBL line	4
Dynamic baselines	LBL line	16
Reference prims	PVG, DXY, DYZ, DZX, DYZ, DZY, DXZ prims	4
Grid lines	STK line	99
Lines created by the PAR GRIS options COL and CIR	L3 line	17
In-sheet commands	TCO text	14

Tables, Symbols, Groups and Error Messages

Parametric Table Elements

These elements are used in tables. Use a table when you need to store a number of values.

Element	Element type	Layer
Table boundary line	LTB line *	14
Table name text	TTB text *	14
Row and column names	TRC text	14
Table entries	Any other text	14

* These elements must be part of a group.

Parametric Symbol Elements

These elements are used for creating parametric symbols and instance groups.

Element	Element type	Layer
Symbol attachment point	ATP text	13
Symbol name	SPS text *	13
Instance group attachment point	SAT text *	13
Instance group command	SCO text *	13

* These elements must be part of a CPI group.

Parametric Group Elements

These elements are used in parametric groups.

Element	Element type	Layer
Parametric group line	LPG line*	15
Parametric group prim	PPG prim*	15

* These elements must be part of a group.

In-sheet Error Message Elements

These elements are used in error messages that are written onto the sheet.

Element	Element type	Layer
Error texts	TS1 or TR1 text	99
Error lines	L6 line	99

Graphical User Interface

MEDUSA provides a dialog, which gives you most of the element types used for Parametrics and which allows you to change them temporary for the current session.


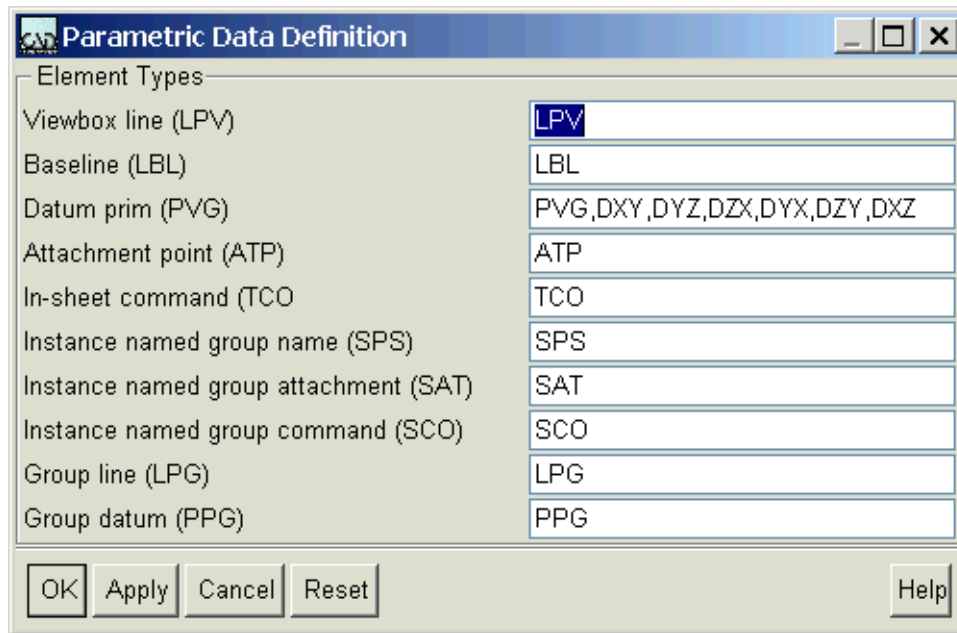
For opening the dialog Parametric Data Definition choose the tool Display the current settings of the parametric line styles  from the tooltray Parametrics.

Figure 196 Dialog Parametric Element Types



Consider that for parameterization 3D prims can be used (entry Datum prim (PVG), values DXY, DYZ etc.).


For details on changing parametric element types see the *Parametrics Reference Guide*, chapter „Changing Element Types“.

APPENDIX B ERROR AND WARNING MESSAGES


This appendix gives an alphabetical list of error and warning messages that occur in the Parametric Design system.

Any relevant MEDUSA error message can also appear either in the sheet or on the screen. For information on MEDUSA error messages, refer to the *MEDUSA Basis1 Design Commands Guide*.

Ambiguous Construction

The new grid on which the construction is based does not result in a unique set of grid lines. When a construction has unnecessary supporting grid lines, Parametric Design uses only the minimum necessary to support the construction. The remaining grid lines are then checked for consistency. This error is normally caused either by over-dimensioning, or by using a tolerance that is too large. Use the tool  to identify the grid lines that are used to support the construction.

