OfficeCAD

User Guide

Automated Design & Drafting

for AutoCAD Professionals

OfficeCAD
2120 Ellis St., Bellingham, WA 98225 360-733-1733
www.officecad.com

OfficeCAD LICENSE AGREEMENT

Thank you for purchasing an OfficeCAD software license.
YOU SHOULD CAREFULLY READ THE FOLLOWING TERMS AND CONDITIONS BEFORE INSTALLING OR OPENING THIS PACKAGE AND/OR BY USING THE SOFTWARE. OPENING THIS PACKAGE OR USING THE SOFTWARE INDICATES YOUR ACCEPTANCE OF THESE TERMS AND CONDITIONS. IF YOU DO NOT AGREE WITH THEM, YOU SHOULD PROMPTLY RETURN THE PACKAGE TO THE LOCATION WHERE YOU PURCHASED THE SOFTWARE, UNOPENED WITH PROOF OF PURCHASE, AND YOUR MONEY WILL BE REFUNDED.

OfficeCAD provides this program and licenses its use. OfficeCAD retains the ownership of this product. This Agreement shall be governed by the laws of the state of Washington.

OfficeCAD LICENSE
Permitted Uses/You May:

- Use the software on any computer provided the software is used on only one computer and by one user at a time.
- Site License Terms: If you have an OfficeCAD site license you may make an unlimited number of copies subject to restriction by company name and division.
- Copy the program into any machine readable or printed form for backup or modification purposes in support of your use of the program on a single machine.
- Modify the program and/or merge it into another program for your use on the single machine. (Any portion of this program merged into another program will continue to be subject to the terms and conditions of this Agreement.)

You must reproduce and include the copyright notice on any copy, modification or portion merged into another program.

Prohibited Uses/You May Not:

- Make copies of the documentation or software, except as noted above.
- Distribute, rent, sub-license, transfer, or lease the software or documentation.
- Run OfficeCAD in a multi-user or Internet application without permission from OfficeCAD.
- Alter, modify or adapt the software or documentation, including, but not limited to, translating, decompiling, disassembling, or creating derivative works.

This license and your right to use the software automatically terminate if you fail to comply with the provisions of this License Agreement.

TERM
The license is effective until terminated. You may terminate it at any other time by destroying the program together with all copies, modifications and merged portions in any form. It will also terminate upon conditions set forth elsewhere in this Agreement or if you fail to comply with any term or condition of this Agreement. You agree upon such termination to destroy the program together with all copies, modifications and merged portions in any form.

IF YOU TRANSFER POSSESSION OF ANY COPY, MODIFICATION OR MERGED PORTION OF THE PROGRAM TO ANOTHER PARTY, YOUR LICENSE IS AUTOMATICALLY TERMINATED.

TRADE SECRET
You acknowledge that the software constitutes valuable trade secret information that is the exclusive property of OfficeCAD.

LIMITED WARRANTY
OfficeCAD warrants for a period of ninety (90) days from the date of original delivery to you that the program will substantially conform to the published specifications and to the documentation, provided that it is used on the computer hardware and with the operating system for which it was designed. OfficeCAD warrants the CD on which the program is furnished to be free from defects in materials and workmanship under normal use for a period of ninety (90) days from the date of delivery to you as evidenced by a copy of your receipt. This warranty gives you specific legal rights. You may have other rights that vary from state to state.

DURING THE WARRANTY PERIOD, IF THE SOFTWARE DOES NOT PERFORM AS WARRANTED, YOUR EXCLUSIVE REMEDY SHALL BE TO SEND THE SOFTWARE TO OFFICECAD WHICH SHALL, AT ITS OPTION, EITHER REFUND TO YOU THE PRICE PAID OR REPAIR OR REPLACE THE SOFTWARE.

To the extent permitted by applicable law, OfficeCAD disclaims all other warranties, either expressed or implied, including, but not limited to, warranties of merchant ability and fitness for a particular purpose. OfficeCAD makes no warranty as to title. No OfficeCAD dealer, distributor, agent, or employee is authorized to make any modification or addition to this warranty.

LIMITATION OF LIABILITY
Because programs are inherently complex and may not be completely free of errors, you are advised to validate your work. TO THE EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT WILL OFFICECAD OR ANY OF ITS PRINCIPALS OR AGENTS BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES WHATSOEVER ARISING OUT OF THE USE OF OR INABILITY TO USE THE PROGRAM OR DOCUMENTATION, even if advised of the possibility of such damages. Specifically, OfficeCAD is not responsible for any costs or damages including, but not limited to, those incurred as a result of lost profits or revenue, loss of use of the computer program, loss of data, the costs of recovering such programs or data, the cost of any substitute program, claims by third parties, or for other similar costs. IN NO CASE WHATSOEVER SHALL OFFICECAD'S LIABILITY EXCEED THE PRICE PAID FOR THE LICENSE TO USE THE SOFTWARE.

SOME STATES DO NOT ALLOW THE LIMITATION OR EXCLUSION OF LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES SO THE ABOVE LIMITATION OR EXCLUSION MAY NOT APPLY TO YOU.

Should you have any questions concerning this Agreement, you may contact OfficeCAD by writing to: OfficeCAD, 2120 Ellis St., Bellingham, WA, USA 98225.

You acknowledge that you have read this Agreement, understand it and agree to be bound by its terms and conditions. You further agree that it is the complete and exclusive statement of the agreement between us which supersedes any proposal or prior agreement, oral or written, and any other communications between us relating to the subject matter of this Agreement.

OfficeCAD Manual Contents

Chapter 5 – Arcs, Circles and Polylines 57

Chapter 6 – Blocks and Layers 73

Chapter 7 – Leaders, Text, Attributes 79

Chapter 8 – Scale and Tscale 83

Appendix A – Using OfficeCAD with Vdraft . . . 143

Appendix B – Error Messages 149

Chapter 1 – Introduction

Capabilities of OfficeCAD

Welcome to OfficeCAD. OfficeCAD software makes automated drawing creation available to anyone who is familiar with operation of an Internet browser or Microsoft Office products. Any salesperson, clerk, secretary, shop floor person, manager, or draftsman who can enter data in a browser, Access, Word or Excel can now create new AutoCAD drawings automatically. OfficeCAD will even create new custom, detailed, CAD drawings from your existing product configurator.

OfficeCAD is also revolutionary in providing Internet-based 2D and 3D rendered CAD drawing automation. Now, CAD drawings can be created and viewed with only an Internet browser.

OfficeCAD Features

1. Ability to create new drawings from an Internet browser on your server.
2. VBA (Microsoft Visual Basic for Applications) based automation tools to run existing and new designs under Excel.
3. Library of Visual Basic for Applications (VBA) functions available for use under most any VBA compatible application.
4. AutoCAD Release 2000, 2000i, R14 and Vdraft 2.2 drawing compatibility.
5. Windows-based help files with lots of graphics and examples.

For Synthesis Parametric users:

- Dialog option to convert Synthesis master drawings to OfficeCAD format.
- Functions to convert your pre-existing Synthesis parametric SpecSheet (DEZ) designs to Excel files automatically.
- Easily convert existing DEZ designs. (Master drawings are upwardly compatible.)
- AutoCAD 2000, 2000i and R14 compatible drawing output.
- Free technical help on line on our OfficeCAD website.
- Automated conversion of SpecSheet DEZ file designs into Excel format.

OfficeCAD Benefits Summary

- Easy to use. Graphic approach to parametric master drawing creation.
- Achieve design error reduction through design and drafting standardization.
- Create dramatic time savings and improved quality. Each time an error is eliminated, you save time and money, and your customers benefit.
- Create new sales or submittal drawings in seconds, without being a draftsman.

- Open up drafting to non-CAD experts including clerks, salespeople and anyone who can run Excel. AutoCAD drawings are created without draftsmen.
- Increase sales efficiency, eliminate errors or omission and calculation.
- Create drawings on laptops in the field. One visit to close the sale.
- Create instant, new AutoCAD drawings from Excel.
- Automated production of AutoCAD drawing sets generated.
- Create one or 2000 drawings at one time for sales, numerical control and/or production use.
- Standardize design and drafting process for ISO 9000 quality control.
- Open architecture fits in with your automation efforts. OfficeCAD will work with most any program that can create the ASCII specification file required for creating new drawings.
- Powerful CAD: OfficeCAD can create drawings 24 hours per day without a draftsman or AutoCAD. This translates into real power CAD. Users can create over 2000 new drawings at one time.

Features and Highlights

Create new drawings from your web server instantly, with no draftsman, CAD program or viewing program required. A "pure play" application.

- Create new AutoCAD drawings with Excel automatically.
- Use the Excel graphic user interface and Virtual Basic for Applications for your designs under Windows 98/95 and NT.

Hardware and Software Requirements

It is assumed that you are using OfficeCAD with AutoCAD 2000, 2000i or R14.

OfficeCAD requires the following minimum software and hardware configuration:

- Microsoft Windows 95/98 or Microsoft Windows NT.
- AutoCAD 2000, 2000i, R14 or Vdraft or for creation of drawings.
- Pentium computer.
- Minimum of 16 megabytes of RAM memory.
- Minimum of 22 megabytes of free hard disk space.
- Compact Disc (CD) drive.
- Mouse.
- Microsoft Excel 2000 or Excel 97.

Installing OfficeCAD

1. Vdraft or CADview is required to make output drawings.
2. AutoCAD or Vdraft is required to make master drawings. Master drawings can not be created using CADview.
3. For best results, both Vdraft or CADview and AutoCAD should be installed and working properly before installing OfficeCAD.
4. To create drawings from the Internet via your web server, OfficeCAD server version must be installed.
5. Insert the OfficeCAD CD into your CD drive.
6. Use Windows Explorer to find and run the OfficeCAD Setup.exe from the CD. This may happen automatically, depending on your system settings.
7. OfficeCAD will be installed to C:\OfficeCAD or a folder of your choice. To use OfficeCAD with AutoCAD, there must be a shortcut for AutoCAD with the “start in” property set to start in the OfficeCAD\Working folder, or, if different, the folder which contains the Acad.lsp file installed by OfficeCAD.

Getting Started

This manual is designed to get OfficeCAD working for you as soon as possible. If you are using OfficeCAD for the first time, we recommend that you read and *do* the examples and tutorials. You will need the tutorial drawing and design files provided on the CD.

It is assumed you are using the OfficeCAD interface within AutoCAD. If you are creating master drawings from Vdraft, see **Appendix A**.

Dialog Box Menu Interface within AutoCAD

With OfficeCAD, your menu is dialog box driven.

Once the AutoCAD application is loaded, you can pop up the menu by entering “OFFCAD” (or the shortcut “OC”) on the AutoCAD command line. This brings up the main menu.

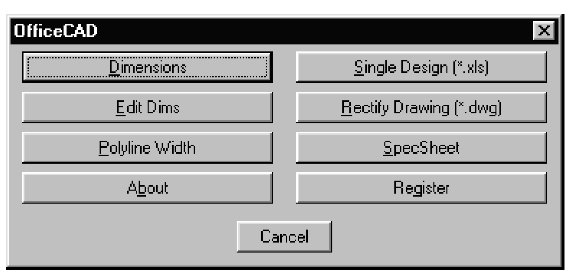


Fig. 1-1 OFFICECAD INITIAL DIALOG BOX

OfficeCAD Initial Dialog Box Options

Dimensions: This displays the Dimension Creation dialog box. (See following. Dimensions are covered throughout this manual, especially **Chapter 4**).

Edit Dims: This displays the Dimension Editing dialog box (see following).

Single Design (*.xls): This opens Excel and prompts for the worksheet that is to be calculated. Any new output drawings from that worksheet are created.

Type "OFFCAD" to bring up the OfficeCAD dialog box in AutoCAD. Select the Single Design (*.xls) button. The user will be asked whether to save any changes to the drawing. The drawing will be closed. A new drawing will be opened. A dialog box will appear prompting for an Excel workbook to be selected. This workbook needs to be an OfficeCAD workbook. Once the workbook is opened, the user will be asked for the worksheet that is to be calculated. The worksheet will be calculated and any new drawings from the worksheet will be created.

Rectify Drawing (*.dwg): This updates (rectifies) the current drawing according to the dimension text.

Type OFFCAD to bring up the OfficeCAD dialog box in AutoCAD. Select the Recify Drawing (*.dwg) button. The user will be asked whether to save any changes to the drawing. The drawing will be closed. A new drawing will be opened. A dialog box will appear prompting for the selection of a drawing to be rectified. The drawing will be rectified, and a new output drawing will be saved and opened in AutoCAD. The original drawing will not be changed. The new output drawing filename will contain the name of the original drawing with "_rec" appended to the end of the name. The new drawing will be saved in the same folder as the original drawing. (See examples later in this chapter.)

Polyline Width: This displays the Parametric Polyline Width dialog box (see following).

SpecSheet: The displays the OfficeCAD SpecSheet dialog box (see following).

About: This displays the About OfficeCAD dialog box (see following).

Register: This displays the Register dialog box (see following).

Cancel: This closes the OfficeCAD initial dialog box without executing any OfficeCAD commands or functions.

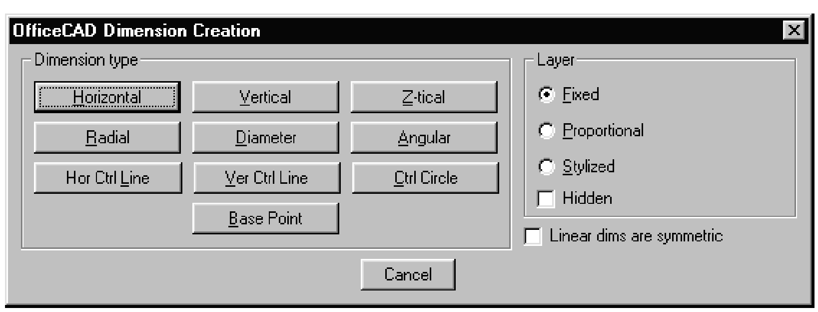


Fig. 1-2 OFFICECAD DIMENSION CREATION DIALOG BOX

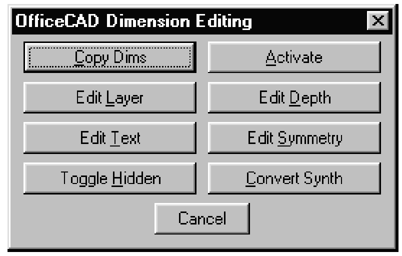


Fig. 1-3 OFFICECAD DIMENSION EDITING DIALOG BOX

Copy Dims: When copying OfficeCAD dimensions it is important to use this option rather than the regular "copy" command. When OfficeCAD creates a dimension or adds parametric polyline width data to a polyline, special information in the form of extended entity data is added. The regular "copy" command does not correctly update this information in the copies it makes. By selecting the "Copy Dims" button, the objects are copied and correctly updated. Note that when Radial or Diameter dimensions are copied, all the objects that are part of the selection set of that dimension are copied as well.

The "Copy Dims" button can be used to copy any object in the drawing. Objects with OfficeCAD data are copied and updated. Regular objects are simply copied. All the usual selection methods are supported: single selection, multiple selection, window, crossing, etc.

Edit Layer: This allows the change of the layer of an active dimension. The different layers which OfficeCAD uses for dimensions create different behaviors in the output drawing. To change the layer of an active dimension, select this option. A list of valid layers will be displayed.

Edit Text: This allows for the easy editing of the text of active dimensions. A dialog is displayed where the new text can be entered or a "default text" button can be checked.

Toggle Hidden: This turns on and off the active dimensions on hidden layers. The layers affected are DIMFIXHIDE, DIMPROHIDE, DIMSTLHIDE and DIM_DOT.

Activate: This is used to turn a regular (passive) dimension into an OfficeCAD active dimension. When a dimension is activated, a dialog box is displayed for the selection of the dimension layer.

Edit Depth: This is used to add, change or remove the depth of an active horizontal or vertical dimension. When prompted for the depth of influence, the user can either select a depth point, or hit return to indicate infinite depth of influence.

Edit Symmetry: The active horizontal and vertical dimensions have the option of being symmetric. A dimension is selected, a dialog box appears with a check box displaying the current status of the dimension. The box can be checked or unchecked, which then sets the symmetry flag for the dimension accordingly.

Convert Synth: Master drawings created by older versions of Synthesis™ can be converted to OfficeCAD master drawings. When "Convert Synth" is chosen, a prompt appears with the options of converting the current drawing or selecting a different drawing. The current option converts the active drawing. Select "select" to browse and select a drawing to convert. When a drawing is selected, it is opened and updated, but not automatically saved. Users can verify the conversion and/or save the drawing with a different name or to a different folder.

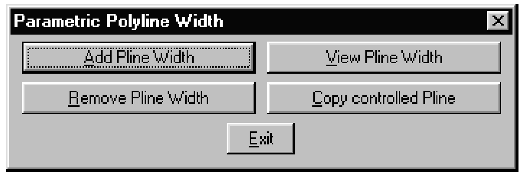


Fig. 1-4 PARAMETRIC POLYLINE WIDTH DIALOG BOX

Parametric Polyline Width: This dialog box features commands to control parametric width of polylines. (See **Chapter 5** for more information on polyline width.)

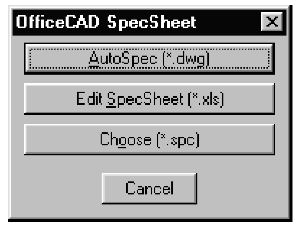


Fig. 1-5 OFFICECAD SPECSHEET DIALOG BOX

AutoSpec (*.dwg): This creates a new Excel worksheet based on your master drawing. An existing Excel workbook will be opened and the AutoSpec macro within that workbook will be called. The workbook needs to have been created using the OfficeCAD template (Officecad.xlt). Once the workbook has been opened, the function will call the AutoSpec function from that workbook.

Type "OFFCAD" to bring up the OfficeCAD dialog box in AutoCAD. Select the AutoSpec button. The user will be asked whether to save any changes to the drawing. The current drawing will be closed. A new drawing will be opened. A dialog box will appear prompting for the selection of an Excel workbook. The AutoSpec macro will be performed in the selected workbook. If Excel is already opened, the program will switch to Excel and call the AutoSpec macro. (For more on using AutoSpec and Excel, see **Chapter 9**.)

Edit SpecSheet (*.xls): This opens an existing Excel workbook for editing. This workbook can be any workbook, not just workbooks created using the Officecad.xlt template.

Type "OFFCAD" to bring up the OfficeCAD dialog box in AutoCAD. Select the Edit SpecSheet button. A dialog box will appear prompting for the selection of an Excel workbook to edit.

Choose (*.spc): This opens a specification (SPC) file and runs the SPC file through OfficeCAD, creating any new output drawing from the SPC file.

Type "OFFCAD" to bring up the OfficeCAD dialog box in AutoCAD. Select the Choose button. The user will be asked whether to save any changes to the drawing. The current drawing will be closed. A new drawing will be opened. A dialog box will appear prompting for the selection of the SPC file to run. The SPC file will be run with OfficeCAD.

Fig. 1-6 ABOUT OFFICECAD DIALOG BOX

Register: When OfficeCAD is first installed, it runs as a 30 day trial version for one user. Beginning five days before the trial period ends, a warning will be given once a day. To extend the number of days and/or to change the number of users, you may request an activation code. When you select the Register button you will see:

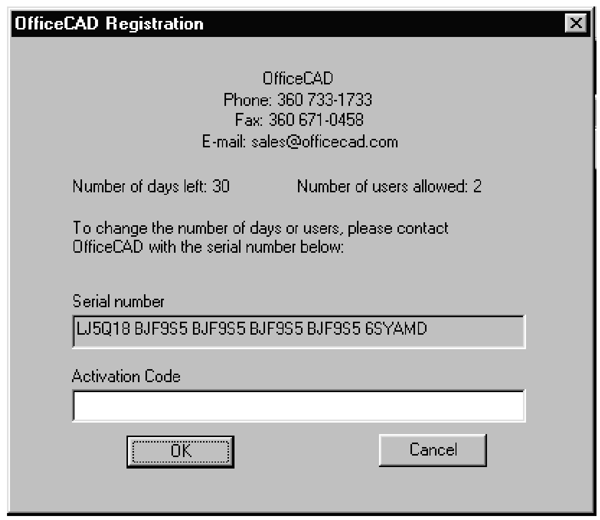


Fig. 1-7 REGISTER DIALOG BOX

This dialog box displays the current number of days remaining and the number of users allowed. It also displays the serial number. If you do not wish to make any changes, you may select “cancel.” (You may visit the Register dialog box at any time to check the number of days remaining and number of users.)

At any time, you may change the number of days remaining or the number of users. To do so, send e-mail to sales@officecad.com including the serial number, company name and contact person. To ensure accuracy, it is recommended that you highlight and copy the serial number from the dialog box, then paste it into the e-mail message. In return, you will receive an e-mail with an activation code that you can copy and paste into the Activation Code box in the Register dialog box. You may revisit the Register dialog box to see the revised number of days remaining and number of users. NOTE: if the values are correct, do not re-enter the activation code. Select “cancel” to exit the dialog box.

When the new “number of days” nears the end you will be warned 30 days before expiration, and again once per day beginning five days before expiration. To extend the number of days remaining, repeat the process described above, sending the new serial number and entering the new activation code.

OfficeCAD Examples

The examples in this section include three rectifiable drawings, “Head,” “Beam,” and “Pipe”. These examples are installed in the OfficeCAD\Working directory on your hard drive.

Parametric Design

Parametric design is a method in which a few outer parameters control the other parameters within the design. OfficeCAD uses three tools to implement the parametric design—the ***master drawing(s)***, the ***SpecSheet***, and the specification (SPC) file. The master drawing serves as a drawing template for the design. The SpecSheet accepts input from a variety of sources, then processes the input to update the other parameters for the design, and outputs the information to the specification file. The specification file and master drawing are used by OfficeCAD to create the new output drawing(s). The examples throughout this manual use an Excel workbook/worksheet for the SpecSheet; however, the SpecSheet can be any program able to create the specification file with the correct values for the parameters in the master drawing(s).

By using the parametric design capabilities of OfficeCAD, it is possible to create a master drawing from which many drawings can be generated. Instead of having to change every drawing by adding dimensions, values can be entered through the SpecSheet and a new drawing created automatically.

Modular Design

In OfficeCAD, parametric design is ***modular***. A simple parametric design can stand alone (as in a single master drawing/single output drawing design). A more complex parametric design can use multiple SpecSheets and multiple master drawings, and can create one or multiple output drawings from that design. Even with multiple SpecSheets, multiple master drawings and multiple output drawings, only one specification (SPC) file is used for the design.

The design examples in this chapter illustrate how to use these commands to create single and multiple parametric designs. Multiple SpecSheet parametric design is shown in **Chapter 12**.

Running OfficeCAD

1. Be sure OfficeCAD is installed.
2. During the installation of OfficeCAD, a shortcut for AutoCAD should have been created with the “start in” property set to c:\Officecad\Working. If you installed OfficeCAD into a different drive or folder, the “start in” property will reflect that change. To work from a different folder, edit an existing shortcut, or create a new shortcut. Set the “start in” property to the folder of your choice, and place a copy of Acad.lsp from the folder named “Working” in the main OfficeCAD folder into your folder.
3. At the AutoCAD command prompt, type “OFFCAD.”

NOTE: If you are a Synthesis software user converting to OfficeCAD it is recommended that you avoid using the following filenames when you save your drawings: ARROW, ATT_BLK and BORDER. These are specific DWG files that you may need to reserve.

Rectification Examples

A drawing with OfficeCAD dimensions can be modified by changing the values of the dimensions, either within the drawing (using OfficeCAD dimension editing) or from a SpecSheet outside the drawing. To rectify a drawing is to have it appear according to the new dimension values.

The first drawing, “Head,” is a simple introduction to rectification. The second drawing, “Beam,” illustrates the three types of active dimensions: fixed, proportional and stylized. The third drawing, “Pipe,” contains polylines.

“Head” Drawing – Rectification

In this example, you will rectify a simple drawing of a head. As you can see in the diagram below, only four dimensions are used. It is not necessary to dimension every feature of a drawing to ensure proper rectification. In this example the dimensions are proportional dimensions. Because of this, the lines controlled by each dimension will remain proportional to each other. The difference between proportional dimensions and the other dimension styles will be explained further later in this manual.

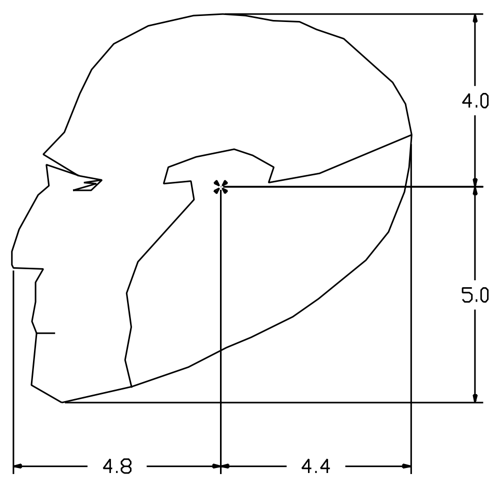


Fig. 1-8 THE HEAD MASTER DRAWING

1. Open the “Head” drawing from the OfficeCAD\Working directory.
2. Type “OFFCAD” at the AutoCAD command prompt. Next, select “Edit Dimension” to display the Dimension Edit dialog box. Pick “Edit Text” to change the text of one or more dimensions to a new value. Pick the text, then type in new text. Repeat this for each dimension to change. (Note that altering text does not alter the proportions of the drawing until you actually *rectify* the drawing with the new dimension text.)

3. After the new values for the dimensions have been entered, select “Rectify Drawing” from the OfficeCAD dialog box. You will be prompted to save the current drawing (“head”), which you will need to do. The drawing will be closed and you will be prompted for the drawing to rectify. Select the “head” drawing you just saved. The drawing will be rectified to appear in AutoCAD with the new dimension values.

If you would like to rectify the head drawing again, open the head drawing again and redo steps 2 and 3 above, as many times as you like.

OfficeCAD output drawings created this way will be given a temporary name. If you would like to save one of your newly-rectified drawings, save the drawing to a name other than the temporary name and different from the original name of “head” (e.g., call it “newhead”) or the master drawing will be overwritten. (The DWG suffix is added automatically.)

“Beam” Example

The beam example illustrates three styles of active dimensions: ***fixed***, ***proportional***, and ***stylized***. Each dimension style has a different effect on entities within its region of influence. (This is covered in greater detail throughout this manual.)

- With **fixed** dimensions, control points remain a *constant* distance from the dimension extension line *nearest* them.
- With **proportional** dimensions, the distance between control points increases or decreases in proportion to the change in the dimension value.
- With **stylized** dimensions, control points do not move during rectification; however when the text value is set to zero, then the entities controlled by that dimension are removed.

To rectify the Beam:

1. Bring up “Beam”, a drawing with three beams, each with a row of holes.
2. Edit the dimension text, setting each dimension to a value of 5 or greater.
3. From the OfficeCAD dialog box, select Rectify Drawing (*.dwg).

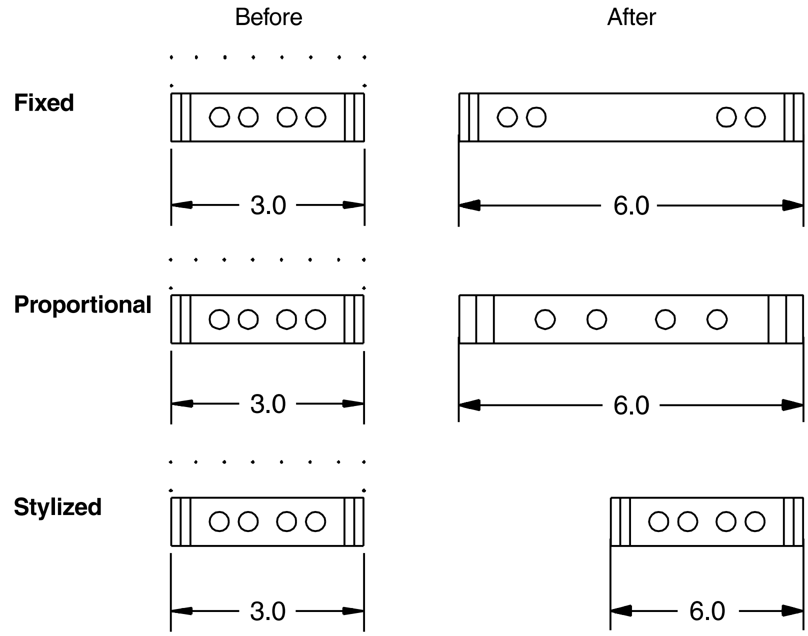


Fig. 1-9 BEAMS BEFORE AND AFTER RECTIFICATION

When you change the dimension value of the beams, notice how the holes behave. Each beam was dimensioned with a different dimension style: fixed, proportional, or stylized. You can also make a dimension ***visible*** or ***hidden***. (A hidden dimension is used if you don't want the active dimension to appear in the final drawing. Hidden dimensions can be turned on and off using the Toggle Hidden button in OfficeCAD, or by layer control.)

Depth of Influence – General Rules

In the beam drawing, note the dotted lines that connect to the ends of each dimension's extension lines. They mark the depth of influence of the active dimension; i.e., the area that affects the relocation of entities. Only the entities that lie within its indicated region are affected by the dimension. When prompted for a depth, press return to give an infinite depth of influence between extension legs. Dotted lines do not appear with infinite depth of influence.

You must select your dimension points carefully, as it is possible to create dimensions such that a design can't be rectified or drawn correctly. Bad dimensions produce bad drawings. Use grid or snap to assure that dimension points are placed accurately.

"Pipe" Drawing

To see the pipe drawing, use the same steps as in the first two examples:

1. Open the "Pipe" drawing. From the OfficeCAD dialog box select Edit Dims, Edit Text to set the dimensions to new values.
2. Rectify the "Pipe" drawing with OfficeCAD, using the Rectify Drawing button. Observe how the new values affect the rectified drawing.
3. Experiment with other values.

The curve of each elbow requires a constant 1.5 unit radius, to complete its bend. In the following example, if you pick dimensions which locate the elbows' centers less than three units apart, a bad drawing results.

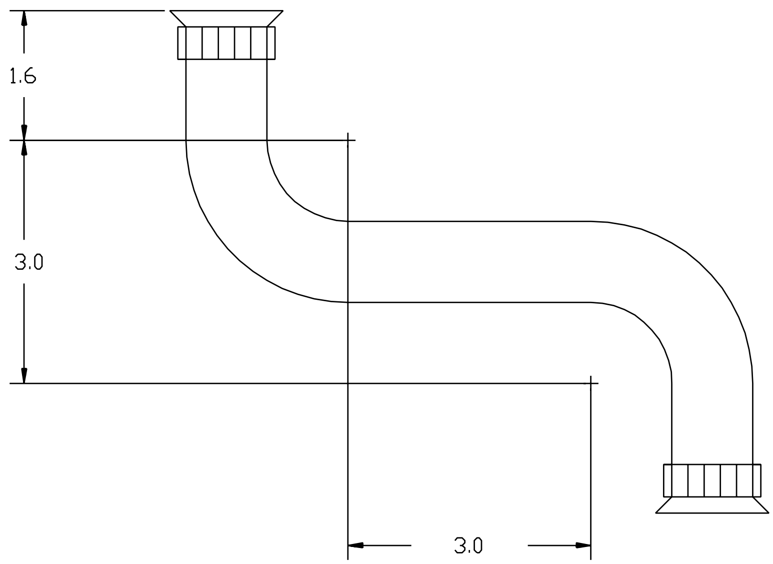


Fig. 1-10 THE ORIGINAL PIPE DRAWING

The "Pipe" example is drawn with polylines. The control points of the arcs are their centers of curvature, which are dimensioned to with horizontal and vertical dimensions. As you will see, each of these dimensions has a very different effect on the appearance of the drawing. (See **Chapter 5** for additional information on polylines.)

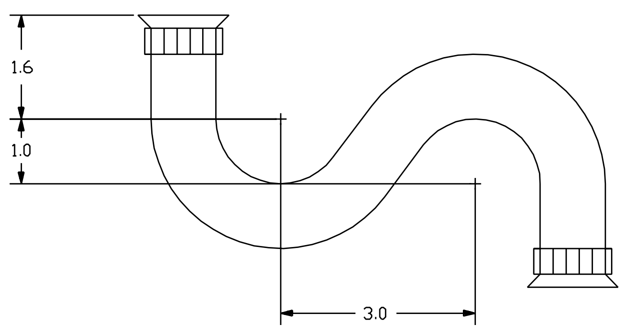


Fig. 1-11 THE MODIFIED PIPE DRAWING

Chapter 2 – Introduction to CAD Drawing Generation and Control via Excel

This chapter explains how to create the two components of an **OfficeCAD** parametric design: the master drawing and the specification file generation program which will be referred to generically as a SpecSheet.

The tutorials assume that you are familiar with AutoCAD or Vdraft and that you have completed the examples in **Chapter 1**.

"Quad" is a very simplified example designed to introduce the way **OfficeCAD** works. The drawing tutorial points out some of the differences you will encounter when you draw using the **OfficeCAD** commands instead of the AutoCAD or Vdraft commands. The SpecSheet tutorial introduces you to SpecSheet editing, a powerful design tool which can be used inside of Excel.

When you finish this tutorial, you may want to skim through the rest of the manual for a quick look at the features not covered in the tutorial. **Chapters 3** and **9** are the reference chapters on master drawings and the SpecSheet, respectively, and **Chapter 5** contains reference material on arcs, circles and polylines.

Overview of Design / Draw Process

This figure illustrates how the components of the parametric design work together:

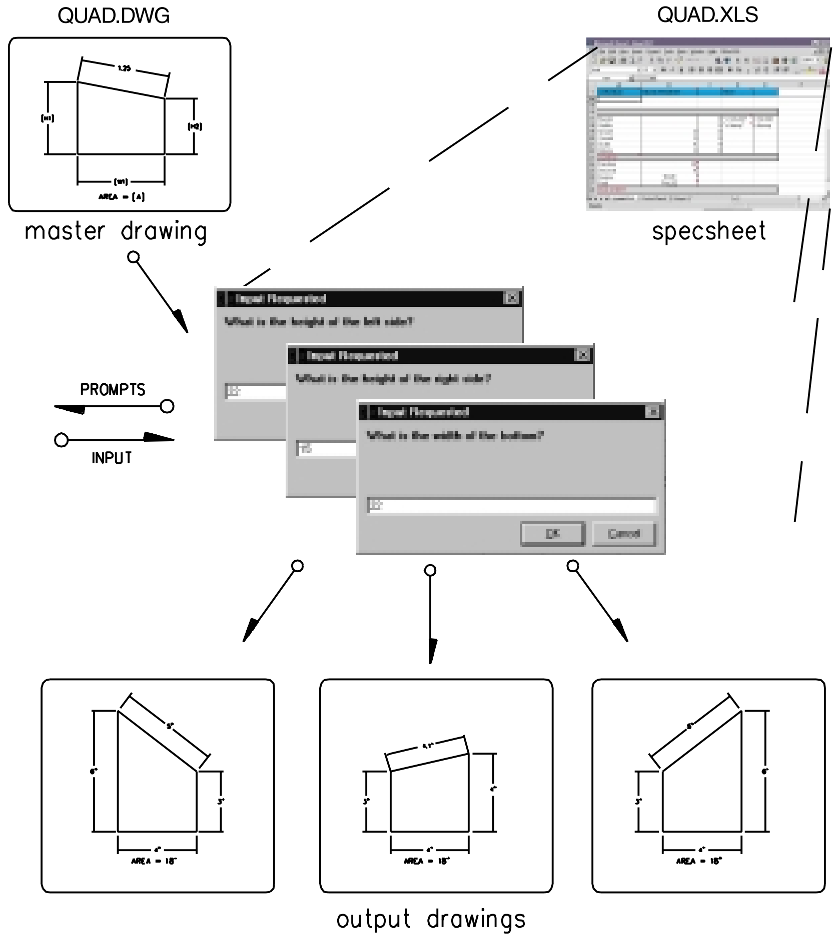


Fig. 2-1 PARAMETRIC DESIGN COMPONENTS

Parametric Control Features

OfficeCAD parametrically controls the replacement of text variables, replacement of dimension text, recalculation, entity disappearance, block and layer control.

Master Drawing Creation

The master drawing is a specialized rectifiable template drawing. A master drawing contains dimensions with variables in place of the default dimension values. New values are calculated by the SpecSheet, and the output drawing is rectified to the new values. The master drawing is used as a basis for generating drawings and can contain both unchanging and variable dimensions as well as unchanging and variable text.

A single master drawing can be used to represent a wide range of related designs. It can be drawn to allow for the most complicated case. From this original master drawing, OfficeCAD allows you to select which entities are to remain and which are to be eliminated. Thus, many drawings of varying complexity can be designed from a single master drawing.

In this part of the tutorial, you will create a simple rectifiable drawing and then change it to a master drawing. As noted earlier, a master drawing is a rectifiable drawing with one difference: there must be at least one variable that represents dimension text (or ordinary text) in the drawing.

Drawing Setup

Master drawing setup is done automatically upon using your first OfficeCAD command. Enter AutoCAD or Vdraft. In AutoCAD, type "offcad" (or the shortcut "OC") from the keyboard to bring up the OfficeCAD dialog box. If you are using Vdraft 2.2, see **Appendix A**.

Creating the "Quad" Drawing

Draw a quad to resemble the one in the diagram below. Set SNAP on and draw the box shape with straight, vertical sides, a base that's 90 degrees in relation to the sides, and a tilted top. From OfficeCAD, select "Dimensions," "Horizontal." Pick the two bottom corners of the quad as the end points, and select the text position below the quad. The "Text for dimension" prompt will appear. Hit enter. The "Select depth point" prompt will appear. Hit enter. The "Select text reference point" prompt will appear. Hit enter. From OfficeCAD, change the text to a variable by selecting Edit Dims, Edit Text. Pick the dimension. For the text, enter **[W1]** (be sure to enclose variable names in brackets). Now place vertical dimensions on either side of the quad, giving them the texts of **[H1]** and **[H2]** respectively. Next, create an aligned dimension on the top corners of the quad using the standard (not OfficeCAD) drafting tools.

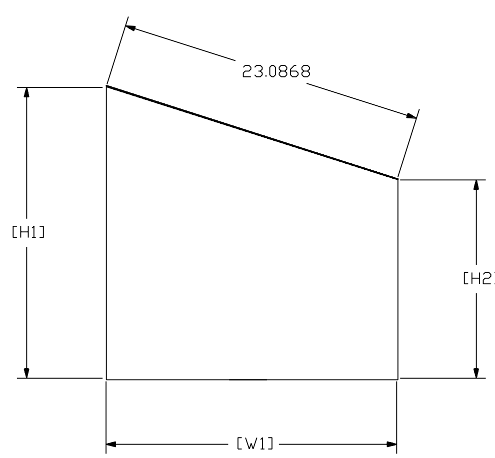


Fig. 2-2 QUAD DRAWING

Next, insert a base point from OfficeCAD, Dimensions, Base Point. In this example, position the base point at the lower left corner of quad. The base point determines the insertion point and center of rotation for the drawing. Dimensions will grow away from, or shrink towards, the base point.

Variable Text

Finally, put some text under the figure that will show the correct area of the quad when the parametric design is run. To do this, use AutoCAD to create centered text, select the location, and enter the following:

AREA = [A]

The master drawing is now complete. It should resemble the following diagram:

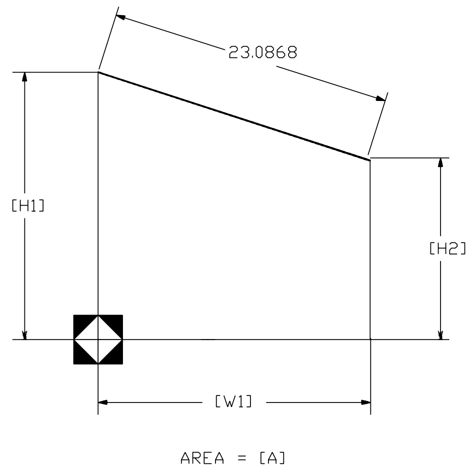


Fig. 2-3 THE QUAD MASTER DRAWING

The master drawing now contains four variables: [H1], [H2], [W1] and [A]. The first three will be user-defined parameters, the last one will be calculated from the first three. Save your finished quad drawing. The aligned dimension is passive, but its value will change according to the length of the top line in the output drawing.

The next section describes the main components of the SpecSheet and shows you how to create a SpecSheet to define the quad dimensions.

Creating a SpecSheet

The SpecSheet (special worksheet) is an optional way to create the specifications for the parametric design. A SpecSheet's job is to create your drawing configuration information into a simple ASCII file which OfficeCAD uses to generate your new drawings. The SpecSheet is a generic term for a design or configuration program. It could be an Excel worksheet, an Access database or an HTML page.

Through it, you create the interface to the end user. Like a spreadsheet, it calculates complex design values. In advanced designs a SpecSheet can command other SpecSheets to work together within one design, or to create multiple drawings.

Main Components

The following diagram illustrates the main components of a SpecSheet within an Excel workbook created using the OfficeCAD template.

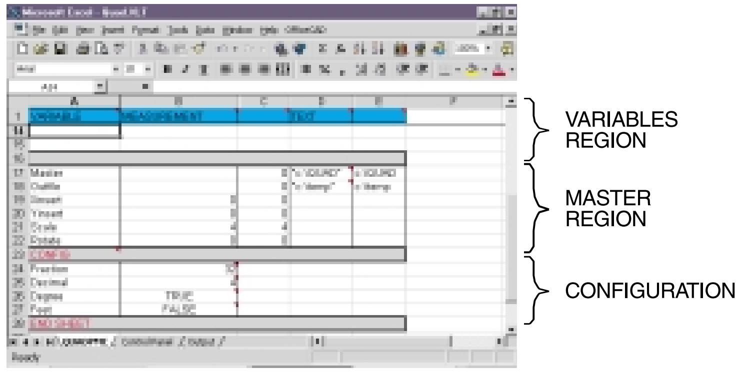


Fig. 2-4 EXCEL WORKSHEET FORMATTED AS A SPECSHEET

Editing the SpecSheet in Excel

Once a new workbook is created using the OfficeCAD template, one or more SpecSheets may be edited within that workbook in the same way as any other worksheet in Excel. To directly enter and edit a cell's contents, double-click on that cell. To replace the text within any cell, select the cell and type the new text. The existing text will be replaced. See the Help contents for Excel for further instructions on how to work with Excel.

Note that the layout of the worksheets must not be changed. The five columns, the column header row, the master region and the configuration region must remain as they are laid out in the OfficeCAD template. Rows may be added only between the column header row and the master region. Data in cells outside of column E and below the row labeled "END SHEET" below the configuration region will be ignored by the SpecSheet. The SpecSheet can import from and export to any of those cells outside the described area, but the data contained in those cells cannot be used directly by the SpecSheet.

Automatic SpecSheet Creation

OfficeCAD includes an Autospec routine in Excel for creating SpecSheets automatically. The Autospec routine will open a selected DWG drawing file, extract dimensions and text variables from the drawing, and create a generic SpecSheet that contains those variables.

A new SpecSheet design is made and loaded. Each variable is listed with a generic prompt. For Example:

VARIABLE	MEASUREMENT	TEXT
A	Prompt(" Enter value for A.")	STR_N(A#,2)

The default prompt string "Enter value for" can be changed at a later time when you edit the SpecSheet.

SpecSheet Columns

The SpecSheet is divided into 5 columns: one for variables and multiple SpecSheet commands, one for numeric formulas and prompts, one for text formulas and prompts, and two for output from the numeric and text formulas. Three of these columns are summarized here:

Cells in the variable column can contain:

- Variables which appear in the master drawing
- Variables local to the SpecSheet
- Messages printed to the user
- Internal comments that document the SpecSheet
- Commands to other SpecSheets in a multiple design

Cells in the measurement column can contain:

- Prompts to the user for numeric design values
- Formulas which calculate numeric design values
- Table functions to look up numeric values
- Formulas which convert text to numeric values
- Numeric values

Cells in the text column can contain:

- Formulas which convert numeric values to text strings
- Formulas which access design tables
- Prompts for text that will appear in the output drawing or for information necessary to the design
- Text strings

For all formulas and commands, for a variable “A”, the associated measurement (numeric) value is “A” or “A#”, and the associated text value is “A$”. If the “#” is left off of the variable, it is assumed to be the numeric value for that variable. The “$” must be used to denote the text value for any variable.

Creating the QUAD.XLS Workbook

Create the Quad.xls SpecSheet by using the OfficeCAD.xlt template. From the Excel menu select File, New. Click on the OfficeCAD tab and select the OfficeCAD.xlt icon. Double-click the icon or select OK to create the new file. Next, depending on your settings, you may need to select Enable Macros. You will be prompted to save the new workbook. Save it to the name Quad.xls.

Use Autospec from Excel or from AutoCAD to create the new SpecSheet (worksheet) for Quad.dwg.

Using Excel: From the Excel menu select OfficeCAD, New Sheets, AutoSPEC.

Using AutoCAD: Type “OFFCAD” or “OC” to bring up the OfficeCAD dialog box. Select SpecSheet, AutoSpec. You will be prompted to save your drawing. Select yes or no. If Excel is open, you will be switched to Excel. If Excel is not open, a dialog box will prompt for the file to open in Excel (in this case, Quad.xls). Double-click on Quad.xls, and it will be opened in Excel.

Both methods will start the Autospec routine. The AutoSpec dialog box will appear:

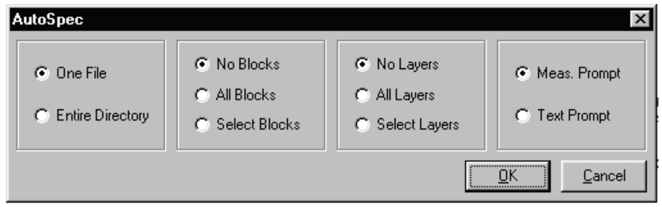


Fig. 2-5 AUTOSPEC DIALOG BOX

Choose the appropriate settings, and select OK. A dialog box appears prompting for a drawing file name. Select the quad drawing. The Dimension Text Format dialog box appears.

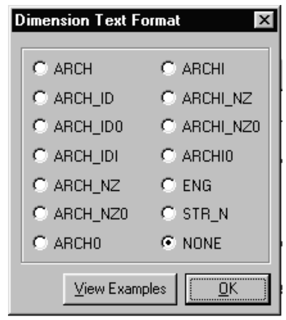


Fig. 2-6 DIMENSION TEXT FORMAT DIALOG BOX

These formats are described in **Chapter 9**. Choose the desired format, or choose none. In this example, choose STR-N. Quad.xls will contain fields populated with the variables from Quad.dwg. The Autospec routine is now completed.

Text Conversion

In Quad.xls, change the text of the prompts, or use the default text. In the row with the H1 variable, change the prompt in the measurement column to read:

PROMPT(“What is the height of the left side?”)

The text column in the row with the H1 variable will read:

STR_N(H1,3)

This is the default text conversion formula. A variable followed by the “#” has a numeric value. A variable followed by a “$” has a string value. When no symbol is used, the variable has a numeric value. If the string value is the intended value, the “$” must be used. In the worksheet from Autospec, the default decimal places value is 3 (from the “3” parameter).

This completes the design information for the H1 variable. For the other vertical dimension follow these steps:

In the row with the H2 variable, enter the following in the measurement column:

PROMPT (“What is the height of the right side?”)

For the horizontal dimension do the following:

In the row with the W1 variable, enter this measurement:

PROMPT (“What is the width of the bottom?”)

You now can specify a formula to calculate the area of quad. The area can be computed by dividing the sum of the sides by two and then multiplying by the width.

To compute the area variable, A, and add the words “square units” to the text, follow these steps:

In the row with the A variable, enter this formula in the measurement column:

((H1+H2)/2) * W1

Modify the formula in the text column to read:

STR_N(A,3) + " SQUARE UNITS"

In the formula above, the plus sign (+) is the text concatenation operator (it joins the text). The ampersand (&) also can be use as a text concatenation operator. When the SpecSheet is run, the text output for the A variable will be the area of the quad, and will output the value “407.00 SQUARE UNITS”.

Using the SpecSheet

A SpecSheet can be set up to provide messages and prompts to the end user, to calculate design values, and to control the insertion of subordinate drawings into a larger output drawing.

A SpecSheet is a powerful and flexible tool which can accept input from many sources: from the user; from design tables; or from other databases. (Using input from other programs is discussed in **Chapter 10**.) Based on input from the user, the SpecSheet replaces variables in the master drawing with dimension values or ordinary text and automatically creates the custom drawings rectified to the new values.

Running Your First Design from Excel

To execute the quad parametric design:

1. With the Quad SpecSheet open in Excel, select OfficeCAD, Calculate Current Sheet.
2. You will now see the run-mode of the design. Answer the prompts which you created in the SpecSheet.

   What is the height of the left side? <22.00>:
   What is the height of the right side? <15.00>:
   What is the width of the bottom? <22.00>:

3. Make sure the filename of the output drawing in the master region is different (or in a different folder) from the filename for the master drawing. Otherwise the existing drawing of that name will be overwritten without warning. Save or copy any outfile you want to keep, using a different filename or path, or, in the cell for the outfile in the text column, create a formula that will change the filename according to some user prompt or formula.
4. To view the new quad that you have designed, from the OfficeCAD menu in Excel, select Create New Drawing. If you entered values larger or smaller than the default, notice the change. If the dimensions or text appear to be too small or large, consider changing the SCALE# value in the master section of the SpecSheet.
5. Try running this example several times using different values.

Chapter 3 –
Overview of Design / Draw Process

This chapter summarizes the information on rectifiable and master drawings introduced in **Chapters 1** and **2**, and includes additional features and techniques not previously mentioned. The contents of this chapter apply to all dimensions in general, and to horizontal, vertical and z-tical dimensions in particular. See **Chapter 4**, Arcs, Circles and Polylines, for information specific to these entities.

Master Drawing Setup

To create a new drawing with **OfficeCAD**, do the following:

1. With OfficeCAD installed, start AutoCAD.
2. Create a new drawing, or open an existing drawing.
3. Use OfficeCAD dimensions on the drawing. Add the appropriate dimensions from the OfficeCAD Dimension Creation dialog box, accessed by selecting Dimensions from the OfficeCAD dialog box.
4. Convert existing dimensions to OfficeCAD dimensions by selecting Edit Dims from the OfficeCAD dialog box, then Activate from the Dimension Editing dialog box.

Prototype Drawing

The AutoCAD manual provides information on creating a prototype or default drawing.

Drawing Compatibility

OfficeCAD can be used with drawings created by all AutoCAD versions through Release 2000i.

Converting Synthesis to OfficeCAD Drawings

Use the “Convert Synth” command to use or modify an existing Synthesis drawing with OfficeCAD.

From the OfficeCAD dialog box, select Edit Dims, Convert Synth. This prompts for either the current drawing or another drawing. You can select DXF files as well as DWG files. Save the converted drawing. You may need to re-open the converted drawing in order to continue using OfficeCAD commands.

Converting Regular Drawings

To convert regular dimensions to OfficeCAD dimensions, these dimensions must be “activated.” To activate a dimension from OfficeCAD, select Edit Dims, Activate. This will change a dimension into an OfficeCAD controlling dimension. The Dimension Layer dialog box will appear, prompting for the dimension type (Fixed, Proportional, etc.).

Save the drawing to a new name, unless you choose to overwrite the unconverted drawing.

Entities and Control Points

Every drawing entity has one or more control points used for rectification. When a drawing is rectified, the control points are relocated by the active dimensions and the entity is redrawn. Table 3-1 lists some of the control points of the entities that OfficeCAD supports.