Ambiguous Point

This point is at a point on the old grid that does not transform to a unique point on the new grid. This is because the grid lines which intersect at the point in the old grid do not all intersect at the same point in the new grid. This error is normally caused by over-dimensioning or by using a tolerance that is too large. Use the tool  to identify the grid lines that intersect at the point.

Cannot Resolve Table

The table cannot be separated into rows and columns with the same number of entries in each row and in each column. Check the types, positions and number of texts in the table.

Element Types Insufficiently Specified

This error occurs if one of the construction elements used as input to Parametric Design does not have a valid type specified for it. This is most likely to be a problem in the DDL, as the Parametric Design system has a complete set of defaults.


Fatal Error During Dimensioning

This error is produced when parameterization is executed and an error is found in your dimensioning, for example a corrupt dimension group.

Group Datums Define Impossible Transformation

This error is normally caused by coincident or collinear datum points in a parametric group.

Illegal Context for Element

This error is produced when you use tool Load parametric symbol  and an element is found in the symbol definition which is not valid as input to Parametric Design.

Illegal Dimension Type

Axonometric and tolerance limit dimensions are not supported by Parametric Design. See [“Dimensioning” on page 25](#), to see which dimensioning types you can use as input to Parametric Design.

Illegal Expression

An expression has been used to define a variable which includes an invalid operator or function. See [“Variables and Expressions” on page 91](#), for a list of valid operators and functions.

Illegal Point/line Function

The point function used on the baseline indicated is not recognized by Parametric Design. See [“Geometric Constraints” on page 75](#), for more information on point functions.

Illegal Variable Name

The text indicated should be a valid variable name. This message may occur with symbol attachment points and table row and column names.

Illegal Viewbox Definition

The viewbox indicated interferes with other viewboxes in the sheet. Viewboxes may not overlap or be nested. The X and Y limits of each viewbox line are used when testing for overlap. The other viewboxes are processed as normal.

No Constructions in Viewbox

No grid was generated because there was no valid construction in the viewbox. This is either because there are no dimensions or other constructions in the viewbox, or the constructions are on unhittable layers.



No Current Sheet

There must be a current sheet before you can execute any commands

No Solution Possible

The point or construction has no solution in the new grid. For example, a point may lie at the intersection of two circular grid lines in the old grid, but in the new grid the circles are too far apart to intersect.

No Supporting Grid(s)

The construction indicated does not have sufficient grid lines to support it. Use the tool  Draws the grid corresponding to the original drawing and undoes it  to examine the existing grid. This should reveal the missing link that is required to build up the grid. If the PAR BAS switch is OFF, you may need to add a dynamic baseline.

No Table Found

No table with the specified name can be found. If you give TBL command as an in-sheet command in a parametric symbol definition, the search is restricted to the symbol containing the command.

No Viewboxes in Sheet

The Parametric Design system only processes geometry which is inside a parametric viewbox.

Obsolete command GLMT, use PAR LIM

Obsolete command GLME, use PAR LIM

Obsolete command GLRAD, use PAR LIM

Obsolete command QGLRAD, use PAR LIM

These error messages relate to obsolete Prime Variational Geometry commands for displaying parts of the grid. Use the PAR LIM switch instead. When the PAR LIM switch is ON, the grid lines extend to the edges of the viewbox.

Point Not Dimensioned

The point indicated does not lie on a grid intersection. All line points must lie on grid intersections. This message may be the result of under-dimensioning, but can also be caused by inaccurate drafting, where points that should be coincident are not quite coincident. The error can also occur if the end points of dimensions are not at grid intersections. Be careful to probe the correct points when you are dimensioning objects.

Problem With Construction

Some unidentified problem has occurred with a construction. This is often due to zero length baselines or expressions that evaluate to the wrong data type.

Problem with Parametric Group Datums

An internal problem has occurred with the parametric group datum points.

Problem with Parametric Group Line

A problem has occurred in creating a test polygon from the parametric group line.

Problem with VAR to LIM Conversion

This message is displayed following the unsuccessful completion of displaying the grid or parameterization. A problem has occurred in attempting to convert a VAR dimension to a LIM dimension. Refer to the *MEDUSA Parametric Design Reference Guide* for more information about this message.

Row/col not Found

The specified text string cannot be found as a row or column name in the specified table.

Row/col text Not Outside Body Texts

The row and column texts must be outside the area of the table containing all the entries.