DRAWING ENTITIES AND CONTROL POINTS ***TABLE 3-1***

Point	1		The point itself
Line	2		Ends of the line
MLine	0	None	Not relocated
Arc	1,2		Center OR ends
Circle	1		Center point
Text	1,2	left center Aligned fit	Justification point(s)
MText	1+	A Sample of mtext that covers several lines. The insertion point is the control point.	Insertion point of text
Ellipse	1		Center of ellipse
Polyline	2+	See chapter four	Vertices, arc centers
LWPolyline	2+	See chapter four	Vertices, arc centers
Polymesh	2+	See chapter four	Vertices of polymesh
Spline	0	None	Not relocated
Inserted Block	1		Insertion point
Attribute	1	See "Inserted Block"	Blocks' insert point
AttDef	1,2	See "Text"	Same as for Text
Trace	2		Ends of centerline
Solid	4		Corners of solid
3DFace	4		Corners of 3DFace
XLine	1		Defining point of XLine
Tolerance	0	None	Not relocated
Ray	1		Defining point of ray
Leader	0	None	Not relocated
Shape	1		Insert point of shape
Dimension	1+	See each type	Dimension
Image	0	None	Not relocated
Wipeout	0	None	Not relocated
RText	0	None	Not relocated
Viewport	2		Center and target points
Hatch	0	None	Automatically erased
Region	0	None	Not relocated
3DSolid	0	None	Not relocated
AcadProxy	0	None	Not relocated

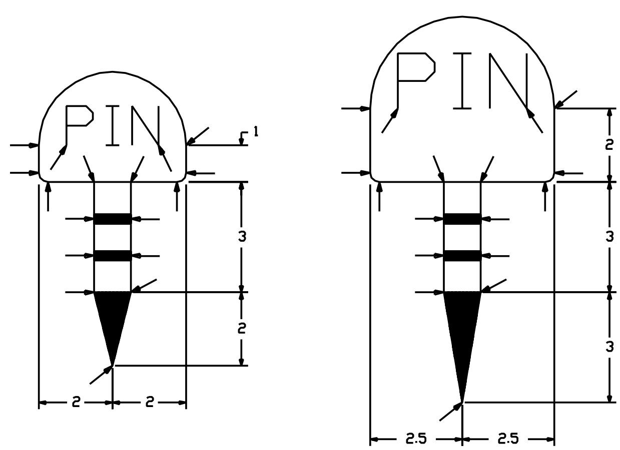


Fig. 3-1 CONTROL POINTS EXAMPLE

The figure above shows a pin before and after rectification. Note the position of its control points (indicated with leaders) as well as their movement with different dimension values. The pin is composed of the head, shaft, and tip. Its head is composed of 3 arcs, 3 lines and 1 aligned text entity. Its shaft contains 2 lines and 2 traces. The tip is a solid.

Aligned Text

In the control points example, aligned text was used to write the word "PIN". Note that its size changes with the distance between its control points.

Arcs

An arc maintains its shape (i.e., its original number of degrees of arc). Only its size changes to accommodate its control points. In the pin example, the shape of the semicircular arc at the top is always 180 degrees. Similarly, the fillets below are always 90 degrees.

In situations where you want an arc's shape to change (in order to maintain a tangency, for example), draw with polyline entities instead of arc entities. Polyline arcs can be controlled using angular dimensions (see **Chapter 5**).

It is often useful to use a pair of semicircular arcs to draw a circle rather than using a single circle entity. This method of creating a circle permits the use of linear dimensions to control the circle's size. This is shown in the next figure, where both views of the disk and the hole through it are controlled by the same linear dimensions. Note the locations of the control points as indicated by the leaders.

Arcs can be affected by radial and diameter dimensions, as are other entity types, such as circles and polylines. (See **Chapter 5** for a more complete discussion of circles and polylines.) Arcs with radial or diameter dimensions relocate by their centers. Arcs cannot have angular dimensions, but polyline arcs can.

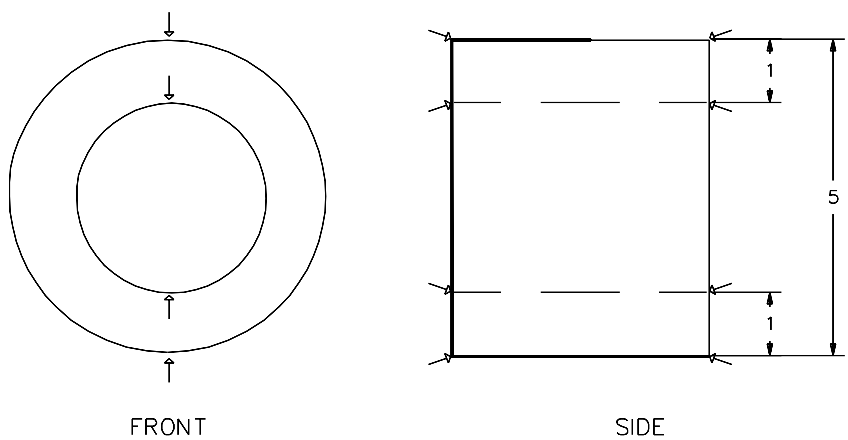


Fig. 3-2 EXAMPLE OF PAIRS OF ARCS USED TO REPRESENT CIRCLES

Blocks

An AutoCAD block entity can be inserted into an OfficeCAD drawing. Entities within a block are positioned as a unit and are not rectified individually. (This is especially useful when the entities are controlled by proportional dimensions. See the screws in the figure below.) A block may include text variables; however, the size of the text within a block remains constant even when the overall design is given a new scale factor.

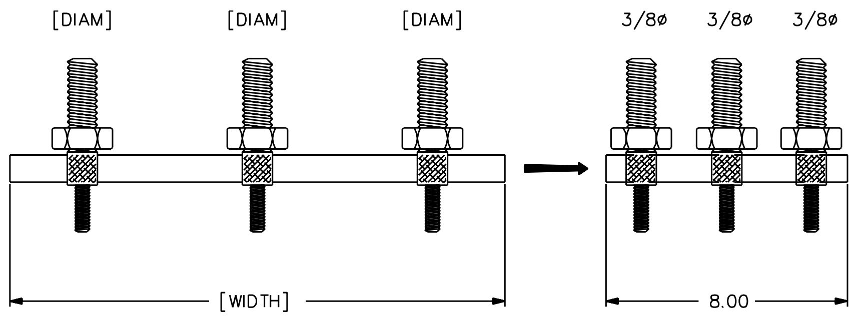


Fig. 3-3 INSERTED BLOCKS CONTROLLED BY PROPORTIONAL DIMENSIONS

Some applications use blocks to attach attributes to a portion of the drawing. Attributes can have variable text.

For more information on Blocks, see **Chapter 6**.

Text

Dimension Text

OfficeCAD supports the following formats for dimension text for all types of rectifiable dimensions, shown here with sample results:

1. Decimal: 15.50
2. Engineering: 1'-3.50"
3. Architectural: 1'-3 1/2"
4. Fractional: 15 1/2

More dimension text formats (available in Excel) are described in **Chapter 9**.

Passive Dimensions

Passive dimensions (ordinate, aligned and rotated, and dimensions not created or activated via the OfficeCAD menu) have three text options at creation:

- If you accept the default dimension text, the text is updated during rectification.
- If you insert a variable name, the dimension text is changed by the SpecSheet during rectification.
- If you insert any other text, the text is not changed during a design.

Variable Text

The text of a dimension, regular text, or block attribute of a master drawing may be a variable. A variable is alphanumeric, must be delimited by square brackets, and can be up to 23 characters long. It cannot start with a number.

For example, [width] or [W1A] are valid variable names in a master drawing, whereas [22] and [2nd] are not.

See **Chapter 7** for more details on variable text.

Layers

Parametric Layers

Instructions can be given to control layers in output drawings. Layers can be turned on, off, frozen or thawed as they would be by AutoCAD commands. Layers can be renamed; all entities on the layer being renamed will be moved to the new layer. Layers can be erased; when a layer is erased, all entities on that layer are also erased.

For more information on layers, see **Chapter 6**.

OfficeCAD Layers

OfficeCAD master drawings will create the following reserved layers as needed:

DIMFIX, DIMPRO, DIMFIXHIDE, DIMPROHIDE, DIM_DOT, DIMSTL, and DIMSTLHIDE.

Error Checking

If OfficeCAD encounters any errors during rectification, a numbered error message will appear on the screen. Write the number(s) down and refer to **Appendix B** for an explanation. Most errors involve improperly constructed dimensions.

Chapter 4 – Dimensioning Techniques: Overview of Geometry Control via Dimensions

Dimensioning Basics

Dimensioning with OfficeCAD can be quite different than with using AutoCAD alone. Often, what appears to be a poorly dimensioned drawing will actually result in properly formatted output drawings (and vice-versa) when passed through OfficeCAD. This section will touch on three areas of importance concerning dimensioning with OfficeCAD. They are:

- Rules for AutoCAD dimension variables
- Linear dimensioning rules
- Radial and angular dimensioning rules

OfficeCAD will redraw the **active** dimensions of the output drawings to the current variable settings. The active dimensions are all dimensions created or activated using the OfficeCAD dimensioning dialog boxes during the creation of the master drawings.

Rules for AutoCAD Dimension Variables

The AutoCAD dimension variable named DIMASO requires a setting of ON. Other than DIMASO, all variables may be changed to adhere to the drafting requirements of the particular application.

Linear Dimensioning Rules

The most powerful and easy to use OfficeCAD dimensions are the “linear” dimensions (horizontal, vertical and z-tical). These dimensions geometrically affect every entity on the drawing in all elevations. Horizontal and vertical dimensions have depth of influence (see “Depth of Influence of Dimensions” later in this chapter). Some important rules of thumb for linear dimensioning are:

- Dimension the drawing with linear dimensions and test linear movement prior to adding any radial, diameter or angular dimensions.
- Dimension from the base point in a continuous chain whenever possible. See “Dimension Chaining” later in this chapter for further information on chaining.
- Dimension for the appearance of the output drawings, not the master drawing.
- Always use snap to create accurate connections.

Linear Dimensions in Output Drawings

The linear dimension reconstruction appearance of the output drawings’ dimensions can be controlled to some extent. For example, the dimension line location is fixed at the length of the shortest dimension leg. When a linear dimension is part of a set of continuous dimensions, the shortest leg of the set is used.

If you specify a "text reference point" when creating a dimension, use that point in place of the shortest leg length.

The appearance of rotated and aligned dimensions (whose activity replicates an associative dimension when drawn with default text values) are similarly scaled by the shortest leg.

The text for aligned and rotated dimensions is re-created according to an algorithm comparable to AutoCAD's construction procedure, where text orientation is always orthogonal with respect to the drawing plane despite the rotation angle of the dimension.

Passive Dimensions

Passive dimensions are *ordinate*, *aligned* and *rotated*, and any dimension not created or activated via the OfficeCAD menu. They do not affect rectification. They are relocated by active *horizontal* and *vertical* dimensions. Passive dimensions are *not* affected by *radial*, *diameter* or *angular* dimensions. If a passive dimension has default text, the value is automatically updated during rectification. A passive dimension may also have variable text (see **Chapter 7**).

Active Dimensions

Active dimensions relocate entities during rectification. They *must* be created or activated via the OfficeCAD commands. The *horizontal*, *vertical* and *z-tical* dimensions change the location of an entity's control point(s). *Angular*, *radial* and *diameter* dimensions affect the size and shape of polylines, *radial* and *diameter* dimensions affect circles and arcs.

Horizontal, Vertical and Z-tical

Horizontal, **vertical** and **z-tical** dimensions support three dimension types that affect entity manipulation: fixed, proportional and stylized. For more on z-tical dimensions, see later in this chapter.

Radial and Diameter Dimensions

The radial and diameter dimensions affect circles, arcs and arcs within polylines. Arcs, circles and polylines are described more fully in **Chapter 5**.

Angular Dimensions

Angular dimensions control 2D polylines. See "Angular Dimensions" later in this chapter, and **Chapter 5** for more detail.

Dimension Types

Fixed Dimensions

For Horizontal, Vertical and Z-tical dimensions, entities' control points remain a constant distance from the nearest extension line when the dimension changes. For Radial and Diameter dimensions, see **Chapter 5**.

Proportional Dimensions

The distance between entities' control points changes in proportion to the change in dimension size.

The figure below illustrates the effect of choosing an inappropriate dimension type for a drawing. The original head drawing (with proportional dimensions) is on the left. The head in the middle also has proportional dimensions; the dimensions of the head on the right are fixed. When rectified to exaggerated dimensions, the proportional head in the middle retains its curves. The entities in the fixed head maintain their original relationship to their extension lines, resulting in a badly distorted figure.

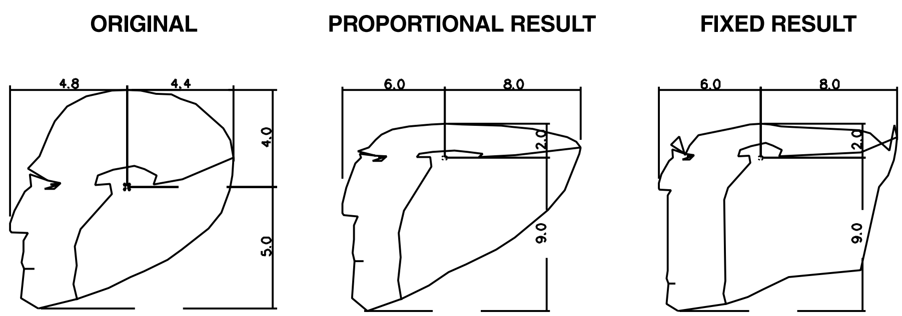


Fig. 4-1 FIXED AND PROPORTIONAL HEAD EXAMPLE

Stylized Dimensions

Stylized dimensions are relocated by other *horizontal* and *vertical* dimensions, unless they are given a value of 0 (zero), in which case they disappear and remove the entities related to them. The entities they control are not rectified, except when the dimension text is set to zero. (See "Deleting Dimensions" later in this chapter.)

Dominance of Dimensions

The dominance of a dimension determines which dimensions and entities are affected by other dimensions. Horizontal, vertical and z-tical dimensions are the major controlling dimensions. They have the highest dominance and are affected only by other horizontal, vertical and z-tical dimensions. They are relocated according to the chaining and nesting of dimensions. They affect the location and size of all types of dimensions and entities. (See "Dimension Chaining" later in this chapter.)

Radial, diameter and angular dimensions are relocated by horizontal and vertical dimensions along with the entity they are dimensioning.

The example below illustrates the effects of a radial dimension and a horizontal dimension on each other. Drawings using linear (horizontal and vertical) dimensions and angular dimensions may not appear as one might expect. For further information, refer to the following tutorial on angular dimensions.

Ordinate, aligned and rotated dimensions (and passive horizontal and vertical dimensions) have the lowest dominance. They are relocated by horizontal, vertical and z-tical dimensions only. These dimensions are passive and, therefore, do not affect anything. When an aligned or rotated dimension has been relocated, the text may or may not be updated, depending on how the user entered the text for the dimensions. If the default text was chosen, it is updated accordingly.

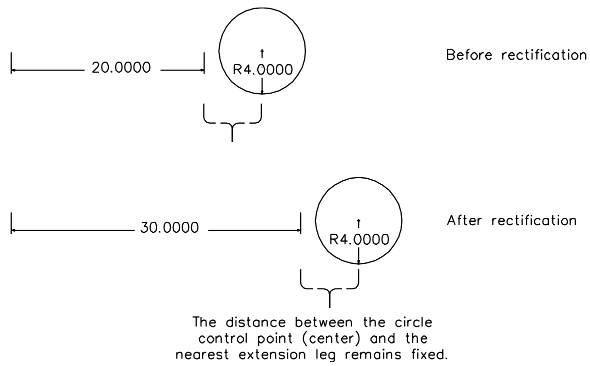


Fig. 4-2 RELATION BETWEEN HORIZONTAL AND RADIAL DIMENSIONS I

Notice that the circle and the radial dimensions were relocated as the horizontal dimension changed size.

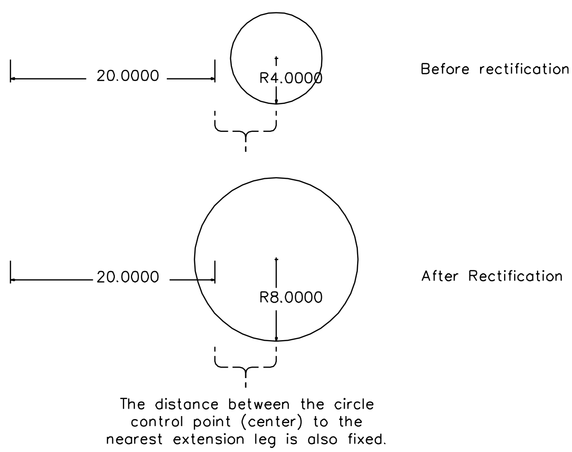


Fig. 4-3 RELATION BETWEEN HORIZONTAL AND RADIAL DIMENSIONS II

Notice that the horizontal dimension's location remained fixed regardless of the size of the radial dimension. The radial dimension has no effect on the horizontal dimension.

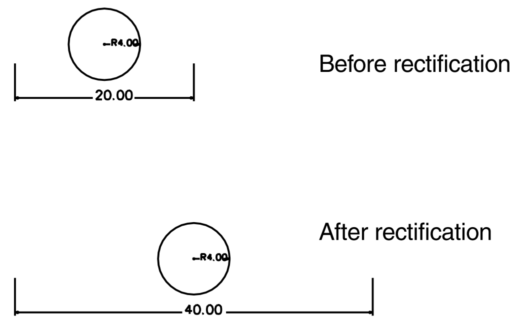


Fig. 4-4 RELATION BETWEEN HORIZONTAL AND RADIAL DIMENSIONS III

Notice that the horizontal dimension didn't change the size of the radial dimension or the entity. The dominance of vertical and horizontal dimensions only affects the position of radial dimensions.

Depth of Influence of Dimensions

Region of Influence

The effect of an active Horizontal or Vertical dimension is limited to entities whose control points fall within its region of influence. The region of influence is the area bounded by the dimension line, the two extension lines, and the depth-of-influence line. You select the position of the depth-of-influence line when the dimension is created. If no depth-of-influence line is specified, the effect of a dimension extends to the extents of the drawing in both directions.

Entities to the left and right of the region of influence of a horizontal dimension (HDIM) remain a fixed distance away from it. This can be seen with the letter "D" in the following region of influence example. Any entities beyond (in this case below) the depth-of-influence line do not move. (This is the case with letters "A", "B" and "C".)

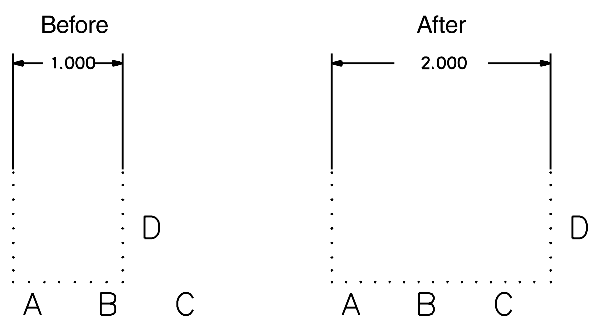


Fig. 4-5 REGION OF INFLUENCE EXAMPLE

Angular Dimensions

Angular dimensions only work with polylines. The dimension can be placed at the vertex of two straight segments or at the center of a poly-arc. Angular dimensions only affect the moving side of the polyline, locating it at the specified number of degrees from the original fixed side of the dimension.

For an angular dimension on a poly-arc, the arc maintains its radius and locates the moving endpoint to the new position on the arc (unless it is selected by a radial or diameter dimension, in which case both changes are applied).

When the angular dimension is at the vertex of two straight segments the effect is that of "freezing" the length of the moving side and rotating it to the new angle.

The fixed side segment or arc endpoint is free to conform to other polyline constraints.

Fig. 4-6 ANGULAR DIMENSION AT VERTEX

One potential problem is when the desired location of the angular dimension is not at the vertex of two straight segments:

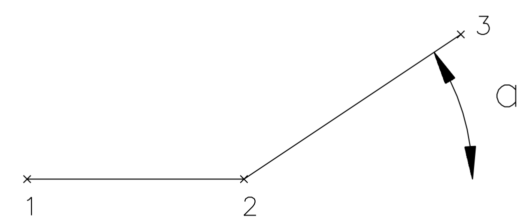


Fig. 4-7 ANGULAR DIMENSION NOT AT VERTEX
(Numbers indicate the order in which the vertex points are drawn)

One way to overcome this limitation is to draw the polyline with a very small leg in the direction you want the angle to start from:

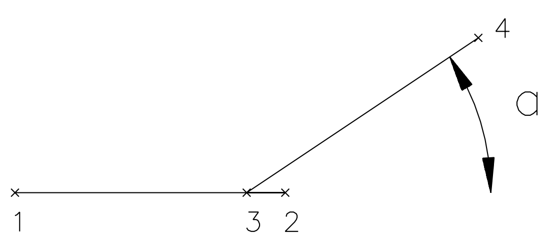
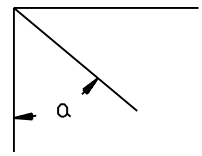


Fig. 4-8 EXTENDED POLYLINE LEG

Here is one way to provide angular dimensions:



Draw this part first, dimension it, and then add the third leg.

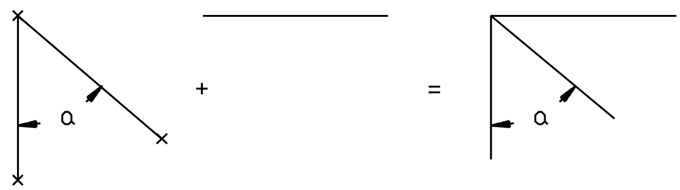
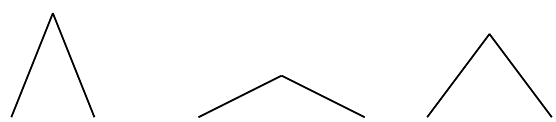


Fig. 4-9 ANGULAR DIMENSIONING

If you wish the angular dimension to have a symmetric effect, you can mimic it by using pairs of dimensions.

For a drawing with variation of:



A Master Drawing of the following form could be used:

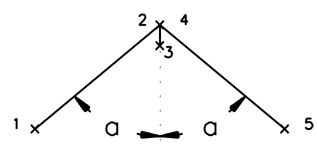


Fig. 4-10 SYMMETRY

For symmetric arcs, draw the arc as two segments and dimension each half.

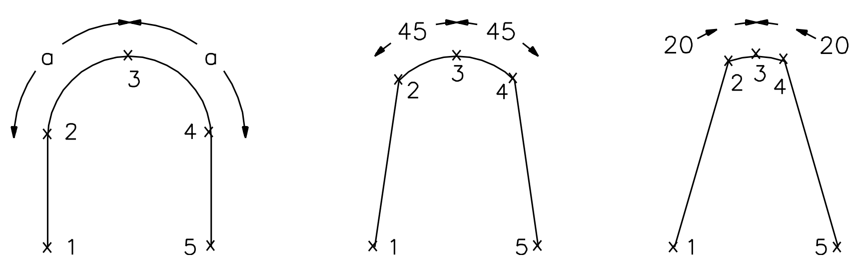
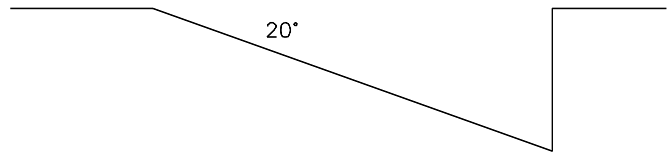


Fig. 4-11 SYMMETRIC ARCS

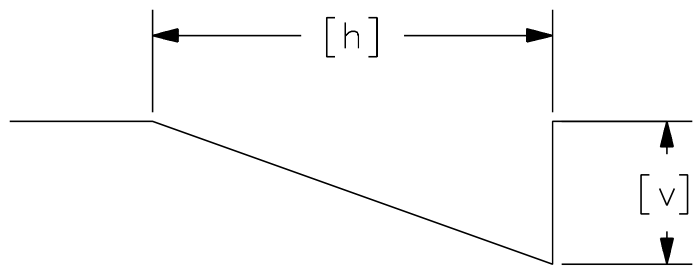
Some angular needs are better met by the use of trigonometry and horizontal and vertical dimensions.

One such example is the case where a given angle is to be maintained, as well as the tangency on a filleted corner.

1. Draw the polyline with straight segments with the critical side at the proper angle. (For this example, 20°.)



2. Dimension the top and side with horizontal and vertical dimensions, noting the default values (h is 2.8, v is 1.0).



3. Determine the radius of the fillet (0.2 in this case) and fillet the corner by selecting the two appropriate sides.

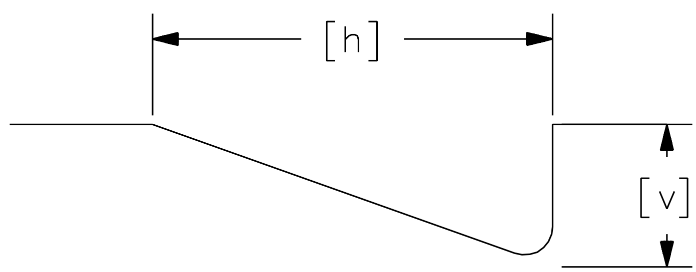


Fig. 4-12 MAINTAINING ANGLES

The goal—for any given h, change v so that the 20° angle is maintained.

One way to look at this problem is to consider that any output polyline will still be a 20° – 90° – 70° triangle (disregarding the filleted corner). So the ratio of h to v is the same for all cases. We set up the SpecSheet as follows:

Variable	**Measurement**
ratio	2.8 / 1.0
h	prompt("What is h: ")
v	h / ratio

Here are some results of this design:

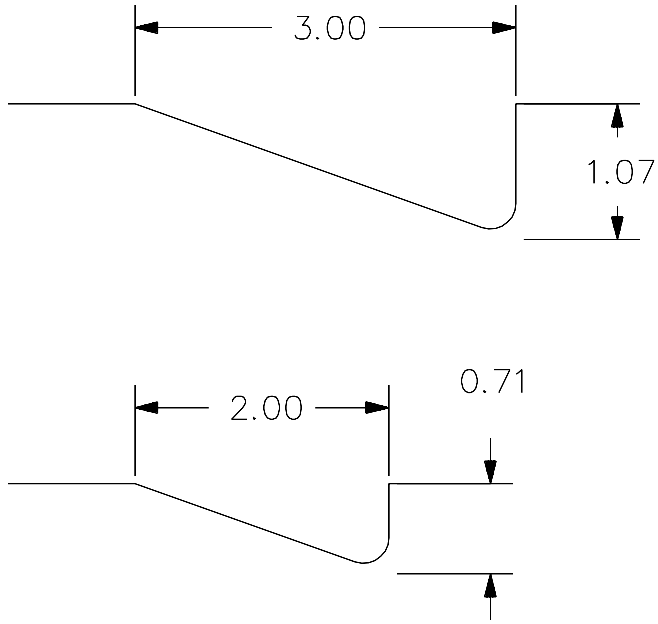


Fig. 4-13 TWO RESULTS OF THE DESIGN

Some drawings include curved shapes which do not need to be rectified to the full accuracy possible with OfficeCAD. An example is the head drawing in **Chapter 1**. In these cases, it is easiest to generate the curves with a series of straight line segments. Use a proportional dimension to control the resulting curved shapes.

Z-tical Dimensions Proportional / Fixed

In addition to dimensioning in the X and Y directions (horizontal and vertical), OfficeCAD supports **z-tical dimensions**.

The dimension function allows you to pick a proportional, fixed or stylized dimension type for control of entities in the Z direction.

When creating a dimension you will get the prompts as follows:

From OfficeCAD, select Dimensions to open the OfficeCAD Dimension Creation dialog box.

- Select the dimension layer: Fixed, Proportional or Stylized. In this example, Fixed is selected.
- The dimension layer will be visible unless a hidden dimension layer is selected.
- Select Z-tical, and apply the dimension.

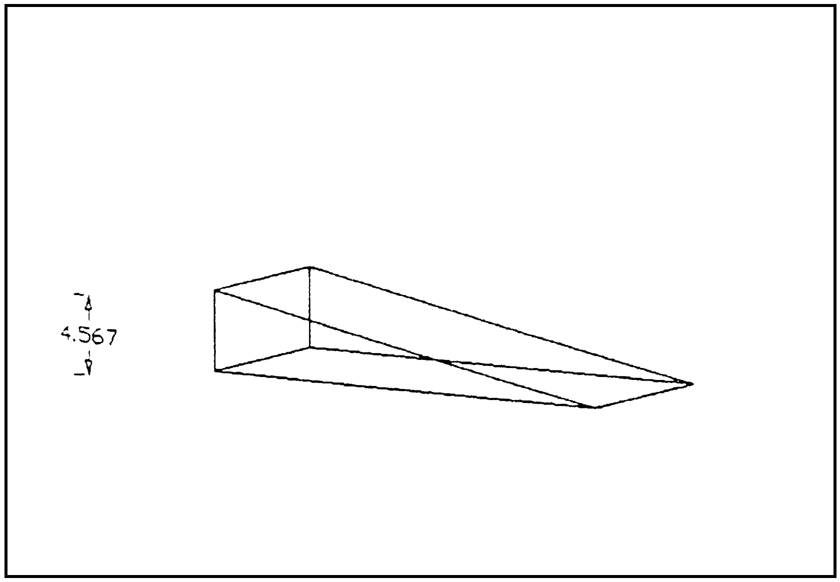


Fig. 4-14 PRE-RECTIFIED WEDGE DRAWING WITH Z-TICAL DIMENSION

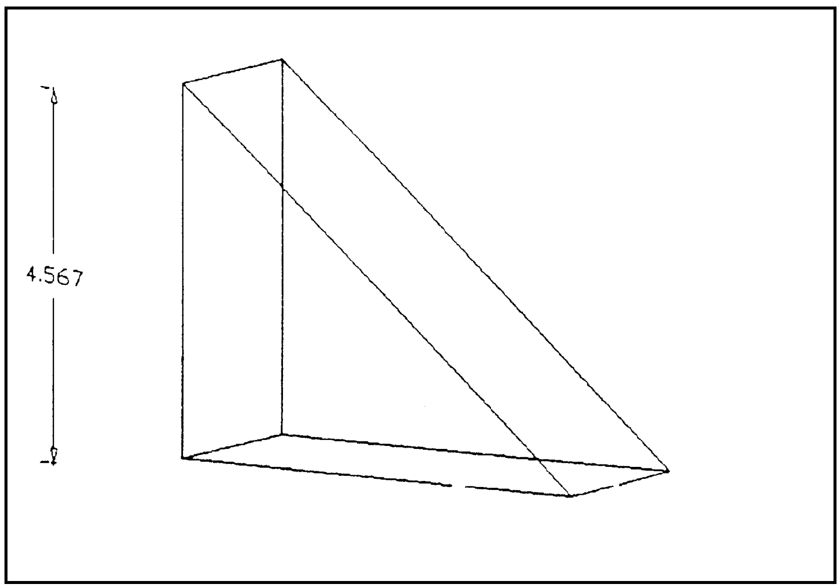


Fig. 4-15 AFTER RECTIFICATION

The dimension's legs can be thought of as a set of horizontal infinite planes extending in all directions from 0,0 axis and at the levels selected for the low and high values of the Z dimensions.

All entities supported by the z-tical dimensions will have their Z coordinates affected by changes in the controlling z-tical dimensions. The thickness of entities can also be changed this way.

The same basic rules apply for z-tical dimensions as do for horizontal or vertical dimensions.

1. **You can choose from six different types.**
   Fixed, Proportional, Stylized, Fixed Hidden, Proportional Hidden, or Stylized Hidden. (Note: in Vdraft, Stylized Hidden may not be selected.) Differences between dimension types are described elsewhere in this manual. The type of dimension selected controls the type of behavior of entities within its range of control.

2. **You can have variable text** for the dimension values which will be generated by your SpecSheet program or other method of creating SPC files containing data for OfficeCAD.

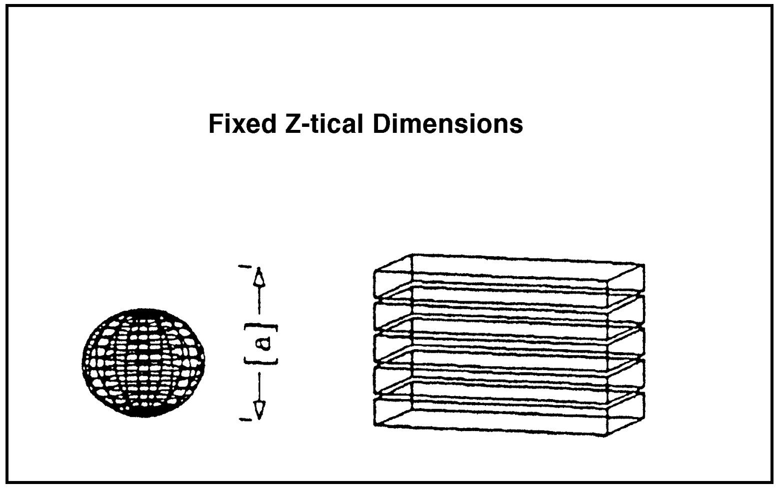


Fig. 4-16 FIXED Z-TICAL DIMENSIONS BEFORE

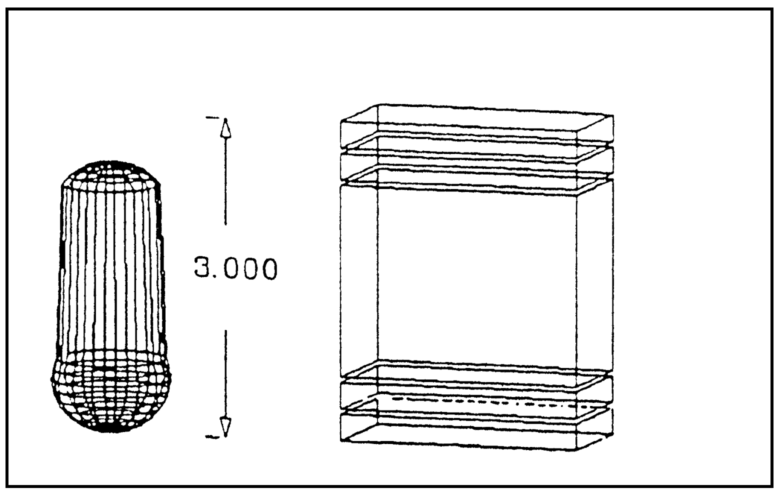


Fig. 4-17 FIXED Z-TICAL DIMENSIONS AFTER

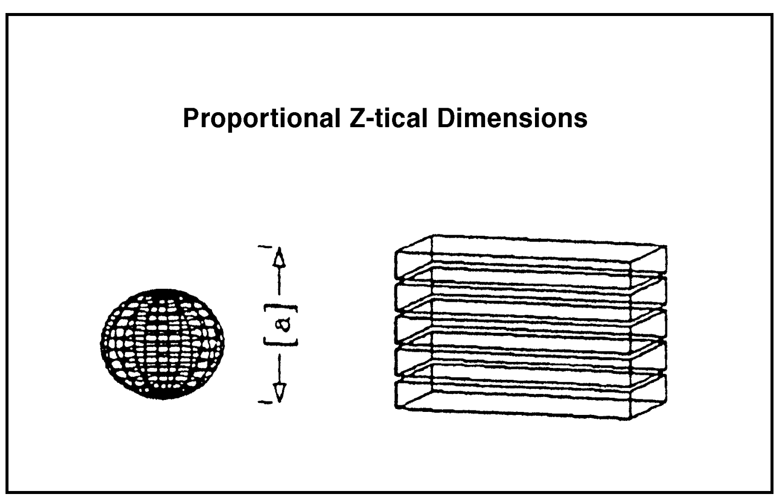


Fig. 4-18 PROPORTIONAL Z-TICAL DIMENSIONS BEFORE

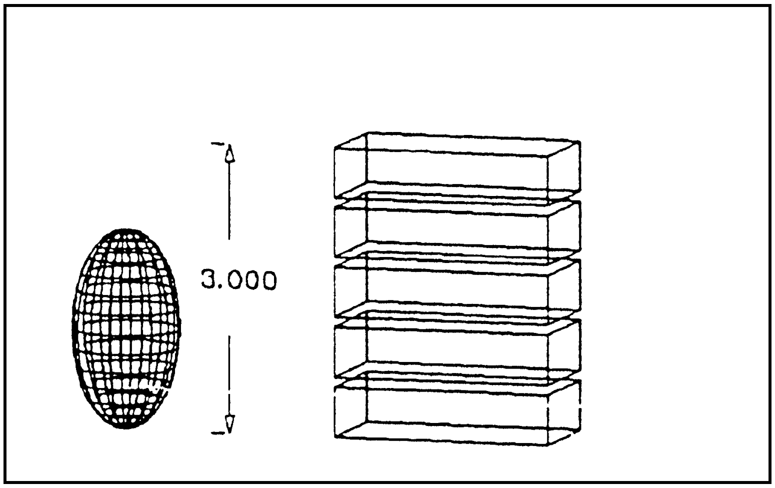


Fig. 4-19 PROPORTIONAL Z-TICAL DIMENSIONS AFTER

3. **You can "chain" dimensions** together and nest them. Dimension chaining in Z direction:

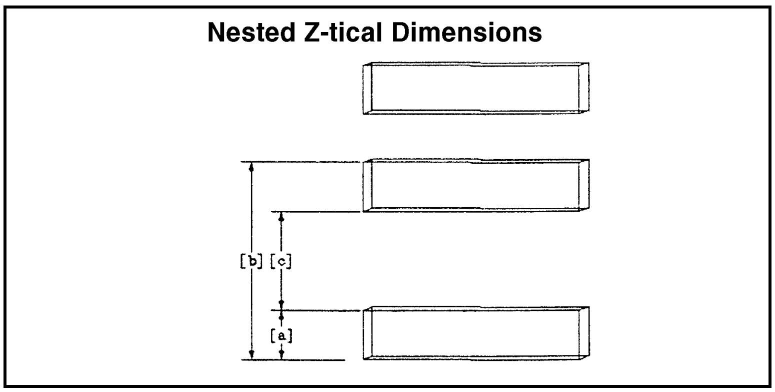


Fig. 4-20 BEFORE

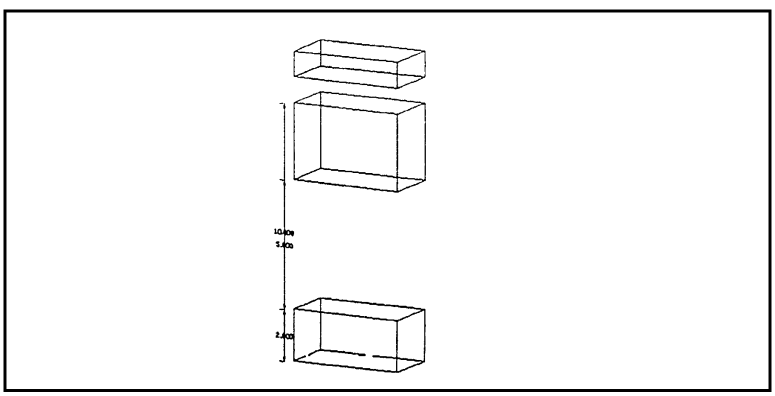


Fig. 4-21 AFTER RECTIFICATION

4. **Entities that are affected by Z dimensions include:**
(Note: These entities may be moved to different Z coordinates but will not turn into 3D objects simply because of a Z dimension acting upon it.)

- 2D entities with thickness greater than zero.
- 3D lines and polylines.
- Polymeshes are supported on world UCS but not polymesh blocks.

Z-tical dimensions will affect the position of blocks, but not the size of blocks.

As with horizontal and vertical dimensions, OfficeCAD will relocate entities according to predictable rules based on relative distance to the nearest controlling dimension, the entities control point(s) and the type of dimension controlling the entity.

The following is a simple 3D rectification example. This example will give you a feel for how OfficeCAD z-tical dimensions function.

Use AutoCAD to draw a polyline 1 unit square, with polyline thickness set to 1.

From OfficeCAD, select Dimensions, and choose "Fixed".

Select "Z-tical" from the dialog box.

Pick the dimension endpoints.

Enter **4.125** for the dimension text. Now, look at how the dimension appears. Use the VPOINT command (in Vdraft, use the "viewpoint" command) and enter **3,-3,1**.

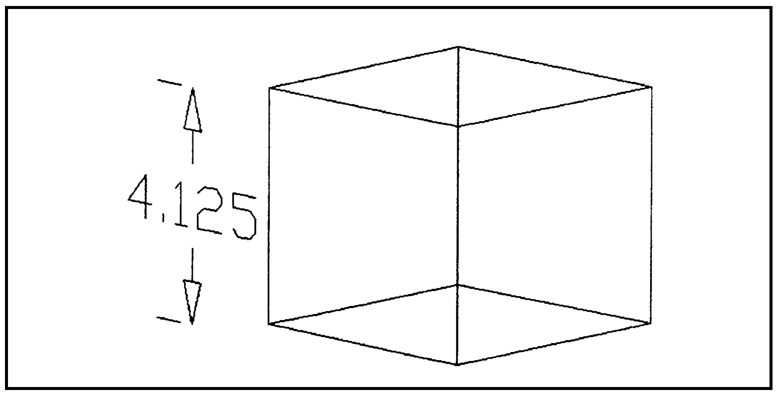


Fig. 4-22 CUBE WITH Z-TICAL DIMENSION TEXT BEFORE RECTIFICATION

As you can see, the dimension is located along the imaginary Z axis running through 0,0.

Now, select OfficeCAD, Rectify Drawing.

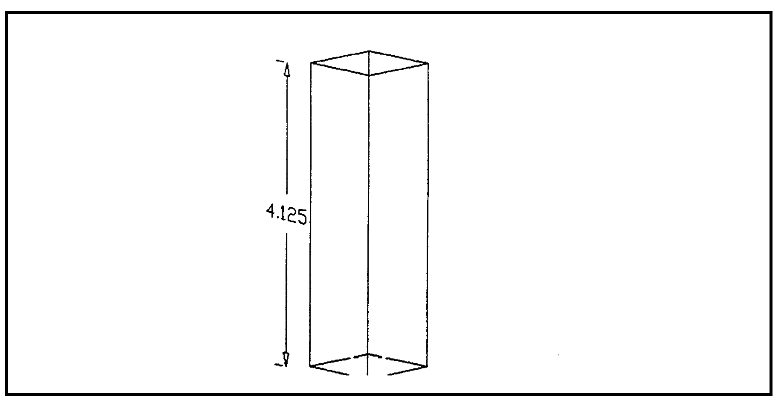


Fig. 4-23 Z-TICAL DIMENSION AFTER RECTIFICATION

Your cube with a thickness of one will now appear with a new thickness of 4.125 units.

Control Circles

OfficeCAD allows you to control an assortment of entities via the control circle method.

This method is valuable for drawings that need to have objects and entities moved in a radial manner. For example, splined shafts, gears, hubs with holes, etc.

The way OfficeCAD controls entities with control circles is straightforward. The user creates a control circle. On that circle (i.e., snapped to the circle) any entity control points which lie exactly on the circle will relocate radially when the circle's radius is changed via a radial or diameter dimension. (Entity control points are shown in Table 3-1 in **Chapter 3**.)

From OfficeCAD, select Dimensions, Ctrl Circle. Draw a circle. Next, from OfficeCAD, select Dimensions, Radial, and pick the control circle. You will be prompted to specify the dimension line location and the text for the dimension. Finally, you will be prompted to select any additional entities (arcs or circles) for the radial dimension to control.

The control circle defines a selection set of entities which will be moved radially either in or out depending on how the radial dimension moves the circle. The circle may be selected to be hidden or normal (visible).

In this way, one radial dimension can control the movement of a large number of entities.

3D Hub Example

This tutorial will take you through a simple example using a 3D hub to show how OfficeCAD can control via construction lines and control circles.

First, begin a new drawing in AutoCAD. We will draw a 3D hub using circles with thickness.

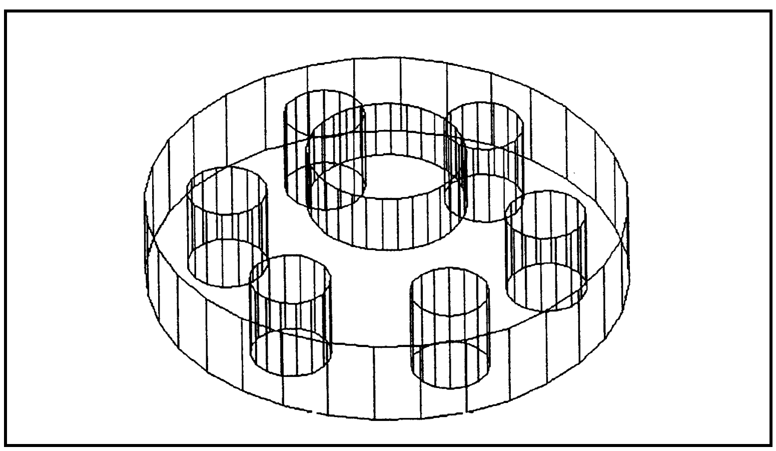


Fig. 4-24

Next, add control circles (no thickness) by selecting OfficeCAD, Dimensions, Ctrl Circle.

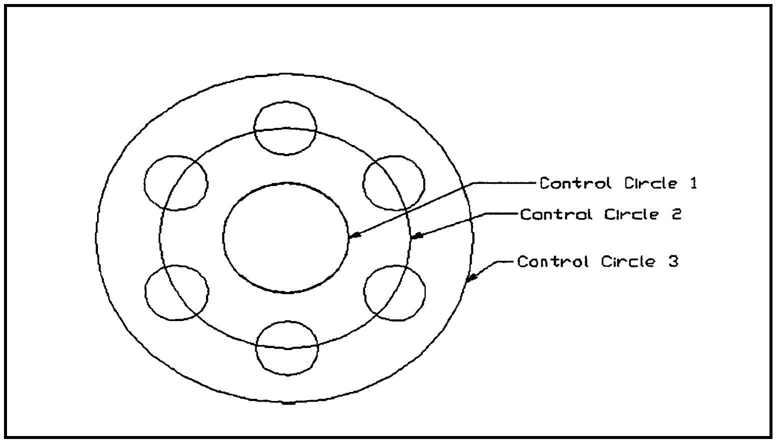


Fig. 4-25 DRAW CONTROL CIRCLES USING CIRCLE COMMAND

Put radial dimensions on the control circles as shown. Add a z-tical dimension.

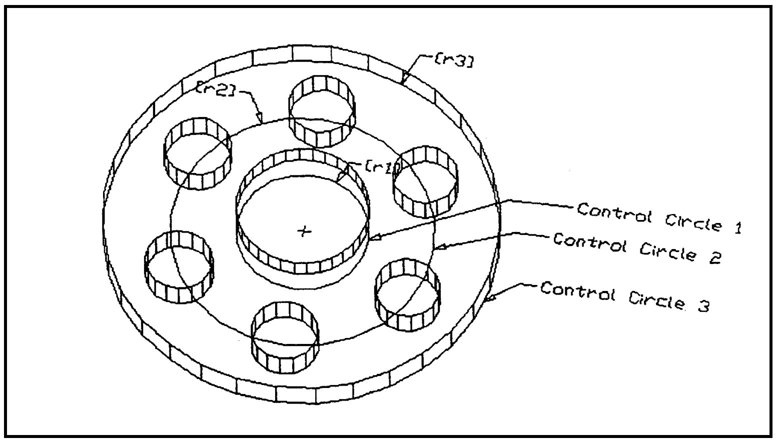


Fig. 4-26 RADIALLY DIMENSION THE CONTROL CIRCLES

Save the master drawing, then use the OfficeCAD Autospec routine to create a design in Excel (as described in **Chapter 2**). Run the design, supplying new values for [R1], [R2], [R3] and the z-tical dimension. You may run the design several times, trying different values.

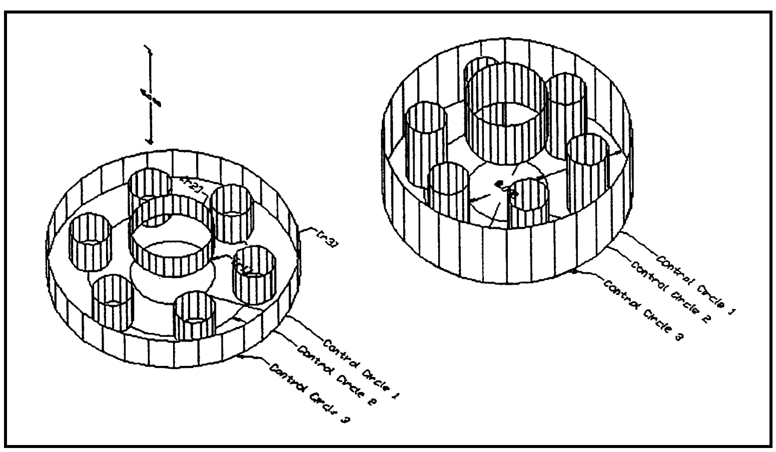


Fig. 4-27 AN EXAMPLE OF A 3D HUB

This example uses:
- Control circles for radial movement of entities.
- Radially dimensioned holes location using control circles.
- 3D z-tical proportional dimension on hub to control thickness of hub.

Control Lines

OfficeCAD has control lines. Entities which have their control points attached to these construction lines will remain "stuck" to these lines.

The benefit of control lines is that you can "nail down" a set of entities which may be within the influence of a set of nested dimensions of perhaps mixed DIMFIX or DIMPRO dimensions. The resulting movement can now be predicted by simply controlling one line or set of lines which control the desired entities.

To use construction lines from OfficeCAD, select Dimensions, then select "Hor Ctrl Line" or "Ver Ctrl Line".

Pick 2 points.

Control Line Example

The following is an example of how control lines can solve some design problems.

A simple master drawing appears in Figure 4-28.

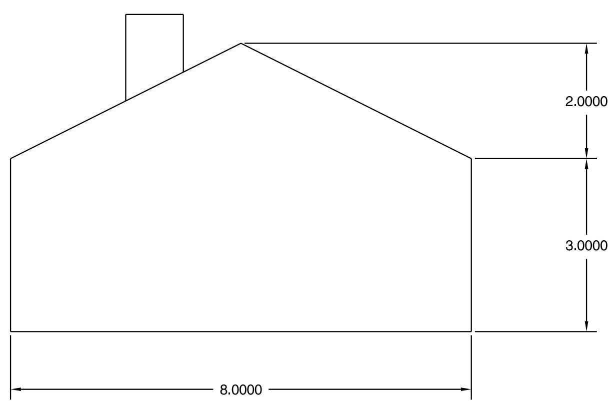


Fig. 4-28 HOUSE MASTER DRAWING

When the roof line is raised, the chimney becomes detached on one side.

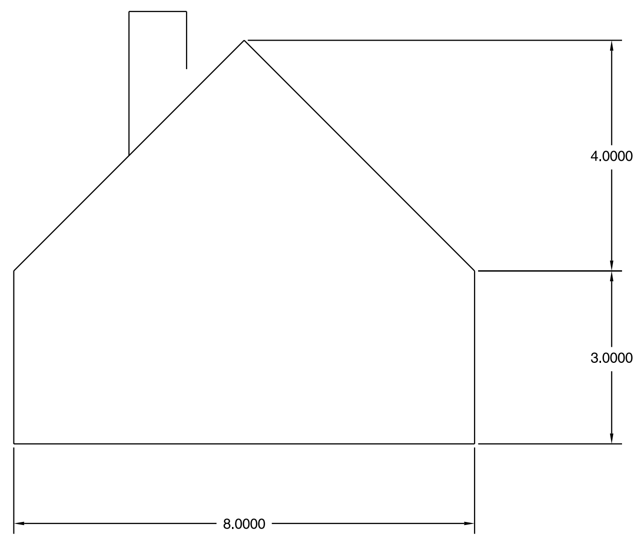


Fig. 4-29 HOUSE DRAWING WITH DETACHED CHIMNEY

To prevent this separation, a Vertical Control Line is added that matches the left side roof line.

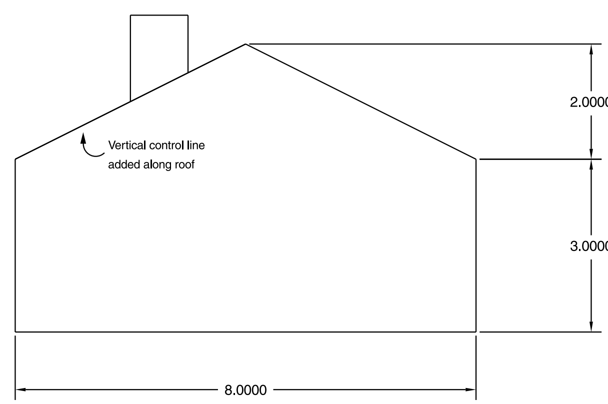


Fig. 4-30 HOUSE DRAWING WITH VERTICAL CONTROL LINE

Now when the lines that make up the chimney are relocated, the endpoints that originally fell on the control line will be adjusted (in this case, vertically) due to the Vertical Control Line, to again fall on the control line.