Specified type is not a line

Specified type is not a prim

Specified type is not a text

These errors are due either to a problem in the DDL or an invalid type specified in the PAR DDL command. Refer to the *MEDUSA Parametric Design Reference Guide* for more information about the PAR DDL command.

System error - Automatic Baseline Buffer Overflow

Grid lines that cross the potential grid line are stored in an buffer. When this buffer overflows you will see the message above. This means that a baseline will not be automatically generated. This buffer is also used with output radial dimensions. To avoid the problem you can add a baseline explicitly.

System error - Grid Finding Buffer Overflow

Each grid lines that passes through a point is stored in a buffer. When too many grid lines pass through a point this buffer overflows and you will see the message above.

System Error - Element Type Storage is Full

This error may occur when you use the PAR DDL command to define new element types. The maximum number of element types you can define with the PAR DDL command is 50. Sixteen are defined by default. Refer to the *MEDUSA Parametric Design Reference Guide* for information about the PAR DDL command.

System Error - Parametric Workspace is Full

This error is normally caused by having too many dimensions in one viewbox. The workspace is used for storing information on the constructions and grid lines.

System Error - too Many in-sheet Commands

The maximum number of in-sheet commands in a viewbox is 200. Less may be allowed if you use large numbers of variables.

Too Few Rows/columns/entries

There must be at least one row, one column and one entry in a table.

Too Many Datums in Parametric Group

The maximum number of datum points in a parametric group is three.

Too Many Entries in Table

The maximum number of entries in a table is 2500.

Too Many Parametric Groups

The maximum number of parametric groups in a viewbox is 100.

Too Many Rows and Columns in Table

The maximum total number of rows and columns in a table is 100.

Too Many Viewboxes

The maximum number of viewboxes in a sheet is 20.

Trying to Move Point Outside max. area

When you parameterize an object, no points can be moved outside the maximum drawing area.

Typing Mistake!

This error is the result of bad command syntax either in an in-sheet command or in a command that you have entered directly. This error also appears, if you used for an attachment point text the wrong text type, e.g. TCO instead of ATP.

Unable to Convert Dimension

The PAR DIM option VTL has failed. Either the VAR dimension being converted or the resulting LIM dimension is invalid. Refer to the *MEDUSA Parametric Design Reference Guide* for more information about the PAR DIM command.

Unable to Load Sheet File

This error occurs when you try to load a parametric symbol definition which does not exist or is not readable. Make sure you have given the correct pathname for the symbol file.

Unknown Element Type

This error is generated when you try to specify an illegal element type in the TBL DDL command. Refer to the *MEDUSA Parametric Design Reference Guide* for information about the TBL DDL command.

Unset Variable

This error occurs when you try to display the grid for or parameterize geometry which includes a variable that has no value.

Warning - Ambiguous Dimension Texts Interpreted as Values

One or more dimension texts may have been interpreted as simple values with a prefix text, but the texts could also have been interpreted as variable names. Alternatively, the dimension text may have been interpreted as both a value produced by dimensioning and also as an expression.

Warning - Messages Written into Sheet

This message is output to the screen whenever error messages are written into the sheet.

Warning - no Grid Lines Drawn, as PAR LIM switch is OFF

Certain grid options are only available with limited grid lines, that is, when the PAR LIM switch is ON.

Warning - PARDIM command interrupted

Warning - PARGRIS command interrupted

Warning - PARLOA command interrupted

Warning - PARS command interrupted

These messages will appear when you interrupt any of these commands.

Warning - point(s) Scaled Outside Viewbox

One or more points were moved outside the viewbox when the elements were moved.

Warning - Protected Elements Not Transformed or Deleted

Elements on protected layers cannot be transformed or deleted by the Parametric Design system. However, they can be used during the creation of the grid. This warning occurs if there are elements on a hittable but protected layer that may also be transformed or deleted.