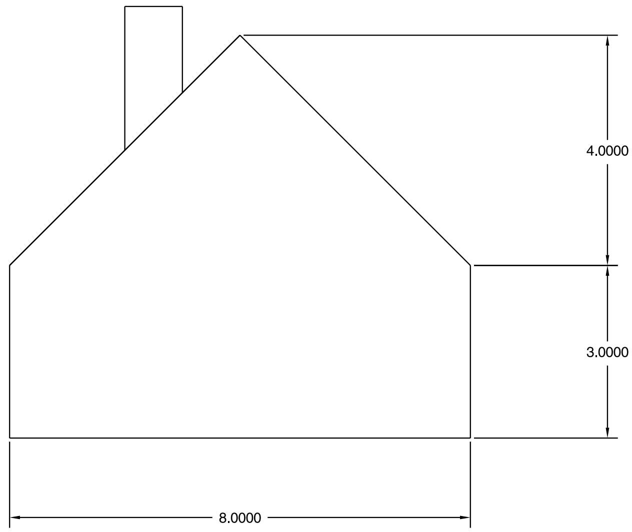


Fig. 4-31 HOUSE DRAWING WITH ATTACHED CHIMNEY

Dimension Chaining

"Chaining" refers to the order in which dimensions relocate with respect to each other during rectification. If the drawing is constructed in such a way that chaining proceeds in an unpredictable pattern, the rectified drawing might shrink or expand in the wrong directions. A house might suddenly have overlapping windows, or an electrical board could end up with a break in the circuits.

In order to have dimensions rectify properly, you must either plan the order in which the dimensions are inserted into the drawing or use hidden dimensions to maintain the desired relationships.

OfficeCAD chains dimensions by first grouping dimensions into sets. Sets of dimensions are either a single dimension or a series of dimensions in which two or more dimensions "share" a leg. Dimensions that "share" legs can be defined as follows:

Horizontal dimensions share a leg when the dimension legs of two or more dimensions are at the same X coordinate. Vertical dimensions share a leg when the dimension legs of two or more dimensions are at the same Y coordinate. Horizontal and vertical dimensions that are symmetric also chain on their centerpoints.

Figure 4-32 illustrates two different outcomes of chaining. For the first example, ignore the base point and assume that no base point was specified before rectification. Note that when no base point is specified (as in this case), chaining will start at coordinates 0,0. As each dimension is chosen to join the chain, it is rescaled to its new size. The next dimension chosen will be placed with respect to the previous dimension.

Dimension chaining begins at point (0,0) and moves horizontally to the right. Thus, the sequence of the chain is [H1], [H2], [H3], [H4], [H5]. When there are no further dimensions in this direction, chaining then returns to 0,0 and continues to the left. In this case, there are no dimensions to the left of this point and rectification is complete.

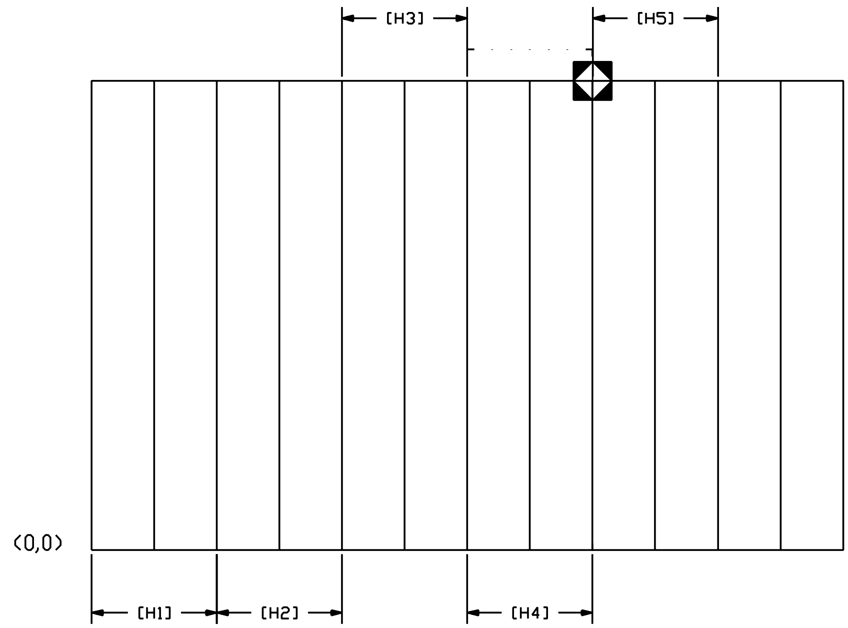


Fig. 4-32 CHAINING OF DIMENSIONS

Now assume that a base point has been specified as shown in the above drawing. The procedure is the same as in the previous example with two important differences: chaining now begins at the base point rather than at 0,0, and, secondly, there are dimensions on both sides of the base point. This means that [H5] will rectify first, as it is to the right of the base point, and, since there are no more dimensions to the right, chaining then continues left from the base point. The rest of the sequence is [H4], [H3], [H2], [H1] and it ends when no more dimensions are found.

Another point to note in this example, is that the extension leg on [H3] is at the same X coordinate as [H2] and [H4]. This shows that extension legs do not need to overlap.

Since depth-of-influence has no effect on chaining, [H4] is treated as part of the set.

The order in which dimensions will chain is fairly easy to see in Figure 4-32 since all the dimensions are horizontal. Most drawings, however, include more than one type of dimension. In a case where a drawing includes both horizontal and vertical dimensions, the horizontal dimensions will be chained first. When no more horizontal dimensions can be found, OfficeCAD will look for a vertical dimension with a leg at the base point, or, if there is no base point, at coordinates 0,0. Dimensioning continues exactly as it did with horizontal dimensions, except in the vertical direction, chaining upwards until no more dimensions are found and then downwards in the same way.

NOTE: If there are no dimensions starting at the base point (or 0,0), then a dimension is arbitrarily picked from which the first set of dimensions will chain. After the first set of dimensions has chained, the next dimension will be arbitrarily picked to start a new chain.

Ideally, distances between dimensions will be maintained; however, the above characteristics might interfere with this.

The following example illustrates how distance between dimensions might be lost.

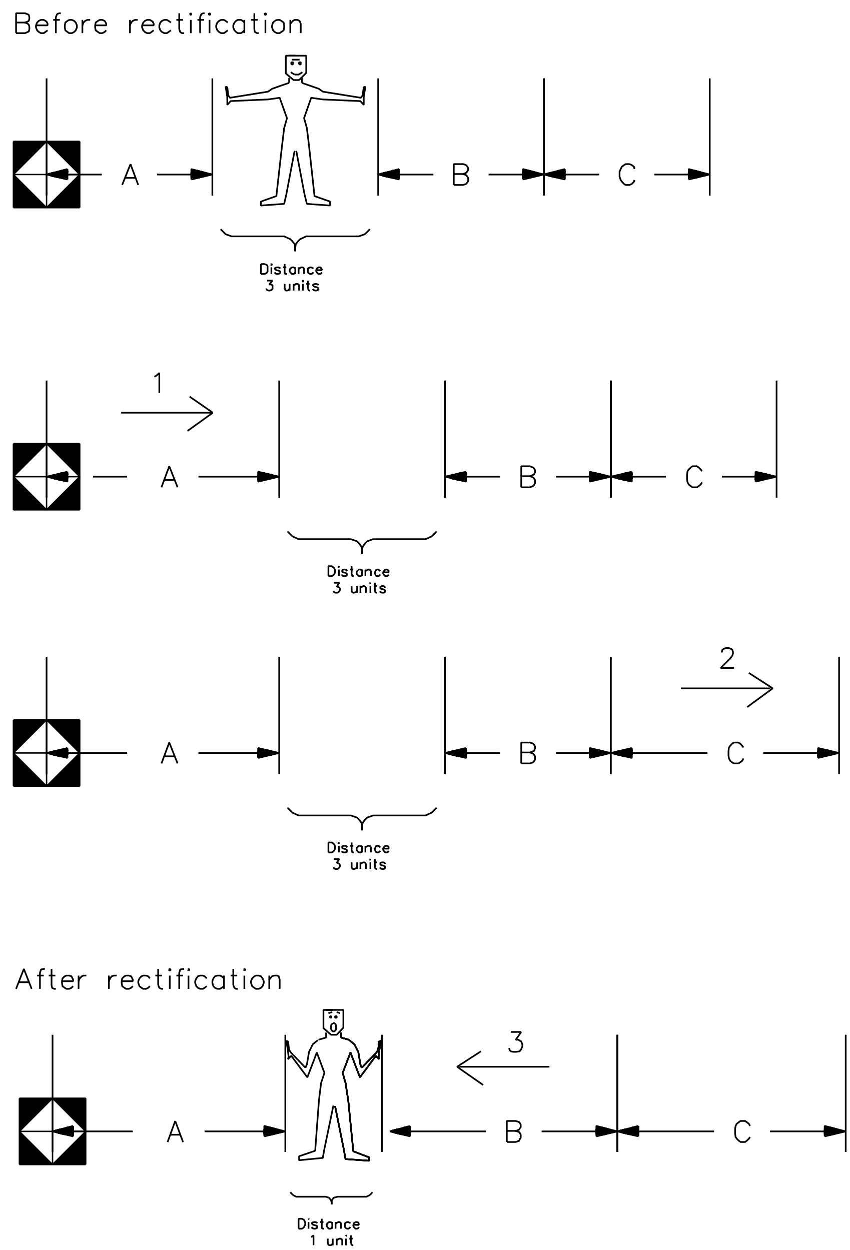


Fig. 4-33 HOW RELATIONSHIPS CAN BE LOST

Figure 4-33 illustrates two sets of dimension chains and what could happen if dimensions are inserted in the wrong order or hidden dimensions are not used. (Note that the letters A, B, C are used as labels for the purpose of explanation and are not intended as variable names.)

In the original drawing before rectification, there is a distance of 3 units between the two sets of dimension chains. Set 1 consists of A and set 2 consists of B and C.

Dimension A has a leg on the base point and hence is rectified first. The other dimensions maintain fixed distances from the first dimension.

Because B and C are not part of the set which includes A, one or the other may arbitrarily chain first. If B chains first, correct relationships are maintained. However, if C chains first as in this example, the correct distance between dimensions is lost. In this case, B rectifies with respect to C, in the direction of A and the distance between A and B is reduced—not the desired result. To correct this problem, add a hidden dimension between B and C (indicated by dotted lines) as follows:

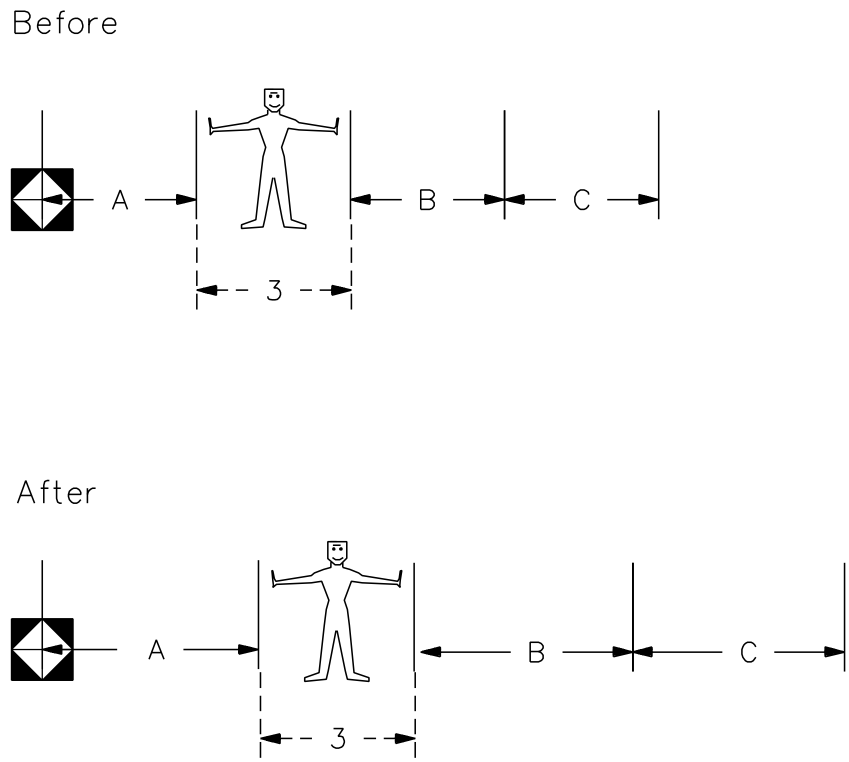


Fig. 4-34 USING HIDDEN DIMENSIONS

Here are some general guidelines for controlling the order in which dimensions are chained:

- Plan the order in which dimensions are inserted before creating the drawing. (Horizontal—left to right and vertical—bottom to top.)
- Insert the base point with chaining taken into consideration.
- Position the drawing in an acceptable relationship to the origin if no base point is specified.
- Take into consideration how “nesting” can be used to maintain relationships. (This is explained in the following section.)
- Use hidden dimensions (see “Hidden Dimensions” later in this chapter).
- Use symmetric dimensioning (see the “Symmetric Dimensioning” tutorial later in this chapter).

Nesting of Dimensions

It is recommended that you read and understand chaining of dimensions before you proceed with this section on nesting of dimensions.

“Nesting” refers to the way in which dimensions fit one inside of the other. Nesting affects the order in which the dimensions will chain and, consequently, the outcome of the drawing when it is rectified. Relocation of entities which are “nested” follows the “inside of” rule. Entities will always relocate according to the dimension they are most inside of. If they are not inside of any dimension, they relocate with respect to the nearest dimension. (See Side-to-Side Influences of dimensions in the following section.)

Figure 4-35 illustrates how the inner dimension relocates such that the leg closest to an outer dimension leg moves with that outer dimension leg. The other leg of the inner dimension relocates according to the size of the dimension. The diagram illustrates that this is true whether the dimensions are fixed or proportional.

If there is a “chain” of dimensions nested inside a larger dimension, those closest to the outer dimension relocate first, and the others then chain as described in the previous section on chaining.

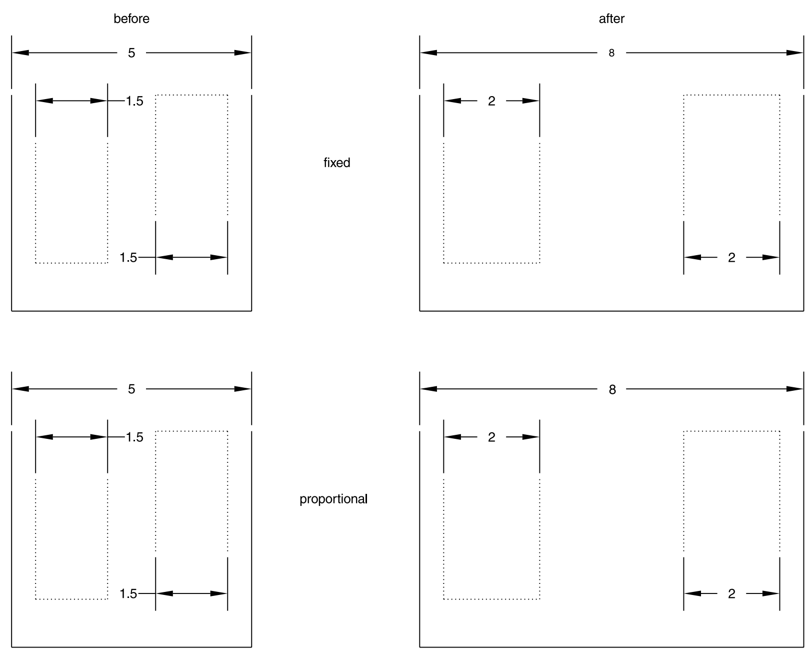


Fig. 4-35 RELOCATION OF DIMENSIONS

Figure 4-36 shows how relocation of entities may not be intuitively obvious. But if you understand nesting and chaining, then you will see that the outcome is correct. The left leg of the inner dimension was closest to the outer dimension, and maintained a fixed distance from it.

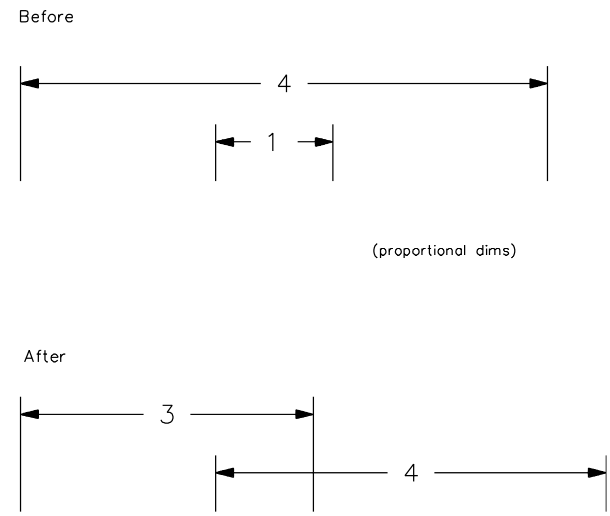


Fig. 4-36 RELOCATION OF ENTITIES

Side-to-Side Influence of Dimensions

Entities relocate according to the dimension they are inside of. If an entity is not inside of any dimension, it will maintain a fixed distance from the nearest dimension.

Figure 4-37 below illustrates entities along the moving side of a horizontal dimension with a depth of influence. After rectification, the circle along the moving side of the dimension has maintained a fixed distance from the dimension leg. Note that the circles on the moving side of the dimension that are above the depth of influence and below the dimension line location are not moved. Also note that the circles on the fixed side of the dimension and the lines above and below the dimension are not affected by any changes.

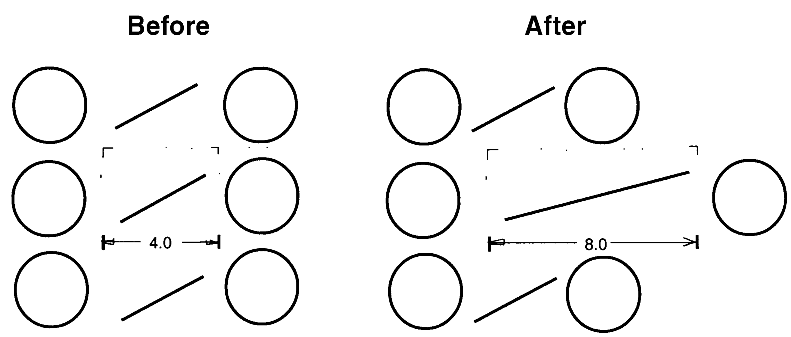


Fig. 4-37 MOVING SIDE OF A DIMENSION

Figure 4-38 illustrates a vertical dimension with a fixed depth-of-influence. Notice circles remain a fixed distance above the depth of influence reacting as they did with horizontal dimensions only in the vertical direction.

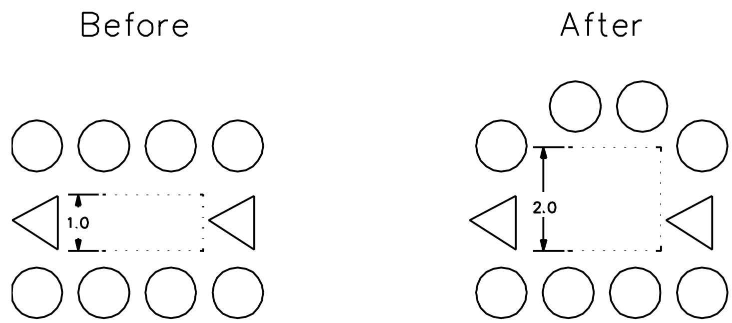


Fig. 4-38 FIXED DEPTH-OF-INFLUENCE

Dimensions with an infinite depth-of-influence are treated differently than dimensions with a fixed depth-of-influence. With infinite depth, the influence extends infinitely in both directions. Figure 4-39 illustrates a vertical dimension with an infinite depth-of-influence. All objects above the dimension leg are moved along with the dimension.

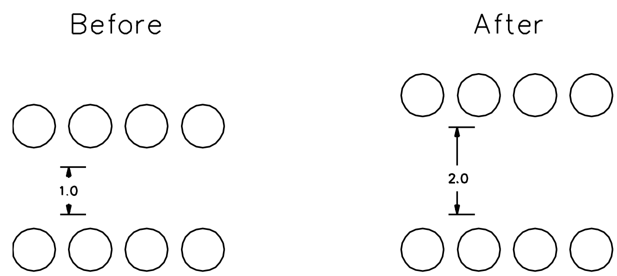


Fig. 4-39 INFINITE DEPTH-OF-INFLUENCE

Figures 4-40 and 4-41 illustrate a dimension with an infinite depth-of-influence and what happens when its dimension text is set to zero. Not only do the entities inside of the dimension disappear, all entities to the right of the dimension have moved to where the dimensioned entities were prior to rectification. Voila, the four-door car becomes a two-door!

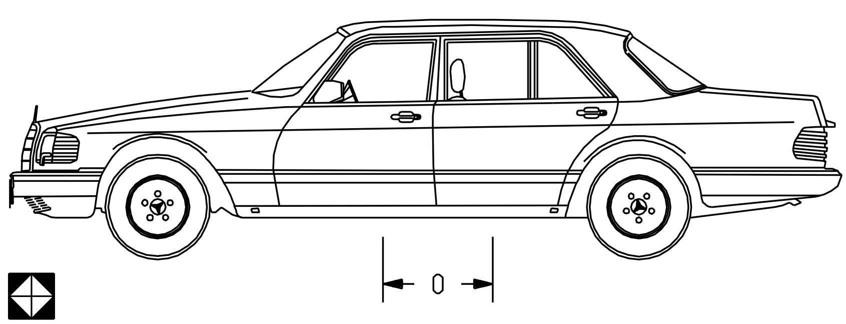


Fig. 4-40 DIMENSION TEXT SET TO ZERO

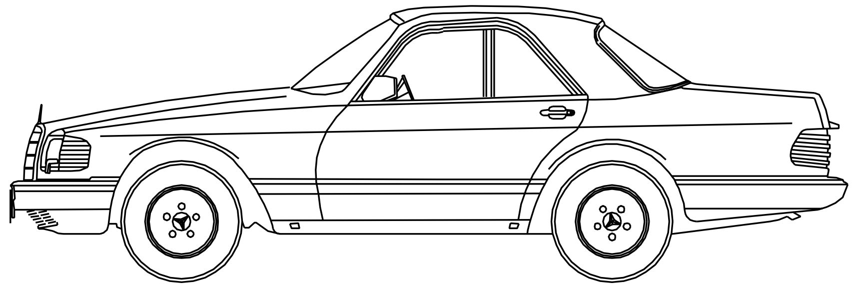


Fig. 4-41 AFTER RECTIFICATION

Symmetric Dimensioning

Centerline Dimensioning

One of the valuable techniques for drawing and dimensioning with OfficeCAD is based on using balanced dimensions about a centerline of symmetry on a drawing. This technique is referred to here as “centerline dimensioning”.

For symmetric drawings, the OfficeCAD program works much better if you can determine an axis of symmetry and put your base point on that axis. Dimensions will then “shrink towards” or “grow away” in relation to this base point.

Centerline Dimensioning Example

The following example of a simplified gear shaft side view will illustrate centerline dimensioning of a symmetrical object. The dimensions radiate out from the center axis of the shaft and are balanced on both sides. The radius of the first gear section has a vertical dimension from the top of the gear to the bottom of the gear.

To do this tutorial, enter AutoCAD, turn snap mode on and use a polyline to draw the top half of the gear shaft figure. Drawing to exact scale is not necessary, since OfficeCAD will create the accurate final drawings for you. Now mirror through the x-axis to create a drawing similar to the one in the figure below. It is strongly advised that you use snap mode when making your master drawing. This assures that connections are exactly joined, and that dimensions will be snapped to endpoints and intersections.

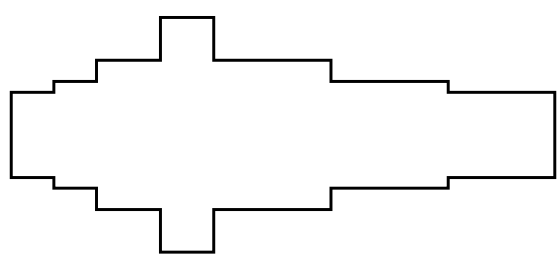


Fig. 4-42 STEPPED GEAR SHAFT – SIDE VIEW

Put a base point on the centerline of the symmetrical axis of the drawing. Drawing entities will rectify in relation to this base line axis. The most logical place to put it is a place where:

- there is symmetry
- you want entities to grow away from or shrink to

- you may want to use a common insertion point for several modules
- you want to have the 0,0 coordinate located on your output drawing

In this case, we will choose a point at the left end and on the center line. From OfficeCAD, select Dimensions, Base Point and locate the base point symbol on the drawing.

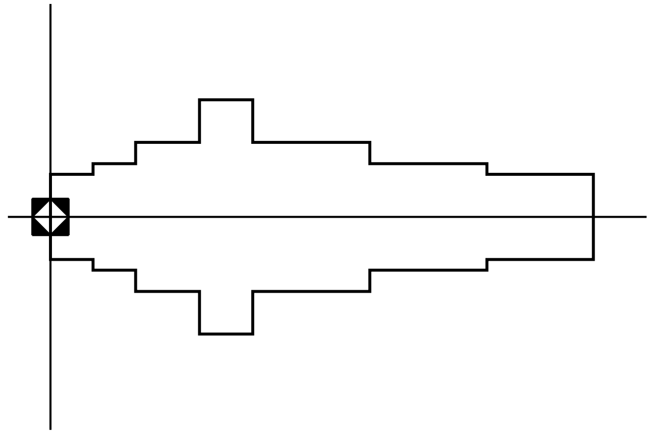


Fig. 4-43 ADDING THE BASE POINT

Next we will put symmetric vertical dimensions on the drawing for each diameter of the separate gears. To do this, from OfficeCAD, select Dimensions. Check the “linear dims are symmetric” box before creating the vertical dimensions, starting with the inner diameter and working out.

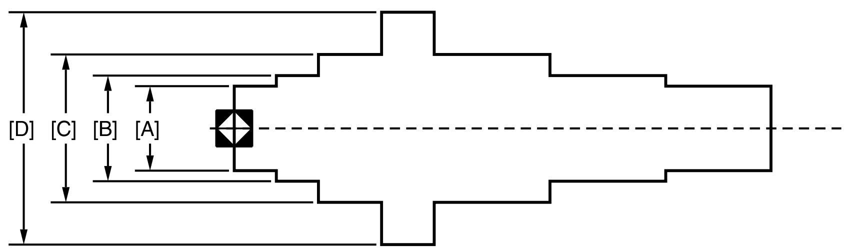


Fig. 4-44 SYMMETRIC DIMENSIONS

The dimensions will grow away from the centerline of the shaft since the base point is on this line of symmetry. The dimensions will control the drawing and show the shaft diameter.

The base point symbol will be deleted when you run your design. All you need to do now is make a SpecSheet which prompts for the values for the shaft diameters.

Hidden Dimensions

When you create an active dimension, you have the option of making it visible or hidden. Hidden dimensions have the same effects as visible dimensions, but do not appear in the plotted drawing. To ensure the proper outcome upon rectification, you can use hidden horizontal, vertical, z-tical, radial, diameter or angular dimensions; also, you can use hidden control circles and control lines. Hidden dimensions are shown as dotted lines in the drawing’s working format, to distinguish them from ordinary dimensions. Hidden dimensions can be turned off in your output drawing by selecting Edit Dims, Toggle Hidden. (Display formats are discussed in the next section.)

The triangle in the next diagram illustrates a situation that requires hidden dimensions. The aligned dimensions [A] and [B] are passive; they do not provide rectifying controls. Hidden horizontal and vertical dimensions, [X] and [Y], are needed to locate the top of the triangle.

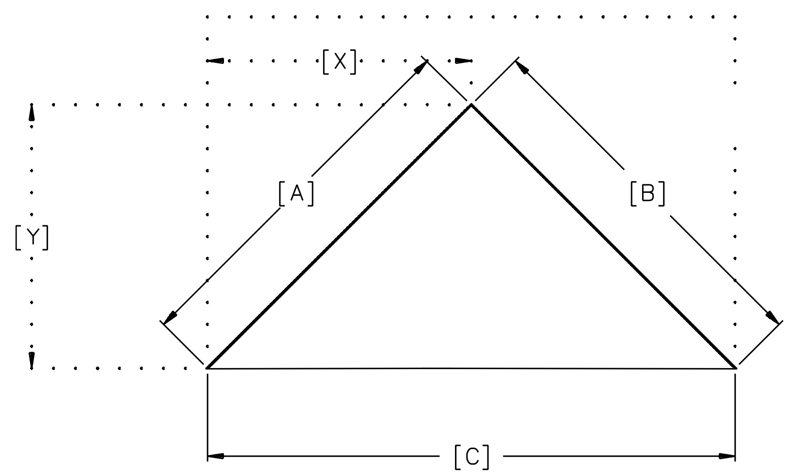


Fig. 4-45 A MASTER DRAWING THAT REQUIRES HIDDEN DIMENSIONS

Negative Dimensions

If you give a horizontal, vertical or z-tical dimension a negative value, the entities controlled by the dimension move in a direction opposite to that of a positive value. Often the entities cannot be re-rectified to their original position because they become superimposed upon other objects when they "flip over".

The figure below shows a drawing in which a dimension of +2.0 is rectified to a dimension of -2.0. Note that with fixed dimensions, the affected control points are pushed along at a fixed distance from the nearest extension line, whereas with proportional dimensions, they flip over, proportionately.

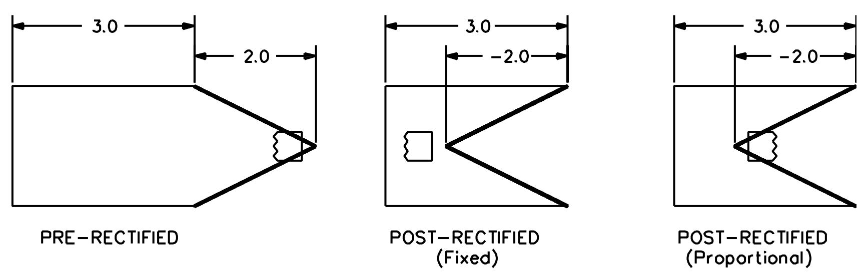


Fig. 4-46 NEGATIVE DIMENSIONING EXAMPLE

On occasion, you may wish to have the mirror image of an object rectified by OfficeCAD. In this case you can use the negative value for a dimension to cause the object dimensioned to "flip" to the other side about the lower value of the z-tical dimension.

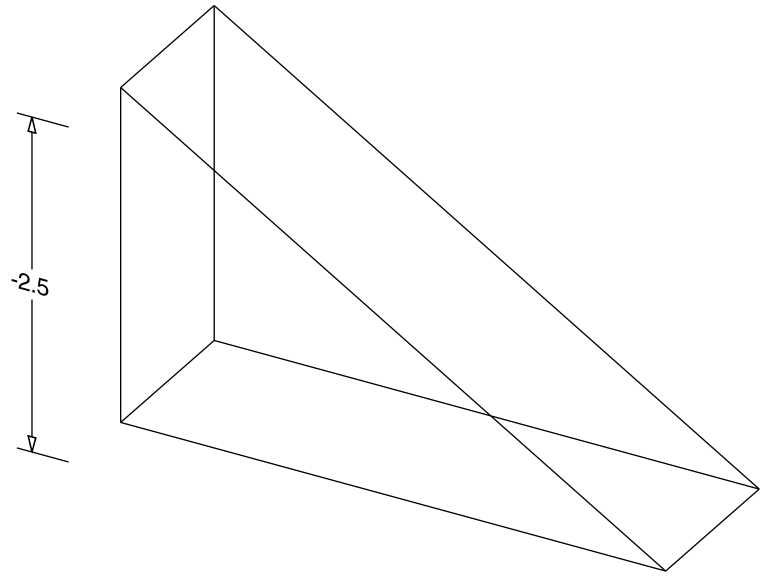


Fig. 4-47 BEFORE

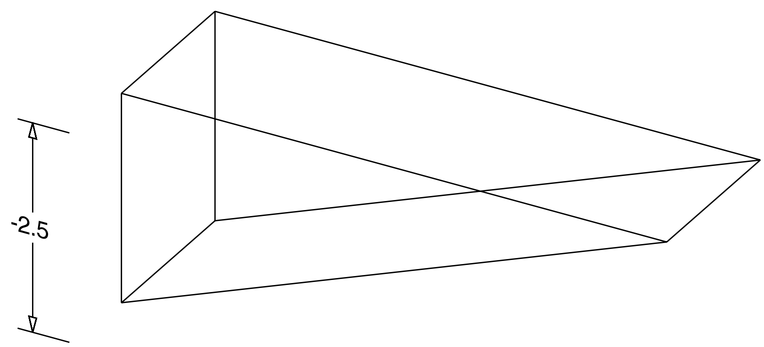


Fig. 4-48 AFTER

Other Elevations

Dimensions are not elevation-specific. The effect of a horizontal or vertical dimension extends above it and below it to control other elevations or z-coordinates. For example, one dimension at either the bottom or the top surface of a cube will rectify the entire cube. Z-tical dimensions have infinite depth in x and y direction.

Deleting Dimensions

Entity Elimination

Entity elimination enables OfficeCAD to produce output drawings of varying complexity from the same master drawing. When you give a horizontal, vertical or z-tical dimension a value of zero, all entities whose control points are completely within the dimension's region of influence are eliminated from the drawing. The dimension itself also disappears. Entities with at least one control point outside the region of influence maintain a fixed relationship to the extension line of the nearest remaining dimension. Circles and arcs are eliminated from a drawing when you give a radial or diameter dimension a zero. One exception is that if both endpoints of a line are on the leg of the dimension, the line remains.

NOTE: Once an entity has been eliminated, it cannot be retrieved. For this reason, it is very important that the output drawing be saved under a different name from the original, unless you want the entities permanently removed from the original as well.

The following two figures illustrate entity elimination. Each figure shows a drawing where a controlling dimension has been changed to zero, and how those entities look after rectification.

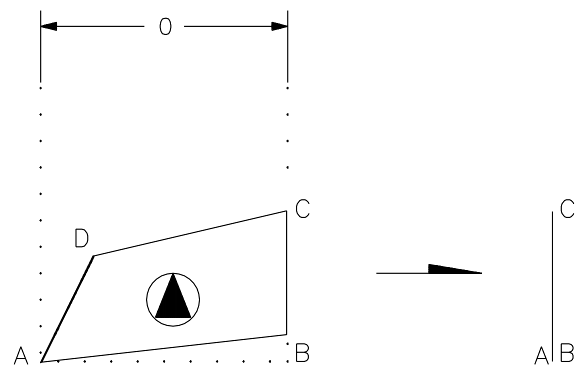


Fig. 4-49 ENTITY ELIMINATION I

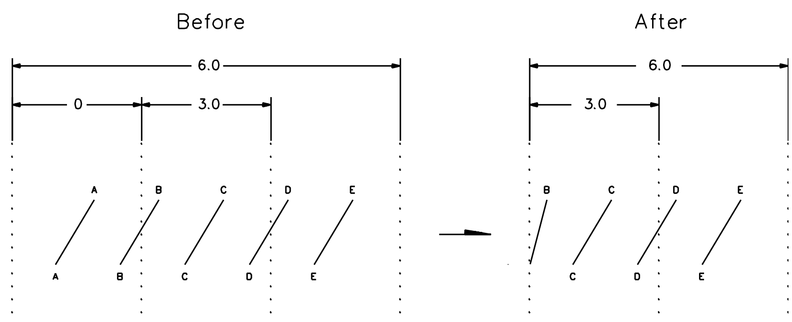


Fig. 4-50 ENTITY ELIMINATION II

Chapter 5 –
Arcs, Circles and Polylines

Radial and Diameter Dimensioning

Radial and diameter dimensions are intended for use on specific entities in the rectifiable drawing. Radial and diameter dimensions will actively control arcs, polyline arcs and circles.

When creating a radial or diameter dimension, you will first select the arc, polyarc or circle for the dimension to control. After the dimension is drawn, you may optionally choose additional arcs, polyarcs and circles to be controlled "invisibly" by the same dimension. When the dimension resizes, the first entity selected will resize to the new radius/diameter. For each additional entity selected (if any), if the additonal entity originally had the same radius/diameter as the first object, then it will resize to match the new radius/diameter of first object. If the additional entity did not have the same radius/ diameter, it will resize according to the formula below:

Given—the orignal radius of the first entity is OldRad1, and the new size of the first entity is NewRad1. For this additional entity, the original radius is OldRadn. To find the new size of the additional entity, NewRadn, depends on whether the radial/diameter dimension is fixed, proportional or stylized.

Fixed	NewRadn = OldRadn + (NewRad1 - OldRad1)
Proportional	NewRadn = OldRadn * (NewRad1/ OldRad1)
Stylized	The first entity and all additional entities selected do not resize

Fixed, Proportional and Stylized

Here we use a cylinder drawing as an example of selecting multiple entities for control by a radial dimension. The drawing shows the end view of a cylinder with a bore through it. When the dimension is drawn, the inner circle is selected as the first entity, and the outer circle is selected as an additional entity. Depending on whether the dimension layer is fixed or proportional, two different results occur when the dimension resizes.

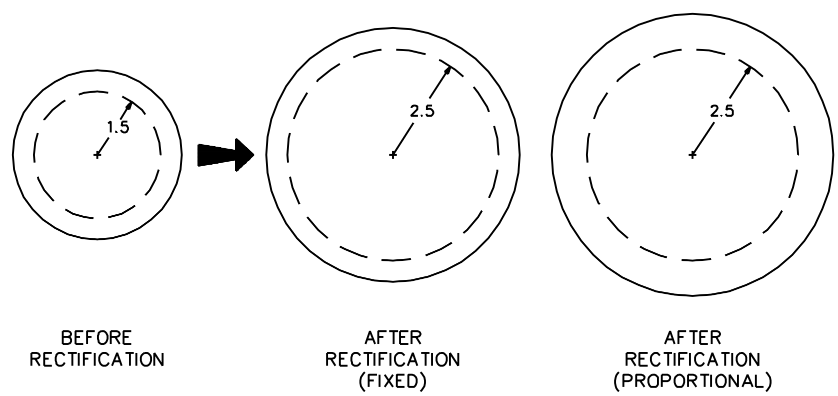


Fig. 5-1 RADIAL DIMENSION AFFECTING CIRCLES OF DIFFERENT SIZES

When the dimension layer is fixed, the distance between the inner and outer circle remains the same. When the layer is proportional the distance between the inner and outer circles changes in proportion to the change of size of the inner circle. (If the inner circle becomes twice as big, the cylinder wall becomes twice as thick.)

Poly-Arc Tutorial

The following tutorial demonstrates the procedure for constructing and dimensioning a polyline. It also illustrates how arcs rectify differently, depending upon whether their centers are controlling points as opposed to envelope points (see "How Polylines Work" later in this chapter).

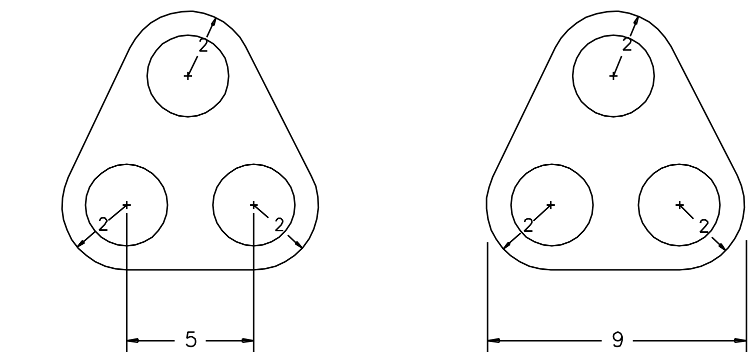


Fig. 5-2 POLY-ARCS (BEFORE RECTIFICATION)

Follow the instructions below to create the drawing shown above and then rectify it to produce the following result:

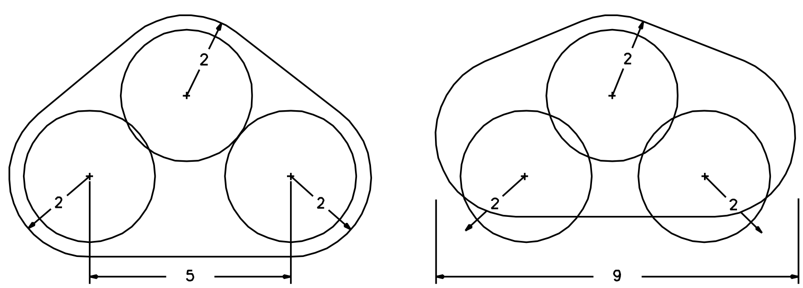


Fig. 5-3 POLY-ARCS (AFTER RECTIFICATION)

1. Construct two identical polyline triangles that are each made up of three straight segments and three arc segments. The segments must be tangent. (Hint: construct equilateral triangles and use the AutoCAD command Modify, Fillet to insert the arcs.)

2. Find the centers of the arcs. In one triangle, dimension the center horizontally. In the second, dimension the "envelope points", as shown.

3. Add three circles to each triangle, as illustrated in Figure 5-2.

4. For each rounded triangle, select the Radial dimension and include in the selection set all three polyline arcs and all three circles. (Only those arcs and circles in the selection set will change during rectification.)

5. Now rectify the drawing and note the results.

Constructing Radial and Diameter Dimensions

This section describes how to construct radial or diameter dimensions and how they affect arcs, circles and polylines.

In AutoCAD, from the OfficeCAD dialog box, select Dimensions, then radial or diameter. Next, pick the leader point. You will be asked to select additional arcs, circles and polyline arcs for the dimension to affect.

You can select any number of arcs or polyline arcs to share a radial or diameter dimension—even segments that are unrelated.

Angular Dimensioning

Angular dimensions have two sides: the fixed side and the moving side. The moving side will move with respect to the original position of the fixed side. (See the tutorial on angular dimensions for more information.)

The most important rule to remember when using more than one type of dimension on the same entity is: Don’t over-dimension. In the case shown below (“problem”), the polyline is being controlled by both linear and angular dimensions. This will cause ambiguous results and can be avoided by using radial and angular together (“solution 1”) or linear dimensions exclusively (“solution 2”).

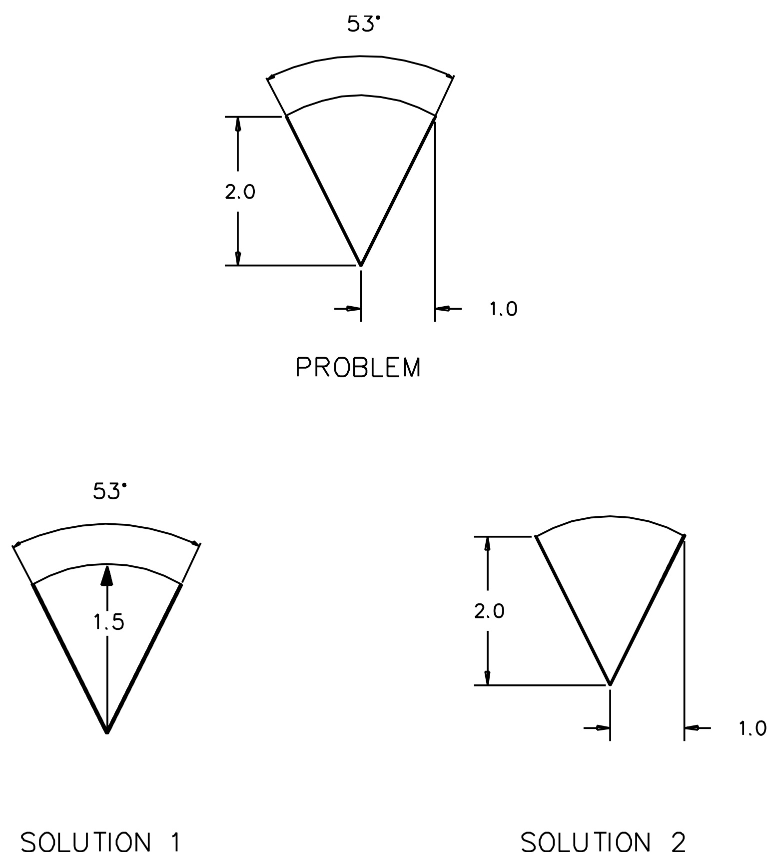


Fig. 5-4 AVOIDING OVER-DIMENSIONING

Dimensioning the drawing in stages allows testing the graphic output (using Rectify) and facilitates error-tracking.

Constructing Angular Dimensions

This section describes how to construct an angular dimension and how angular dimensions affect polylines.

To construct an angular dimension:

1. From OfficeCAD, select Dimensions.
2. Pick the dimension layer.
3. Select angular dimension.
4. Select the center point.
5. The next point picked will be the moving side.
6. Finish selecting the dimension points.

Note that only one side of the arc or angle moves upon rectification. Choose the side you want to be moving.

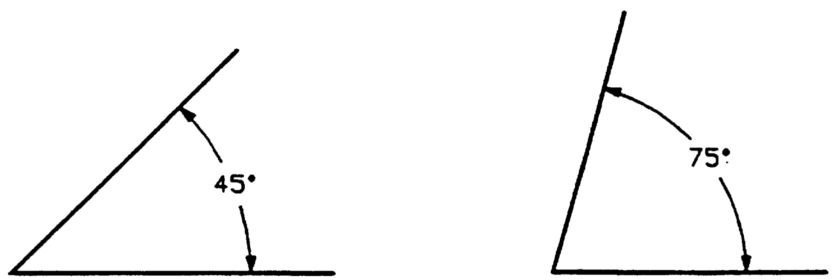


Fig. 5-5 POLYLINE CONTROLLED BY AN ANGULAR DIMENSION

How Angular Dimensions Affect Polylines

An angular dimension whose moving side aligns with a polyline's control point, governs the control point's position. The polyline is rearranged in conformity to its relocated control points. Both arc and straight segments are affected, as illustrated in these examples.

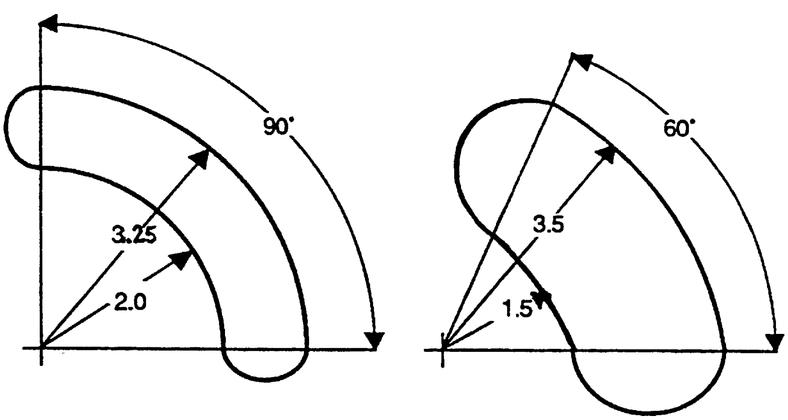


Fig. 5-6 BOTH ANGULAR AND RADIAL DIMENSIONS GOVERN THIS POLYLINE

An angular dimension overrides most other polyline considerations, such as tangency. This makes it easy to over-dimension a polyline, producing a bad result.

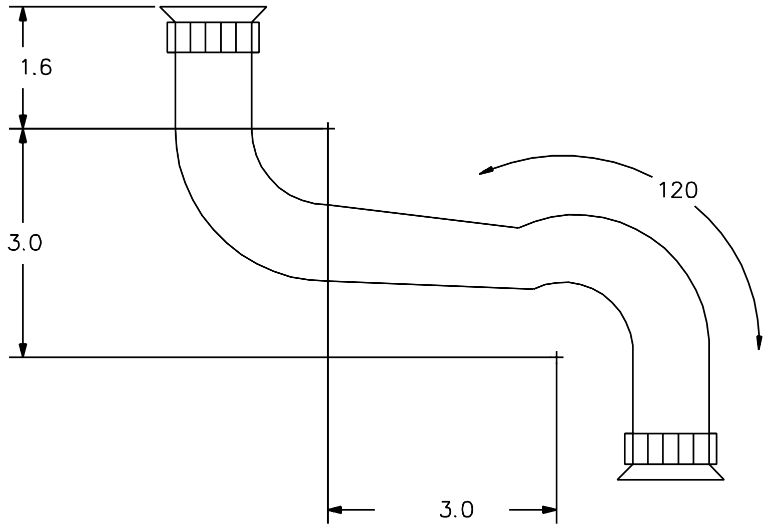


Fig. 5-7 OVER-DIMENSIONING: ANGLE OVERRIDE

Types of Polyline Points

Polylines

Polylines are made up of arc and/or straight segments. Three advantages of polylines are:

- Tangency relationships between contiguous straight and arc segments are maintained in most cases.
- Angular dimensions affect straight segments.
- Angular, radial and diameter dimensions affect arc segments.

How Polylines Work

In OfficeCAD, most of the drawing entities are relocated in a very simple fashion (see Table 3-1 in **Chapter 3**). Lines are relocated by their endpoints, circles are relocated by their center points.

The relocation of polyline points depends on how those points relate to the rest of the polyline. All the examples assume that the polylines are continuous. Points are classified into six types.

S	-	Sharp	**NT**	-	Non-tangent
AC	-	Arc center	**AJ**	-	Arc junction
T	-	Tangent	**E**	-	Envelope

A sharp point (S) is where two straight segments meet or where a polyline terminates with a straight segment.

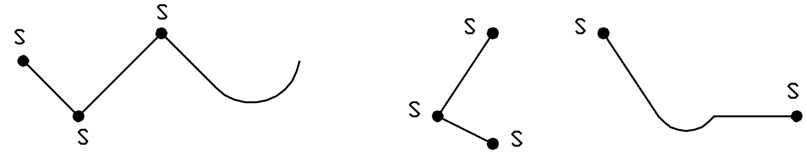


Fig. 5-8 SHARP POINTS

An arc center (AC) is the center point of any arc segment.

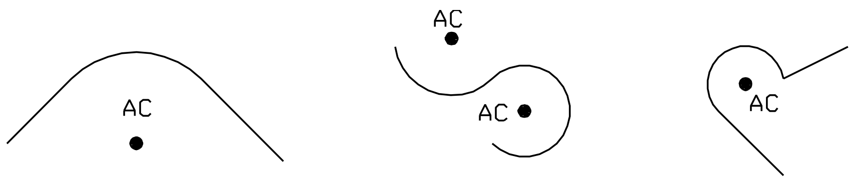


Fig. 5-9 ARC CENTERS

A straight segment can meet an arc segment in one of two ways: tangentially (T), or non-tangentially (NT).

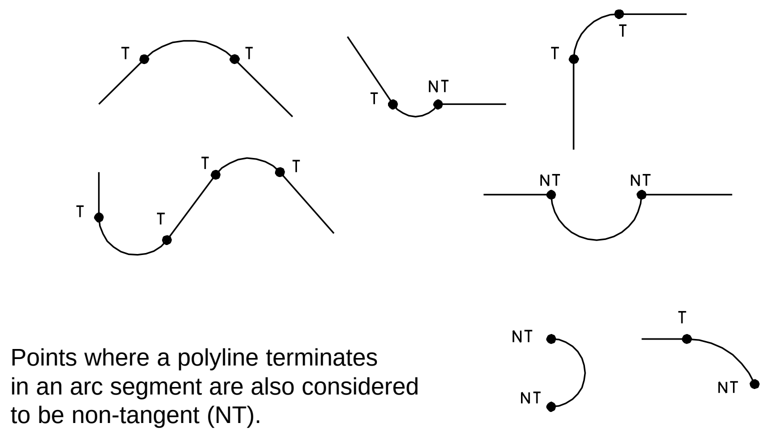


Fig. 5-10 STRAIGHT AND ARC TANGENTS

Arc junction points (AJ) are where two arc segments meet.