LIST OF FIGURES

Figure 1	Changing the Dimensions of an Object 10	Figure 37	Toolset for Drawing Grids 46
Figure 2	A Family of Parts Made From a Single Master Drawing 11	Figure 38	Dialog Grid construction 48
Figure 3	A Symbol Library Created Using Parametric Design 12	Figure 39	Toolset for Parameterizing 50
Figure 4	Simulation of the Movement of a Lift Arm Assembly 13	Figure 40	Error Messages 51
Figure 5	Examples of Parametric Viewbox Shapes . . 14	Figure 41	Example1: Rectangle with DXY Prim Reference Point 53
Figure 6	Intersecting Static Baselines 15	Figure 42	Example1: Effect of Drawing Grid 54
Figure 7	The Parametric Design Datum Prim (Type PVG) 15	Figure 43	Rectangle Dimensioned for Parameterizing . 55
Figure 8	Orthogonal View Prims 16	Figure 44	After Parameterization 56
Figure 9	Positioning Baselines 16	Figure 45	Example2: Rectangle with Intersecting Baselines 57
Figure 10	Positioning a Prim 17	Figure 46	Effect of Changing the Grid Tolerance 59
Figure 11	Parametric Grid Lines 18	Figure 47	Fully Dimensioned Drawing 60
Figure 12	Possible Parameterization 19	Figure 48	Result of Parameterizing 61
Figure 13	Avoiding Coincident Points 19	Figure 49	Component to be Drawn 64
Figure 14	Coincident Points 20	Figure 50	Displaying the Grid 64
Figure 15	Collinear Points on the Same Grid Line . . 20	Figure 51	Addition of First Chain Dimension 65
Figure 16	Aim of Parameterization 20	Figure 52	Adding More Dimensions 65
Figure 17	Example of Parameterizing a Specific Case . 21	Figure 53	Fully Dimensioned Component 66
Figure 18	Example of an Ideal Master Drawing 21	Figure 54	Complete Parametric Grid 66
Figure 19	2D Parametrics Tooltray 23	Figure 55	Old Component With New Parameters 67
Figure 20	Grid Lines Generated By Reference Point . 27	Figure 56	Parameterized Component 67
Figure 21	Extra Grid Intersection Generated by Dimensioning Line AB 27	Figure 57	Object to be Dimensioned 68
Figure 22	Grid Completed by Second Dimension 28	Figure 58	Displaying the Grid 69
Figure 23	Toolset for Tolerance Settings 29	Figure 59	Detail Showing Fillet Grid Lines 69
Figure 24	Dialog for Placing Tolerance Setting Texts . 29	Figure 60	Object With Angular Notch 71
Figure 25	Dialog for Tolerance Settings 30	Figure 61	Result of drawing the grid 71
Figure 26	A Linear Dimension 32	Figure 62	Adding a Line 72
Figure 27	Object With Intersecting Baselines 33	Figure 63	Dimensioned Angle With Grid 72
Figure 28	Chain Dimensioning With Center Support . . 33	Figure 64	Dimensioning an Unsymmetrical Angle . . . 73
Figure 29	Dialog for Filletting Settings 34	Figure 65	Simplified Diagram of a Bicycle Chainset . . 76
Figure 30	Examples of Under-Dimensioned Drawings . 36	Figure 66	Dialog Parametric Point Function, activated 80
Figure 31	Examples of Over-dimensioned Drawings . 37	Figure 67	Object to be Parameterized 81
Figure 32	Dialog for Changing Texts 38	Figure 68	Inferred Baselines 81
Figure 33	Toolset for Creating Viewboxes 41	Figure 69	Object with New Set of Dimensions 82
Figure 34	Examples of Parametric Viewbox Shapes . 41	Figure 70	Object to be Parameterized 83
Figure 35	Toolset for Creating Reference Points 43	Figure 71	Displaying the grid 84
Figure 36	Parametric Point Function 43	Figure 72	Inferred Baselines 84
		Figure 73	Object to be Parameterized 86
		Figure 74	Detail of Fillet Construction from Component in Figure 73 86