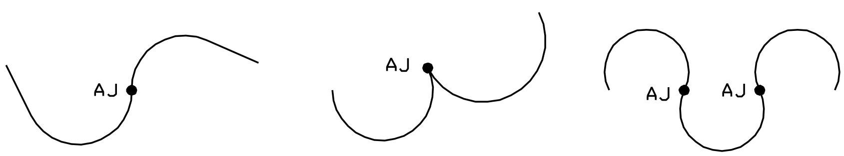
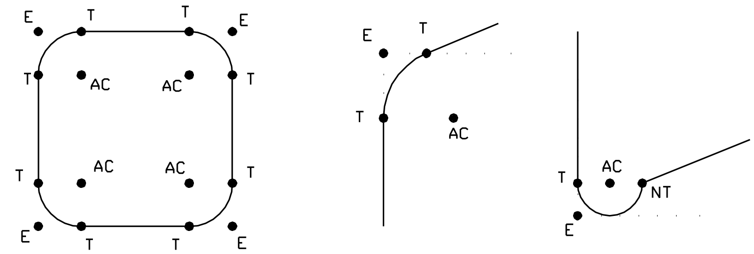


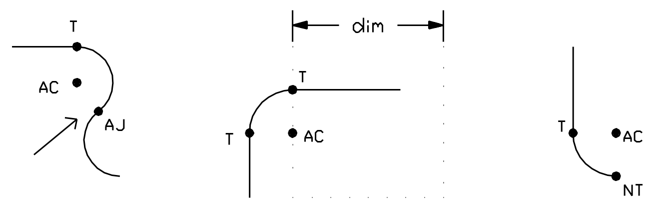
Fig. 5-11 ARC JUNCTION POINTS

Envelope points (E) are a special case where the control points are not on the polyline they are associated with. In order to have an envelope point several things must occur. The polyline must have an arc segment with straight segments joined to both endpoints, and at least one of these straight segments must be tangent to the arc and be horizontal or vertical. Also, there may *not* be a dimension passing through the center of the arc. The envelope point itself is at the 90° corner where a horizontal and vertical tangent would meet. Basically, a radius up (or down) and a radius over from the arc center.

These are envelope points:



These are not envelope points:



Not a linear side   Dim through arc center   Not a linear side

Fig. 5-12 ENVELOPE POINTS

Polyline points can be affected by horizontal, vertical, angular, diameter and radial dimensions, and control lines and control circles, as well as by their relationship to the points around them.

Sharp Points

Sharp points (S) are relocated by horizontal and vertical dimensions. The only exception to this is when the sharp point is the endpoint of a straight segment that is the moving side of an angular dimension.

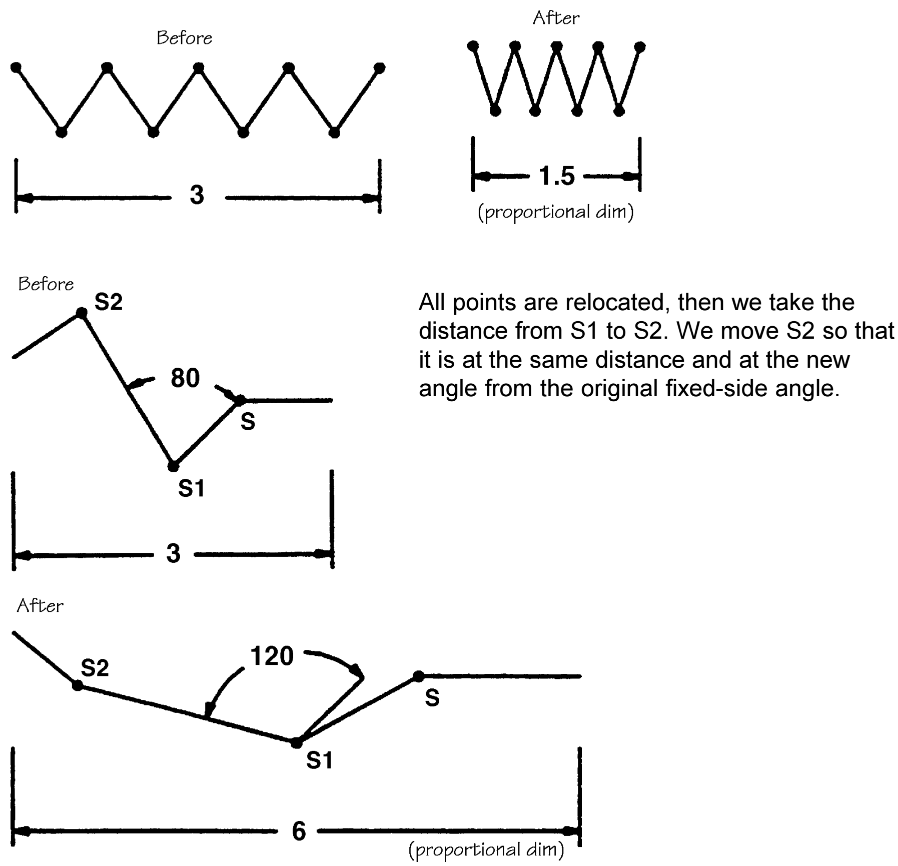


Fig. 5-13 SHARP POINT RELOCATION EXAMPLE

Arc Centers

Arc centers (AC) relocate by any of: horizontal and vertical dimensions; envelope action; or with relation to the endpoints of the arc.

Envelope Action

The envelope point is relocated with respect to horizontal and vertical dimensions, then the arc center is relocated to maintain its relationship to the envelope point, either at the original radius or new radius (if a radial or diameter dimension is present).

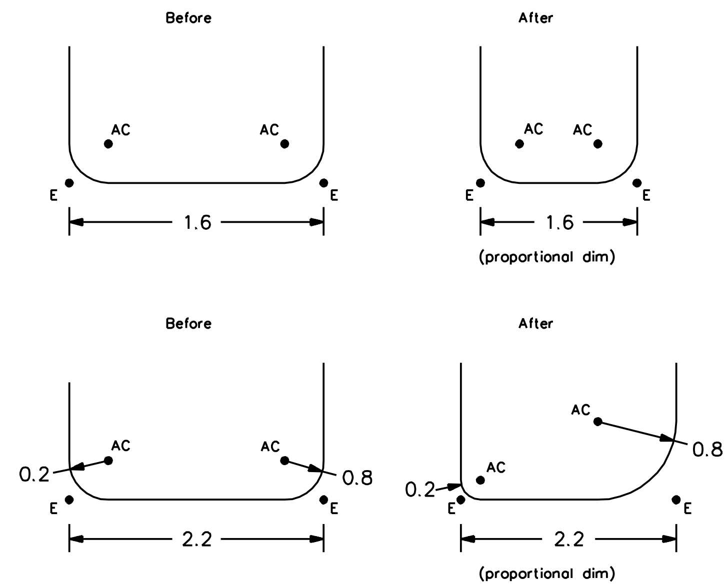


Fig. 5-14 ENVELOPE ACTION

For an arc to be relocated by its center, it must have a horizontal or vertical dimension snapped to its center, or a radial or angular dimension at its center point. (It also must not be related to an envelope point.) These arcs will have their centers relocated by horizontal and vertical dimensions:

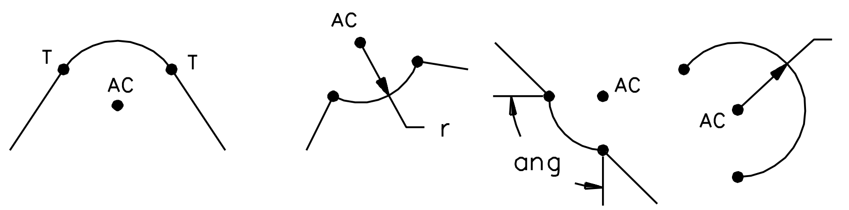


Fig. 5-15 ARC CENTERS

The remaining arc centers are relocated with respect to their endpoints. These are the arcs that are not related to envelope points and do not have an angular or radial dimension on them, nor a horizontal or vertical dimension snapped to the center, nor a tangent point on either end.

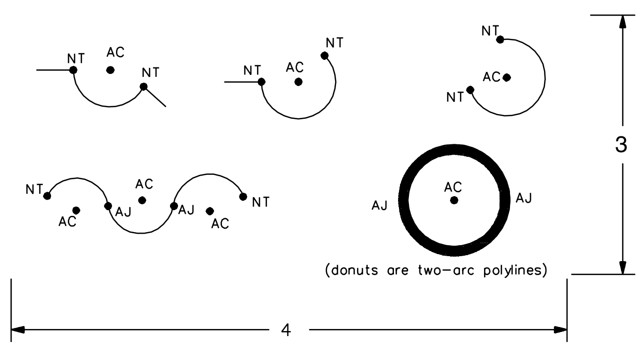


Fig. 5-16 ARC CENTER RELOCATION (Before)

First the arc endpoints are relocated, then the center is placed so that the same degree of arc is maintained (radius may change).

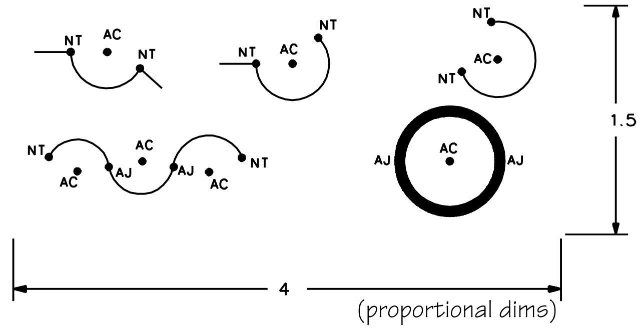


Fig. 5-17 ARC CENTER RELOCATION (After)

Envelope Points

Envelope points are relocated by horizontal and vertical dimensions and affect the arc centers they are associated with. (See arc center relocation.)

Tangent Points

Tangent points are relocated with respect to the arc they are on. The arc may be relocated by horizontal and vertical dimensions or an associated envelope point. The new tangent point is established on the new arc at the same radius or at the new radius if there is a radial or diameter dimension on the arc.

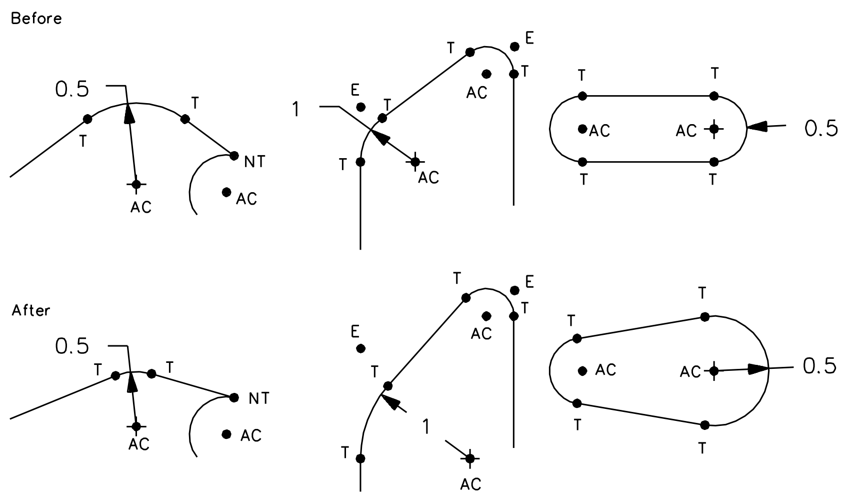


Fig. 5-18 TANGENT POINTS

In some cases maintaining tangency may not be possible. Depending on system parameters, a warning message may appear when tangency is lost.

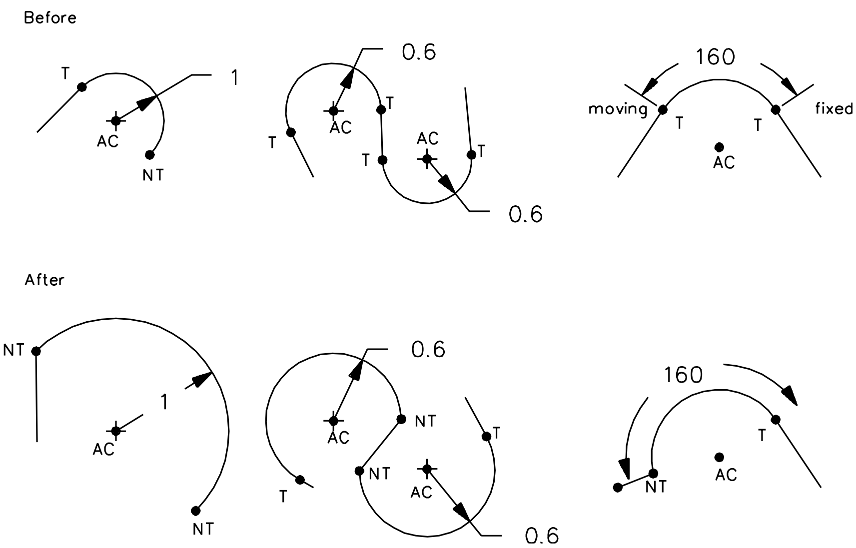


Fig. 5-19 LOST TANGENCY

Non-tangent Points

For non-tangent points (NT) you must first refer to the arc center (AC) the NT point is on. If it is the type of AC that is relocated by its endpoints, the NT points are relocated by the horizontal and vertical dimensions and affect the AC point as described previously.

If the NT is on an arc that is relocated by its center there are two possibilities: one case is where the arc has an angular dimension and the other is the case where there is not an angular dimension.

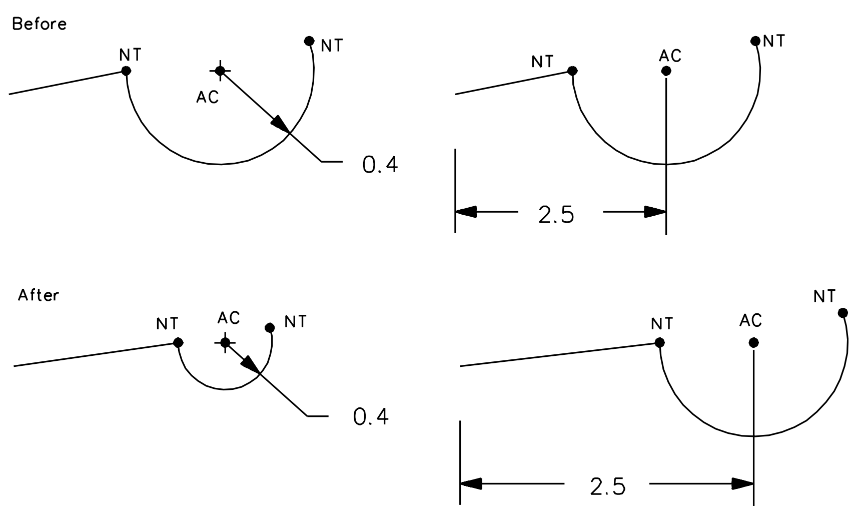


Fig. 5-20 NON-TANGENT POINTS

First the arc center is relocated according to horizontal and vertical dimensions. If there is no angular dimension the NT point is placed at the same angle from center, at the appropriate radius. If there is an angular dimension, the NT point is placed at the new angle and at the appropriate radius.

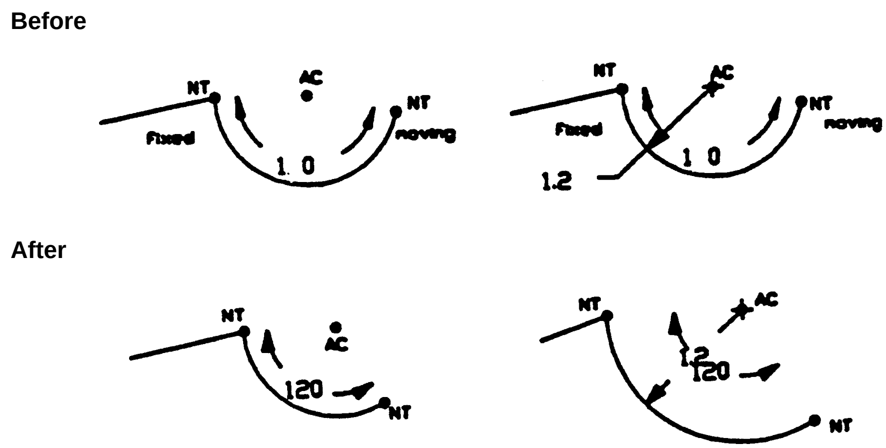
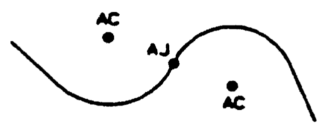


Fig. 5-21 NON-TANGENT POINTS

Arc Junction Points

Arc junction points can be relocated either by horizontal and vertical dimensions, or with respect to the center of one of the arcs it is a part of. Arc junction points relocate with respect to an arc center when one of the arcs has an angular, radial or diameter dimension on its center.



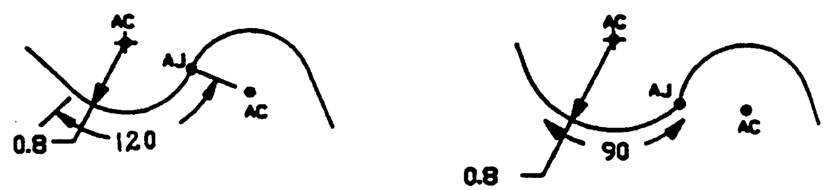
If there isn't an angular, diameter or radial dimension on either arc, the AJ point relocates according to horizontal and vertical dimensions and the AC points relocate with respect to the AJ point.

For other cases, the AJ point relocates with the dimensioned arc, and the second arc adjusts to fit.

Before **After**

AJ point relocates to same angle, new radius with respect to dimensioned arc.

AJ point relocates to new angle, same radius with respect to dimensioned arc.



AJ point relocates to new angle, new radius with respect to dimensioned arc.

Fig. 5-22 ARC JUNCTION RELOCATION

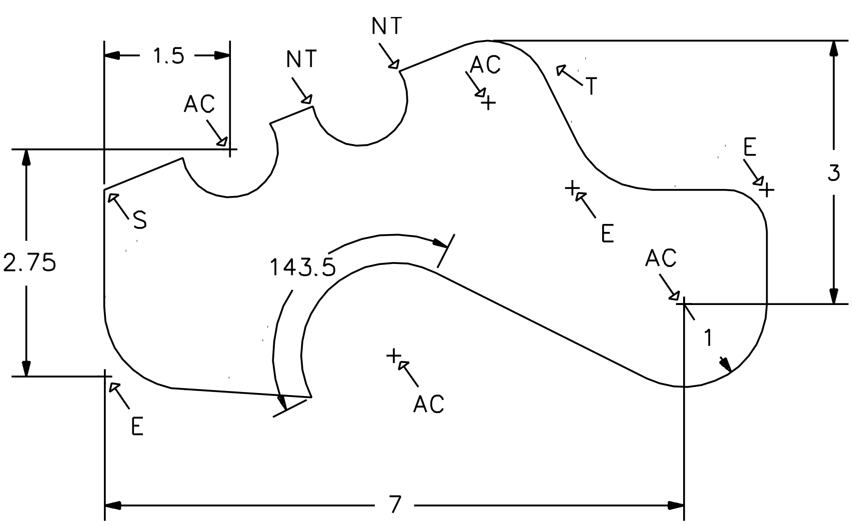


Fig. 5-23 POLYLINE EXAMPLE (BEFORE)

The figure above is a view of a polyline before rectification. It contains many of the control points previously described. The figure below is a view of the polyline example after rectification to new values.

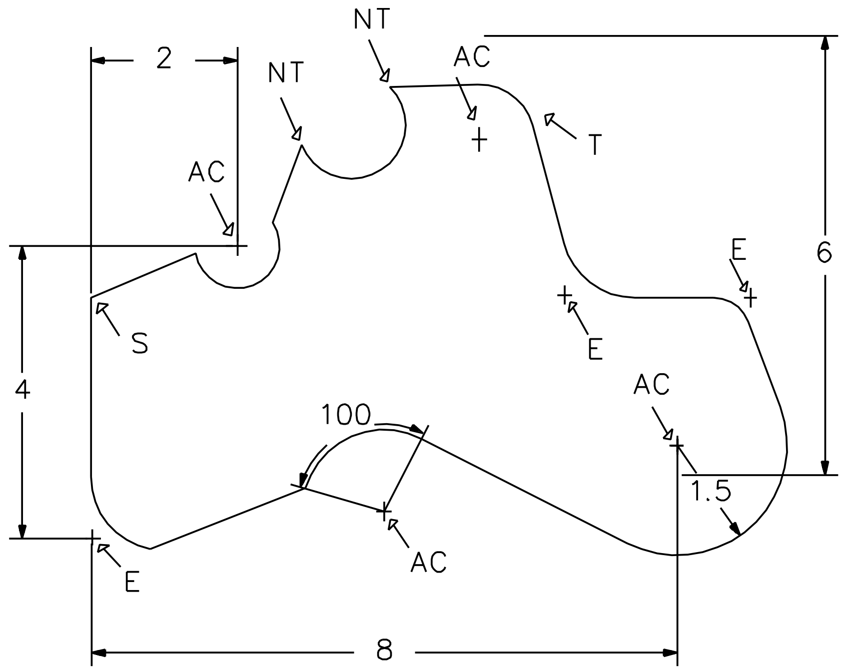


Fig. 5-24 POLYLINE EXAMPLE (AFTER)

Parametric Polyline Width

This function adds parametric width controls to the segments of a polyline. To add parametric polyline width data, from the OfficeCAD dialog box select the Polyline Width button. The Parametric Polyline Width dialog box will appear with the following options: Add Pline Width, Remove Pline Width, View Pline Width, Copy controlled Pline and Exit.

Add Pline Width: To add width to a polyline, select “Add Pline Width.” This will prompt for the following values:

Segment start width:
Segment end width:

Type in a numeric value or a variable, such as [A] for each box. Next, select the segments to be controlled. The segments will highlight as they are selected. Note: only the segments of a single polyline can be selected. To control other polylines, use this option again.

Select all the segments to be controlled by the specified start and ending values or variables. The segments do not need to be selected in order. Unselected segments will retain their original starting and ending widths. When the appropriate segments have been selected, right click to discontinue segment selection. This prompt will appear:

Enter next starting width:

The above may be repeated with a different value or variable for segments of the polyline, or hit Enter when done.

The polyline width data is stored with the polyline in the form of extended entity data. The polyline itself does not change. The new values will be applied when the drawing is rectified, or when it is used as a master drawing in a design.

Remove Pline Width: To remove polyline width control data from a polyline, select "Remove Pline Width." Select the polyline to have width control removed.

View Pline Width: To view the width of a polyline, select "View Pline Width." This will reveal width control data of a selected polyline.

Copy controlled Pline: This will copy any polyline. If width has been assigned to the polyline, the width control data will be copied along with the polyline.

Polyline Pitfalls

Here are some things to avoid when creating or editing a polyline:

- Do not impose an angle of tangency at a vertex. With the Vdraft Properties or the AutoCAD command PEDIT, it is possible to edit a particular vertex and require it to have a particular angle of tangency. OfficeCAD does not recognize this imposed angle of tangency and the rectified result will be wrong.
- Where tangency is important, carefully construct your polyline to ensure arcs are tangent to straight segments. In some cases, it may be easier to draw the polyline with tangencies and then rectify it to achieve the appropriate dimensions. If it is not tangent in the prototype drawing, it will not be tangent in the final drawing.
- Avoid dimensioning a drawing such that there is no logical resolution. For example, it is impossible to draw a tangent to an arc if you are starting from inside the circle that the arc defines. Snap to an arc segment's exact center if you want the center to be a control point.
- Avoid providing a polyline with too many dimensions. In general, if one segment of a polyline is constrained to conform to a dimension, its adjacent segments should be left free to adapt to the change.
- If the polyline includes two adjacent arcs, only one should be radially dimensioned. In a series of arcs, the maximum effective dimensioning would be every other arc. Note that consecutive radially dimensioned arcs are not allowed. Depending on system settings, they will result in an error message.
- Ambiguous dimensions may cause unexpected results.
- If both sides are tangent, and the arc is angularly dimensioned, the degree of arc overrides tangency. (See Figure 5-19.)

Curves

Vdraft can apply a “cubic curve” or “quadratic curve” to a polyline by setting the polyline properties. AutoCAD’s PEDIT command can be used to transform a normal polyline into either a “fit curve” or “spline curve” polyline. In the process, additional vertices are added to the polyline to make it smooth. Figure 5-25 shows a normal polyline and four different curves. (The original polyline is superimposed on the transformed curves to show its relationship with each.) Note that the AutoCAD spline curve yields the same result as the Vdraft cubic curve. With OfficeCAD, the reference points for all curve polylines are the *original* vertices. As with normal polylines, these reference points are relocated according to horizontal and vertical dimensions. However, unlike normal polylines, angular and radial dimensions are ignored.

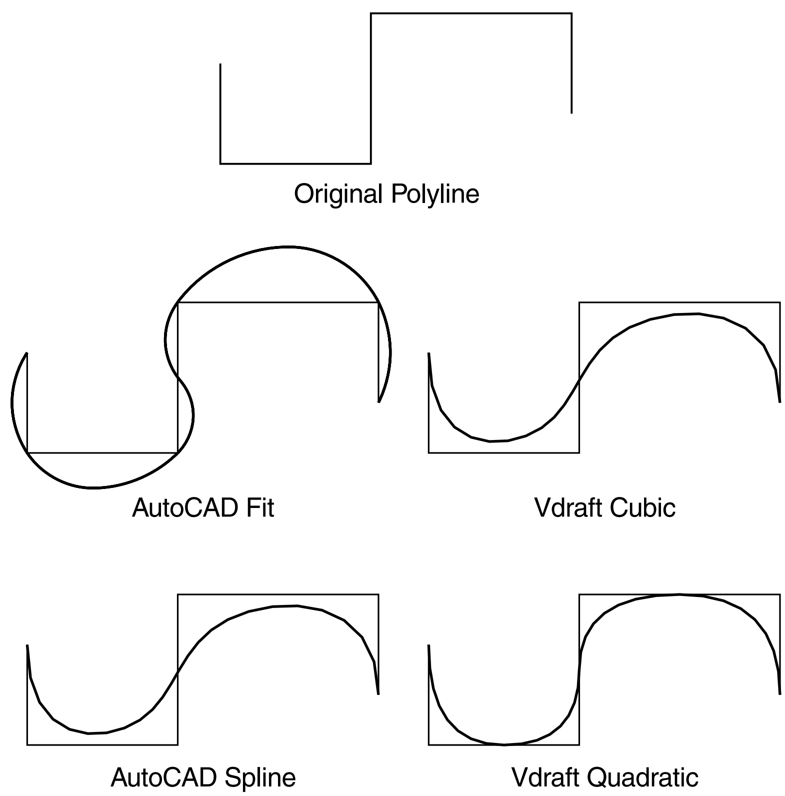


Fig. 5-25 NORMAL POLYLINE, FIT, SPLINE , CUBIC AND QUADRATIC CURVES

Chapter 6 – Blocks and Layers

Parametric Blocks

Parametric blocks can be used in OfficeCAD in two different ways. The first method positions entities within a block as unit, and these entities are not individually rectified. This use of parametric blocks includes:

- Easily moving blocked entities as a group, controlled by location of insertion point.
- Protect blocked entities from dimensional re-sizing via active dimensions.
- Protect blocked text from being affected by the SpecSheet scale factor.
- Protect blocked leaders from getting stretched.
- Assemble variable attribute text into groups.