Figure 75	Displaying the grid	87	Figure 126	Symbol Loaded Using Teble Values From ROW2	147
Figure 76	Fully Dimensioned Drawing With Grid	87	Figure 127	Toolset for CPI Named Groups	148
Figure 77	Fillet Detail from Figure 76	88	Figure 128	Example for a CPI named group	149
Figure 78	Dynamic Baselines	88	Figure 129	Dimensioned Component	150
Figure 79	After Parameterization	89	Figure 130	Changing Dimensions to Variables	151
Figure 80	Use of Variables in Joist Definition	92	Figure 131	Marking Positions of Tabs	151
Figure 81	Result of Changing Joist Dimension Variables 93		Figure 132	Dimensioned Tab	152
Figure 82	Dimensioned Drawing	95	Figure 133	Individual Points on the Fillets	153
Figure 83	Replacing Dimension Texts With Variables .	95	Figure 134	Tab Prepared for Saving	153
Figure 84	Example of In-sheet Commands	97	Figure 135	CPI named Groups in Position	154
Figure 85	Dialog for Creating LCIS Variables	98	Figure 136	Detail of CPI named Group	154
Figure 86	Dialog for Query LCIS Variables	99	Figure 137	Dimensioning the Attachment Points	156
Figure 87	Dialog Administrate parametric variables .	100	Figure 138	Result of temporary parameterization	157
Figure 88	Add new parametric variable	101	Figure 139	Alternative Parameterization	158
Figure 89	Dialog Load and Update parametric variables. 102		Figure 140	Structure of a Parametric Group	160
Figure 90	In-sheet DEF Command	105	Figure 141	Parametric Group With Three PPG Prims. .	161
Figure 91	Dialog Parametric Graphics Control	111	Figure 142	Toolset for Groups	162
Figure 92	Toolset for Switches.	112	Figure 143	Example of Preventing Parameterization with Group Line and Prim	162
Figure 93	Dialog Parametric Switches.	112	Figure 144	Component with Bolts	163
Figure 94	Dialog Command Options	113	Figure 145	Adding a Parametric Group Line	164
Figure 95	Collinear Points With PAR COL OFF.	114	Figure 146	Result of Parameterization	164
Figure 96	Collinear Points With PAR COL ON	114	Figure 147	Prim Placed Inside Parametric Group	165
Figure 97	Effect of PAR CIR OFF	115	Figure 148	Change of Dimensions after parameterization 166	
Figure 98	Effect of PAR CIR ON	115	Figure 149	Dimensioned Component	167
Figure 99	Original Geometry With New Parameters .	116	Figure 150	Change Made to Component	168
Figure 100	Original and Parameterized Versions	116	Figure 151	Creation of New Group and Addition of Group Line	168
Figure 101	The Parts of a Table	118	Figure 152	Addition of Parametric Group Line	169
Figure 102	The Structure of Table Elements	119	Figure 153	Parameterized Component	169
Figure 103	Toolset for Tables	120	Figure 154	Parameterized Component	170
Figure 104	Dialog for Defining Tables	121	Figure 155	Bolt Hole With Parametric Group	171
Figure 105	Example of a Table with 4 entries/rows and 3 variables.	121	Figure 156	Rotating a Parametric Group.	171
Figure 106	Table With Variables	122	Figure 157	Component With Bolt Holes	172
Figure 107	Example of an In-sheet TBL Command.	125	Figure 158	Detail of Bolt Hole from Figure 157	172
Figure 108	Original Component.	126	Figure 159	Parametric Group Line	173
Figure 109	Dimensioned Component With Grid	126	Figure 160	Parameterized Drawing.	173
Figure 110	Changing Dimension Values to Variables .	127	Figure 161	Adding More Prims	174
Figure 111	Table of Variable Values	127	Figure 162	Parameterized Drawing.	174
Figure 112	Table and in-sheet commands.	128	Figure 163	Dialog Post-Parameterization	177
Figure 113	Effect of the temporary parameterization. . .	129	Figure 164	Callbacks available for Post-Parameterization 178	
Figure 114	Dialog for Selecting a Table Variant	131	Figure 165	Message for missing primary datum prim .	178
Figure 115	Example for Variant Parameterization, Input . . 132		Figure 166	Example for an invalid parameter after entering 181	
Figure 116	Example for Variant Parameterization, Result. 132		Figure 167	Example for an invalid parameter after running callback	181
Figure 117	Attachment Points	135	Figure 168	ppcb_array_copy_angled	182
Figure 118	Dialog for Loading Parametric Symbol	138	Figure 169	ppcb_array_copy	183
Figure 119	Example for Error Messages while Loading a Parametric Symbol	140	Figure 170	ppcb_multi_copy	183
Figure 120	Popup while Loading Parametric Symbol .	141	Figure 171	ppcb_copy_rotate_pcd	184
Figure 121	Symbol Definition	142	Figure 172	ppcb_copy_and_mirror	184
Figure 122	Parametric Symbol Loaded Onto Sheet	143	Figure 173	ppcb_copy_rotate	185
Figure 123	Table as Part of a Parametric Symbol	144	Figure 174	Example: Parameter Settings in the Dialog Post-Parameterization.	186
Figure 124	Table	145			
Figure 125	Dialog Select values from Table	146			

Figure 175	Example: Before Parameterization	186	Figure 187	Result of Simulating a Rotation	199
Figure 176	Example: After Parameterization	187	Figure 188	Result of Repeated Parameterization	201
Figure 177	Example: After Running Post-Parameterization 187		Figure 189	Adding Two Circles and Two Rods	202
Figure 178	Dialog for Running Mechanisms	191	Figure 190	Simulated Mechanical Movement	203
Figure 179	Drawing to be Parameterized	193	Figure 191	Example 3D Modelling, Hole	207
Figure 180	Settings for repeated parameterization.	194	Figure 192	Example 3D Modelling, Plate	208
Figure 181	Result of Repeated Parameterization	194	Figure 193	Example 3D Modelling, Prepared for Parame- terization and Modelling	210
Figure 182	Object Definition Drawing	195	Figure 194	Example 3D Modelling, Possible Results	211
Figure 183	Settings for linear motion	196	Figure 195	Parametric Design and MEDUSA Sheet Metal Design	212
Figure 184	Result of Simulating a Motion	196	Figure 196	Dialog Parametric Element Types	216
Figure 185	Component Definition	197			
Figure 186	Replaced angular dimension values by vari- ables	198			

INDEX

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Numeric

2D Design System 206
2D Parametrics Tooltray 23
3D Design System 206
3D Modelling 207

A

access values from a table 124
Arc Dimensions 34
Arithmetic Operators 106
assigning Values to LCIS Variables 98
assigning Values to Variables Using in-sheet commands 96
Attachment Points 134
avoid coincident points 19

B

Baseline Point Functions 78
baselines 78
 draw dynamic - 80
 point functions 44
Basic Parametric Elements 214
building the Parametric Grid 46
buttons for running simulation 192

C

Callbacks available for Post-Parameterization 178
callbacks distributed with the 2D Parametrics module 182
Center Support 32
Change selected dimension text, tool 38
changing layer properties 110
Circle Dimensions 34
Coincident Points 19
collinear points 20
Command Options 113
command texts 112
Composite Drawing 204
constraints
 explicitly specified 80
Coordinate Dimensioning 33
CPI named group 148
CPI Named Groups
 toolset 148
create a dynamic baseline 80

Creates a parametric table 121
Creates free parametric command text 113
 usage 96
Creates parametric command text 113
Creates parametric symbol attachment text 135
Creates parametric tolerance text 29
Creates parametric viewbox lines 41
Creates parametric viewboxes 41
Creates static baselines, tool 80
Creates user LCIS variables 98
Creating
 a parametric group 162
 a table 120
 a Viewbox 41
 an Instance Group 148
 CPI Named Groups (example) 154
 Families of Parts 11
 Parametric Symbols 134
 post-parametric definition sheets 180
 Reference Points 43
 Symbol Libraries 11
 Variable Tolerances 31
 Viewboxes 41

D

Datum Dimensioning 33
DEF Command 105
default layers 108
DEL 109
Deletable 109
delete error messages 52
Deleting Elements During Parameterization 129
deleting grid lines 47
Deleting Inferred, Dynamic Baselines 77
demo sheet
 post-parametric - 188
Dialog
 Administrate parametric variables 100
 for Changing Texts 38
 Load and Update parametric variables 102
dialog
 Command Options 113
 Filleting Settings 34
 for Defining Tables 121
 for Loading Parametric Symbol 138

- for Running Mechanisms 191
 - for Selecting a Table Variant 131
 - Parametric Element Types 216
 - Parametric Graphics Control 111
 - Parametric Point Function 43
 - Parametric Switches 112
 - Post-Parametrization 177
 - Query LCIS Variables 99
 - Tolerance Settings 30
 - Dimension Text Change 38
 - Dimension Types 26
 - dimension values
 - specifying how the system calculates new - 30
 - dimensioning 25
 - an Angle 72
 - an Unsymmetrical Angle 73
 - Hints 37
 - Instance Groups (example) 155
 - over - 37
 - Techniques 27
 - under - 36
 - dimensions
 - After Parameterization 49
 - and the Grid 18
 - in Parametric Design 26
 - Display Post Param Dialog, tool 177
 - Display the current settings of the parametric line styles, tool 216
 - Displaying Inferred, Dynamic Baselines 77
 - Displays the current settings of the parametrics system 112
 - Displays the current tolerance settings 29
 - draw dynamic baselines 80
 - Drawing Grids, toolset 46
 - Dynamic Baselines
 - delete 77
 - display 77
 - draw 80
 - Dynamic Group
 - with One Prim 165
 - with Three Prims 172
 - with Two Prims 167
- E**
- elements in parametric group 160
 - elements of a table 119
 - Enter Your Own Parameters 49
 - error messages 51, 217
 - delete 52
 - while using variables 104
 - examining the Parametric Grid 48
 - Example
 - 2D parametrics and 3D 207
 - Chain Dimensioning With Center Support 33
 - constraints inferred by the system 81
 - dimensioning 27
 - dynamic Group with One Prim 165
 - Dynamic Group with Three Prims 172
 - Dynamic Group with Two Prims 167
 - Family of Parts 11
 - for a CPI named group 149
 - how to deal with fillets 68
 - In-sheet DEF Command 105
 - intersection and tangency constraints 86
 - load parametric symbol with table 144
 - Object With Angular Notch 71
 - of a table producing three parameterized versions of an object 126
 - of a Table with 4 entries/rows and 3 variables 121
 - of an in-sheet TBL command 125
 - of Preventing Parameterization with Group Line and Prim 162
 - parameterize a rectangle using a prim 53
 - parameterize a rectangle using intersecting baselines 57
 - parametric group 160
 - positioning a Prim 17
 - positioning Baselines 16
 - post-parametrization 186
 - Repeated Parameterization 193
 - replace dimension values with variables 95
 - Rotating Parametric Groups 171
 - Simulating a Working Mechanism 202
 - Simulating Linear Motion 195
 - Simulating Rotary Motion 197
 - Simulation of the Movement of a Lift Arm Assembly 13
 - Static Groups 163
 - Symbol Library 12
 - tangency constraints 83
 - three separate instances of one symbol 150
 - Use of Variables in Joist Definition 92
 - Using a Program to Simulate Motion 200
- F**
- example
 - how to load a parametric symbol 142
 - In-sheet Commands 97
 - Examples
 - of Over-dimensioned Drawings 37
 - of Parametric Viewbox Shapes 14
 - of Under-Dimensioned Drawings 36
 - simple parameterization 63
 - Expressions 93
 - usage 105
- G**
- fillet dimensions
 - set automatically 34
 - Fillet Grid Lines in detail 69
 - Filleting Settings 34
 - fillets 68
 - free parametric command text 113
- G**
- Geometric Constraints
 - what are - 76
 - Geometric properties 21
 - geometric tolerance 18
 - Getting Started 39
 - grid
 - building parametric - 46
 - deleting - lines 47
 - examine parametric - 48
 - parametric 17
 - when to draw? 47
 - grid lines 18
 - looking at 18

grid tolerance 18
Group Elements 215
Groups
 toolset 162

H

HIT 109
Hittable 109
Horizontal and vertical dimensions 32

I

Inferred Constraints 77
In-sheet Error Message Elements 215
Instance Group creation 148
INVIS 109
Invisible 109

L

LAY Command 110
Layer Conventions 108
Layer Properties 108
 changing 110
LCIS Variable 98
Linear dimensions 32
Load parametric symbol 135, 138
 procedure 139
load parametric symbol
 mistakes 140
Loading Parametric Symbol
 popup menu 141
Logical Operators 106

M

MEDUSA Applications 206
Mistakes When Loading Symbols 140

N

NC Definitions 206

O

object
 parameterize 50
orthogonal view prim 16
Over-dimensioning 37
overview
 simple parameterization of a part 40

P

PAR BAS Switch 79
PAR BAS switch 115
PAR CIR switch 115
PAR COL switch 114
PAR UND switch 116
PAR VAR 30
Parallel and perpendicular dimensions 32
Parameterize the Master Drawing 22
parameterizing an Object 50

Parametric Design
 and 3D Modelling 207
 and MEDUSA Applications 206
 what it is used for 10
Parametric Element Types 216
Parametric Elements Control 111
Parametric Grid 17
Parametric Group
 create 162
 Elements 160
Parametric Point Function 43
Parametric Switches 112
Parametric Symbols
 creating 134
 load 138
 save 142
 With Tables 144
Parametric Symbols and tables 122
Parts of a table 118
place for a reference point 16
point functions at the ends of baselines 44
Point Functions for baselines 78
Popup while Loading Parametric Symbol 141
postparam.she 188
Post-Parametric demo sheet 188
post-parametric module
 purpose 176
Post-Parametrization dialog 177
ppcb_array_copy 183
ppcb_array_copy_angled 182
ppcb_copy_and_mirror 184
ppcb_copy_rotate 185
ppcb_copy_rotate_pcd 184
Prefix Texts 49
prepare a drawing for parameterization by dimensioning 25
prim of type PVG 15
print screen 204
Printing documentation from Portable Document Format
 (PDF) files 8
PRO 109
Procedure for creating post-parametric definitions 180
procedure for simple parameterization of a part 40
Program to Simulate Motion, example 200
Protected 109
PVG prim 15

Q

Query LCIS Variables 99

R

Radius Dimensions 95
reconfiguring the Geometry of a Part 10
reference point 15, 43
 positioning 16
Repeated Parameterization 190
Repeated Parameterization (Example) 193
rotating 21
Rotating Parametric Groups 171
Rules for Creating Variable Names 94
running simulation buttons 192

S

- Screen Dump 204
- Selects a variant from a table and parameterizes it, tool 131
- Selects elements or groups at sheet level, tool 136
- Sets options for fillet behavior during parameterization 34
- Sheet Metal Design System 206
- Simple Parameterization Examples 63
- simulate the movement of a mechanism, introduction 190
- Simulates mechanism motion, tool 191
- Simulating
 - a Working Mechanism (example) 202
 - Mechanical Movement, overview 12
 - Rotary Motion (example) 197
 - summary 190
- Simulating Linear Motion, example 195
- Special Cases 22
- Special Drafting Techniques 18
- specifying constraints explicitly 80
- specifying how the system calculates new dimension values 30
- Static Baselines 78
- static baselines 15
- Static Groups, example 163
- structure of a table 118
- Suffix Texts 49
- summary
 - simulation 190
- switches 112
 - PAR BAS 115
 - PAR CIR 115
 - PAR COL 114
 - PAR UND 116
- Symbol
 - preparation 135
- Symbol Elements 215
- Symbol Libraries
 - creating 11
- Symbol with Table 136
- Symbols
 - load - Using CPI NamedGroups 148
- symbols
 - mistakes when loading 140

T

- Table
 - accessing values from a - 124
 - creating 120
 - Elements 119, 215
 - structure 118
 - Use With Parametric Symbols 136
 - Values 122
 - where to place 123
- tables
 - and Parametric Symbols 122
 - Variables and Expressions 122
- tangential arcs 68
- TBL Command 124
- Tolerance Settings 30
- tool
 - Control Parametric Elements 111
 - Convert the selected circular arc into a tangent point

- arc 68
- Creates a parametric table 121
- Creates free parametric command text 113
 - usage 96
- Creates parametric command text 113
- Creates parametric symbol attachment text 135
- Creates parametric tolerance text 29
- Creates parametric viewbox lines 41
- Creates parametric viewboxes 41
- Creates static baselines 80
- Creates user LCIS variables 98
- Display Post Param Dialog 177
- Display the current settings of the parametric line styles 216
- Displays the current settings of the parametrics system 112
- Displays the current tolerance settings 29
- Draws baselines inferred when the baseline switch is on 77
- Load parametric symbol 135, 138
- Parameterizes the geometry 50
- Parameterizes the geometry then immediately undoes 50
- Selects a variant from a table and parameterizes it 131
- Selects elements or groups at sheet level 136
- Sets options for fillet behavior during parameterization 34
- Simulates mechanism motion 50, 191
- toolset
 - Creating Reference Points 43
 - Creating Viewboxes 41
 - Drawing Grids 46
 - for CPI Named Groups 148
 - for Groups 162
 - for Parameterizing 50
 - for Tables 120
 - parametric switches and command texts 112
- tooltray 23
- Transformable 109
- TRN 109

U

- UNDEL 109
- Undeletable 109
- Under-dimensioning 36
- UNHIT 109
- Unhittable 109
- UNPRO 109
- Unprotected 109
- Untransformable 109
- Untransformable Layers 123
- UNTRN 109

V

- values
 - access - from a table 124
- values in tables 122
- Variable Tolerances
 - how to create 31
- Variables
 - save within sheet 100
- variables 92
 - assign values to - Using in-sheet commands 96

assign values to LCIS - 98
rules for - names 94
Variables and Expressions in Tables 122
Variant 131
vertical dimensions 32
viewbox 14
 create 41
VIS 109
Visible 109

W

Warning Messages 217
When to Draw the Grid 47
where to place a table? 123
Why Use in-sheet Commands? 96
workflow
 parameterize a rectangle using a prim 53
 parameterize a rectangle using intersecting baselines 57



A B C D E F G H I J K L M N O P Q R S T U V W X Y Z