Attributes from your blocks are retained, and variable attribute values are replaced.

These blocks can be exploded in the rectified output drawing by using the Modify, Explode command in AutoCAD.

In addition to the above parametric block capabilities, you have the ability to use blocks in a more traditional drafting way. OfficeCAD allows blocks to be controlled by the SpecSheet. Blocks may be inserted and various X,Y locations with different X,Y scales and rotations may be specified by the user.

Also, blocks can be substituted and blocks can be erased from a drawing.

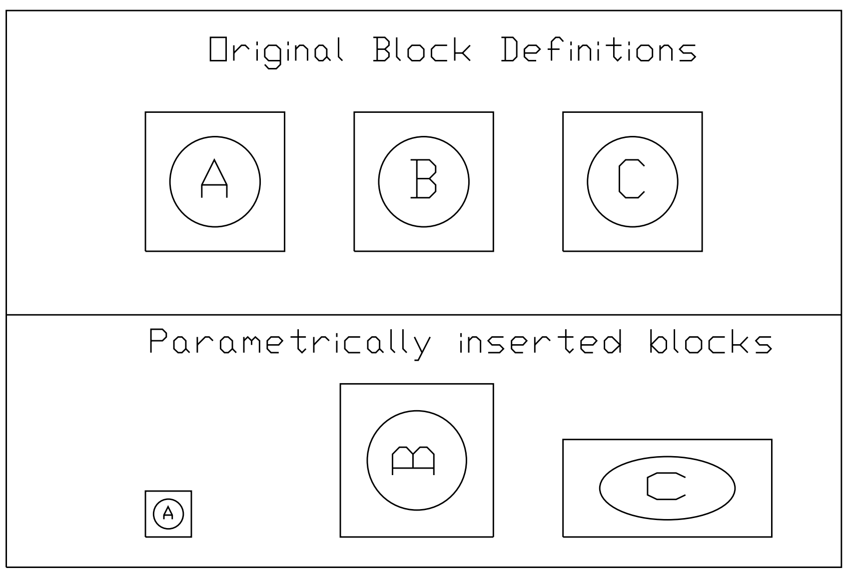


Fig. 6-1 INSERTED BLOCKS

Block Control in Summary

Block Control commands are used by entering them in the SpecSheet.

1. BLOCK_NAME is in the Variable field. Name is the name of the block. It must be preceded by the BLOCK_ to designate to OfficeCAD that this is a block command.

2. In the Measurement field, enter the number related to the type of control desired:
   - -1 = Erase the block.
   - 0 = Replace/substitute with the block named in the text field.
   - 1 = Insert with insertion in text field as X,Y,Xscale,Yscale,rotation,layer.
   - 2 = Insert with insertion in text field as X,Y,Z,Xscale,Yscale,Zscale,rotation,layer.

3. The Text field is determined by the action chosen in the Measurement field.

Block Substitution

With the SpecSheet you can designate one block to replace another block in the output drawing.

For example, a master drawing with a block defined as TABLE and a block defined as SOFA: to replace the inserted drawing TABLE with the block SOFA, place in the Variable field **BLOCK_TABLE**, in the Measurement field place the number **0**, in the Text field, place the name of the replacement block: **"SOFA"** (within quotes).

VARIABLE	MEASUREMENT	TEXT
BLOCK_TABLE	0	“SOFA”

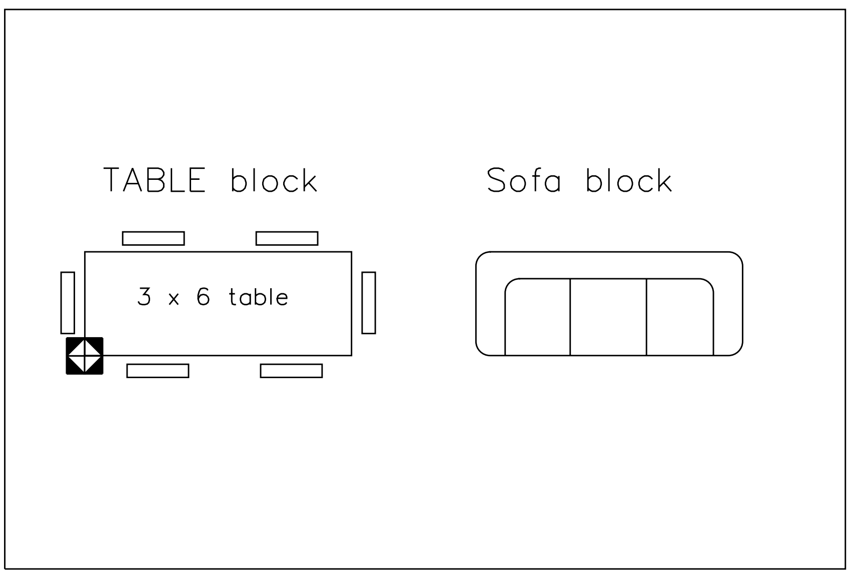


Fig. 6-2 TWO BLOCKS IN MASTER DRAWING

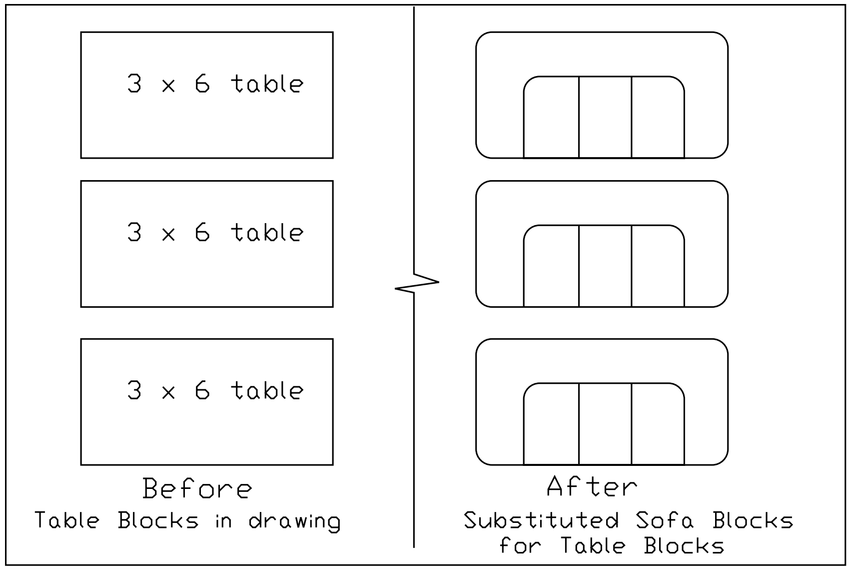


Fig. 6-3 BLOCK SUBSTITUTION

Block Deletion

An existing block in the drawing can be removed or erased by using the block erasure function from the SpecSheet.

Using the Table block, assuming it is defined in the master drawing:

In the Variable column, put the variable **BLOCK_TABLE**, in the Measurement column enter the number **-1**. In the Text column place the word **"ERASE"**. This will cause the block to be erased upon the run of design.

VARIABLE	MEASUREMENT	TEXT
BLOCK_TABLE	-1	"erase"

In the drawing, the block called Table will be erased or removed.

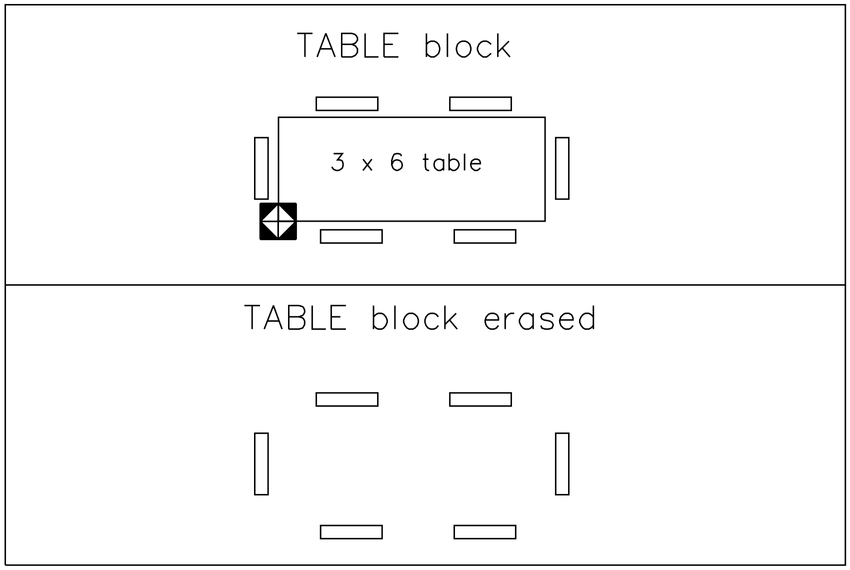


Fig. 6-4 BLOCK DELETION

Block Insertion, Scale and Rotation

To insert blocks into your drawing, you may use pre-defined the blocks in the master drawing (DWG) file, or name an external drawing to be inserted. Another way to insert other drawings into an output drawing is to use the SP_INSERT_OUTPUT$ function in the SpecSheet. This will insert the current SpecSheet's master region information (xins, yins, scale, etc.) into subordinate SpecSheets, then create an output drawing file using the current SpecSheet.

Defining blocks is done with the normal AutoCAD "block" command to group entities together. The blocks can contain variable text and variable attributes.

In the SpecSheet, enter the block control instructions.

For example, to insert a block called "TABLE" into your drawing at coordinates 10,15 (inch or mm units) with a scale factor of X=1, Y=1 and a rotation of 0 on the OBJECT layer:

In the Variable field, type in the name **BLOCK_TABLE**. This designates that this row in the SpecSheet is used to control the Table block.

In the Measurement field in the SpecSheet, enter the number **1**.

In the Text field, type: **"10,15,1,1,0,OBJECT"**.

VARIABLE	MEASUREMENT	TEXT
BLOCK_TABLE	1	"10,15,1,1,0,OBJECT"

The Table block is inserted at coordinates 10,15.
The Table block is scaled at X=1 and Y=1.
The Table block is rotated 0 degrees. Rotation of the block is absolute rotation.
The Table block is inserted on layer OBJECT.

To insert a block using 3-dimensional values called “TABLE” into your drawing at coordinates 10,15,2 with a scale factor of X=1, Y=1, Z=1 and a rotation of 0 on layer OBJECT, enter:

VARIABLE	MEASUREMENT	TEXT
BLOCK_TABLE	2	"10,15,2,1,1,1,0,OBJECT"

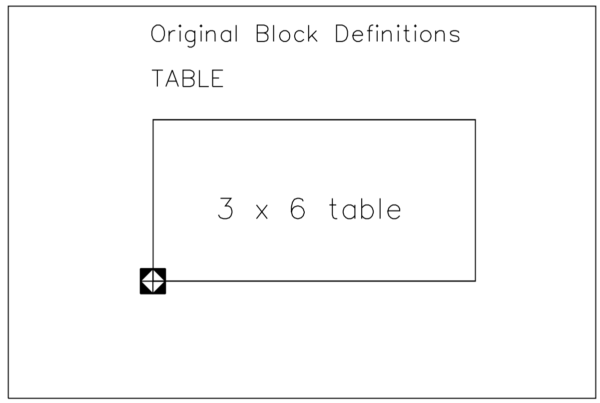


Fig. 6-5 ORIGINAL DRAWING AND BLOCK DEFINITION

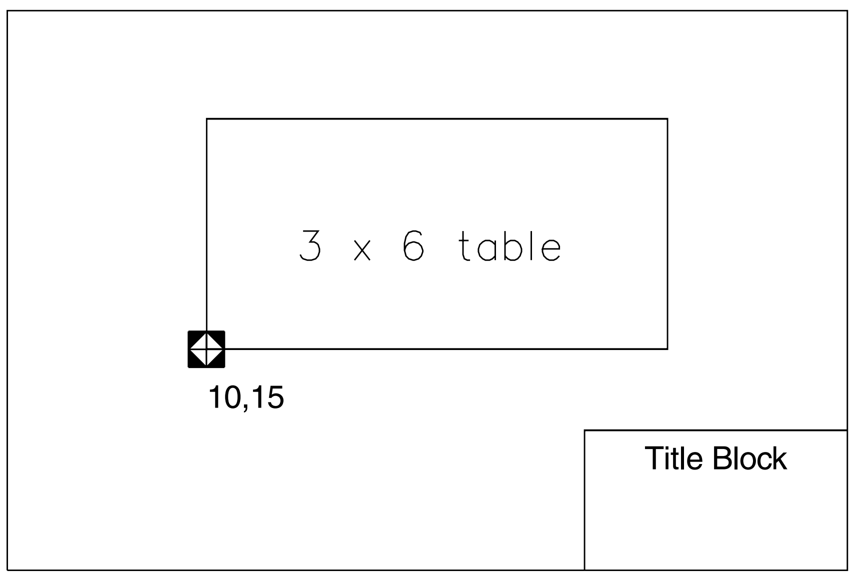


Fig. 6-6 INSERTED BLOCK IN OUTPUT DRAWING

Parametric Layer Control

Layer control commands can be added to the SpecSheet by using the format **LAYER_NAME** in the Variable field, where NAME is the name of the layer to be controlled. The Measurement and Text fields determine the treatment of the layer.

Measurement	Text
0	“freeze”
0	“thaw”
0	“on”
0	“off”
-1	“erase”
1	replacement_layer_name

On/Off, Freeze/Thaw

To turn a layer on or off, or to freeze or thaw a layer in the output drawing, enter the **LAYER_NAME** in the Variable field. In the Measurement field, enter the value **0**. Then, enter the desired action in the Text field: “on”, “off”, “freeze” or “thaw”. For example, to thaw a layer:

VARIABLE	MEASUREMENT	TEXT
LAYER_NAME	0	"thaw"

Erasing a Layer

To erase a layer, enter the **LAYER_NAME** in the Variable field. In the Measurement field, enter the value **-1**. In the Text field, enter the text “erase”.

VARIABLE	MEASUREMENT	TEXT
LAYER_NAME	-1	"erase"

Replacement of Layers

To replace a layer, enter the **LAYER_NAME** in the Variable field. In the Measurement field, enter the number **1**. In the Text field, enter the new layer name.

VARIABLE	MEASUREMENT	TEXT
LAYER_NAME	1	"new_layer_name"

Chapter 7 – Leaders, Text, Attributes

Leaders

When using leaders in drawings, additional planning may be needed to achieve the desired result.

If AutoCAD is used to created the leader, it is treated as two entities. The leader lines and arrow head are treated as one object, and the leader text is treated as a second object. (The text is of type Mtext, and is treated as a single Mtext object even if there are multiple lines of text).

If Vdraft is used to create the leader, each piece of the leader is a separate object. The arrow head is a Solid, the text is a Text entitiy, and each segment of the leader line is a separate Line entity.

In either case, it may be helpful to connect the various leader parts by including them in a block. The base point of the block can be anywhere, but it is recommended that it is positioned at the tip of the arrow head, or snapped to the object the leader points to.

Since the leader text is either an Mtext or a Text object, variable text can be used.

Valid Variable Text

As seen in earlier examples, text replacement is indicated by enclosing a variable name in square brackets, such as [A]. Replacement text can occur in TEXT, MTEXT, ATTDEF, ATTRIBUTE and DIMENSION text. A valid variable begins with a letter, and can contain letters, numbers and the underscore character. The variable can be up to 23 characters long.

In addition such replacement text can be part of a longer phrase:

[A] units

Also, several replacements can be made in one line of text:

The [NAME] is [NUMBER] units long.

Variable replacements have two parts: a numeric value and a text string. Keep in mind that the numeric and text string values do not have to match. In fact, a valid replacement text for a variable can be a blank string, and the numeric value can be zero.

If the drawing contains a variable that does not have a match in the SpecSheet, the text is unchanged.

Variable Math

Variables can also contain mathematical equations. The mathematical operators supported are + - * / ( ). Regular numeric values can be part of the equation. A valid equation:

[A+ B * (C - D)/2.0]

In the above example, the replacement values for A, B, C and D would be used to evaluate the equation. Those variables must exist in the master drawing.

The order of mathematical operations follows the standard order of precedence; namely, operations in parentheses are evaluated first, followed by multiplication and division, then addition and subtraction.

When a variable is an equation, the numeric replacement value is used and the text replacement is ignored. The resulting text is the result of the equation.

Variable Dimension Text

Active and passive dimensions can contain variable text.

In most cases when a passive dimension has its text replaced, the new text value is used, and the numeric replacement value is ignored. If the replacement text is blank, the dimension disappears. If the passive dimension has a variable math equation, the value that results from the equation is used as the replacement text. Remember, passive dimensions never resize as a result of their replacement value, but they can disappear.

When an active dimension has its text replaced, different behaviors can occur.

1. When variable math is used, the value that results from the equation is used to resize/relocate the dimension, and that value is used as the replacement text.
2. When the numeric replacement value is zero, the dimension disappears, as do the entities it controls.
3. When the numeric replacement value is non-zero, but the text replacement string is blank, the dimension relocates/resizes the entities is controls as usual, but the dimension itself disappears.
4. When the text replacement string and numeric replacement value match (and are non-zero, non-blank) the dimension resizes to the numeric value and relocates the entities it controls. The dimension text will be the replacement text string.
5. When the text replacement string and numeric replacement value do not match (and are non-zero, non-blank) the dimension resizes to the numeric value and relocates the entities it controls. The dimension text will be the replacement text string.
6. When no replacement value is found, the dimension keeps its original size and text.

Block and Attribute Text

Attributes in blocks are always replaced. When a block inserted in the drawing contains attributes, any variables appearing as the VALUE of an attribute will be replaced with the text replacement value. Attributes are not visible when given blank replacement text, but they still exist in the drawing. Text in a block is replaced if the block does not have attributes. When a block inserted in the drawing does not contain attributes, any variables appearing in Text, Mtext or Dimension Text in blocks is replaced with the text replacement value.

Notes

Replacement text can be handled via regular text or as block attribute text. Simple text variables can be used as placeholders for notes to be added to the output drawing. Any "extra" variables can be given blank replacement text, which will cause them to disappear.

Chapter 8 – Scale and Tscale

Scaling Options

The scaling options available can affect all the entities in the output drawing, or only affect the "notational" entities, such as text, arrow heads, and tick marks. "Scale" is found in the master section of the SpecSheet. Nscale, Tscale and Legscale are reserved variable names that provide scaling behavior when added to the variable section of the SpecSheet.

Scale

This value, in the master section of the SpecSheet, becomes the new DIMSCALE value of the output drawing. All notational items will be changed according to this formula:

new_height = old_height * SCALE / old_DIMSCALE

Tscale

This option is a multiplier of all entities in the output drawing. The entire drawing is processed, and the TSCALE is applied as a scale in the x, y and z directions from the base point.

VARIABLE	MEASUREMENT	TEXT
TSCALE	10.0	"10.0"

Scale and Tscale are more fully described later in this chapter.

Nscale

This option is a multiplier for the sizes of all notational entities.

new_height = old_height * NSCALE

VARIABLE	MEASUREMENT	TEXT
NSCALE	2	"2"

Legscale

This option controls the dimension leg lengths. This is useful when the output drawing is much larger or smaller than the master drawing.

VARIABLE	MEASUREMENT	TEXT
LEGSCALE	1.5	"1.5"

Scale – Drawing Annotative Scale Factor

Scale has been mentioned earlier in this chapter. The scale factor in the master section of the SpecSheet will override the Dimension Scaling factor, or DIMSCALE, originally assigned to the master drawing. The default scale factor of the SpecSheet is 1.0, but you can enter a value or a formula in that cell that will calculate the appropriate scale factor. Since the design process can result in an output drawing much smaller or larger than the original master drawing, the text and arrowhead sizes may be the wrong size. By changing the Scale value, the dimension and text sizes can be adjusted to an appropriate size.

To edit the Scale of the output drawing in Excel, enter the formula or value in the Measurement column of the Scale row.

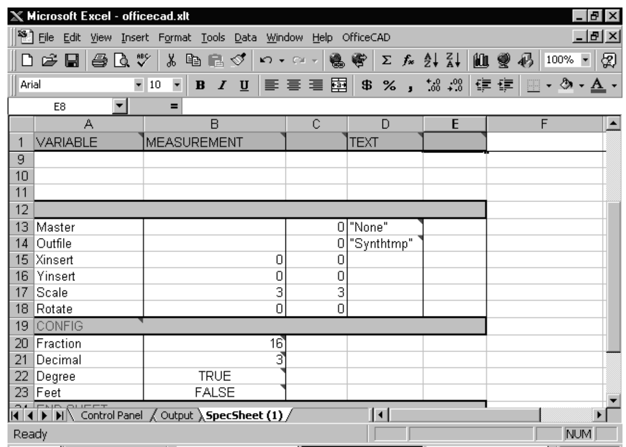


Fig 8-1 SPECSHEET WITH MODIFIED SCALE VALUE IN EXCEL

OfficeCAD expands upon the parametric scaling of your drawing. The scale factor which is found in the SpecSheet controls only annotative information.

This means that changing your scale value to 10 would scale up the size of the non-blocked text, dimensions and dimension text in your output drawing or drawings.

Normally, the scale factor of the SpecSheet is set to match the scale of your master drawing.

Tscale – Total Drawing Scale Factor

Tscale has been mentioned earlier in this chapter. Tscale is a reserved variable name. When added to the SpecSheet design in the Variable column, the Tscale variable will scale up or down *all* entities in your output drawing.

This is useful if you are inserting a detail in an assembly drawing which needs an enlarged or "blown-up" detail. Architectural, structural and civil drawings can have parametric details blown up automatically.

Tscale can also be used to size the final drawing to fit within a drawing border.

Scaling is done from the location of the base point of the drawing.

In this example, the pencil drawing is being combined with the title block drawing. At Tscale = 1, the pencil drawing is too small relative to the title block.

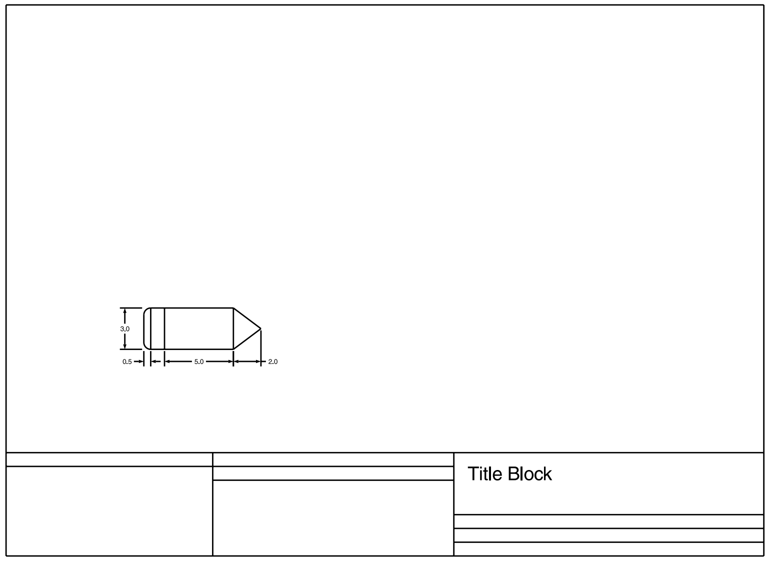


Fig. 8-2 RESULT WITH TSCALE = 1.0

When the design is changed, with Tscale = 4, a better result is obtained.

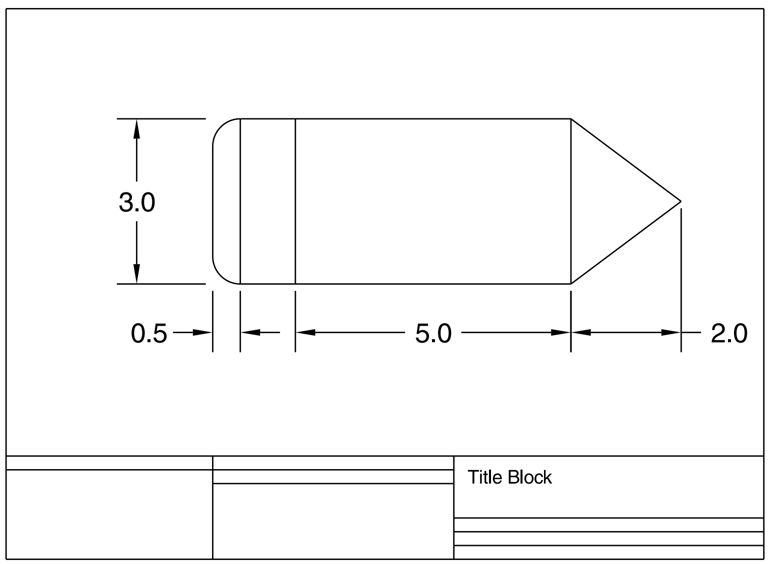


Fig. 8-3 RESULT WITH TSCALE = 4.0

Chapter 9 – Using Excel

Autospec Creation of Your Excel Worksheet

OfficeCAD enables you to use Microsoft Excel as your design front end for creating new AutoCAD drawings. This means you can use your existing Excel worksheets to create new AutoCAD drawings for proposals or shop drawings or quotes.

OfficeCAD creates an intermediate worksheet which can be connected to your existing worksheets to enable new AutoCAD drawing creation.

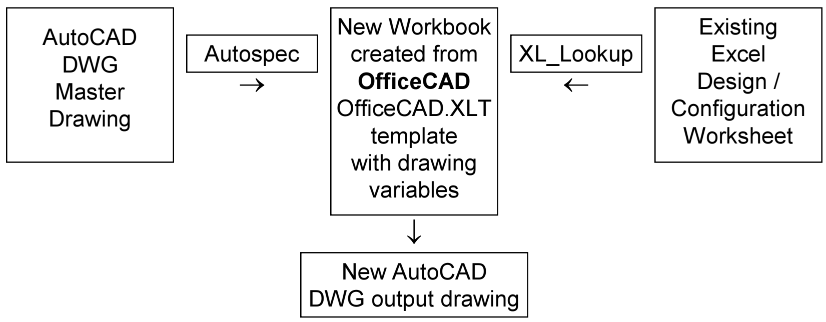


Fig. 9-1 OVERVIEW OF OFFICECAD USING EXCEL PROCESS

Shown below is the first sheet of a new OfficeCAD workbook created using the OfficeCAD template (OfficeCAD.xlt) file.

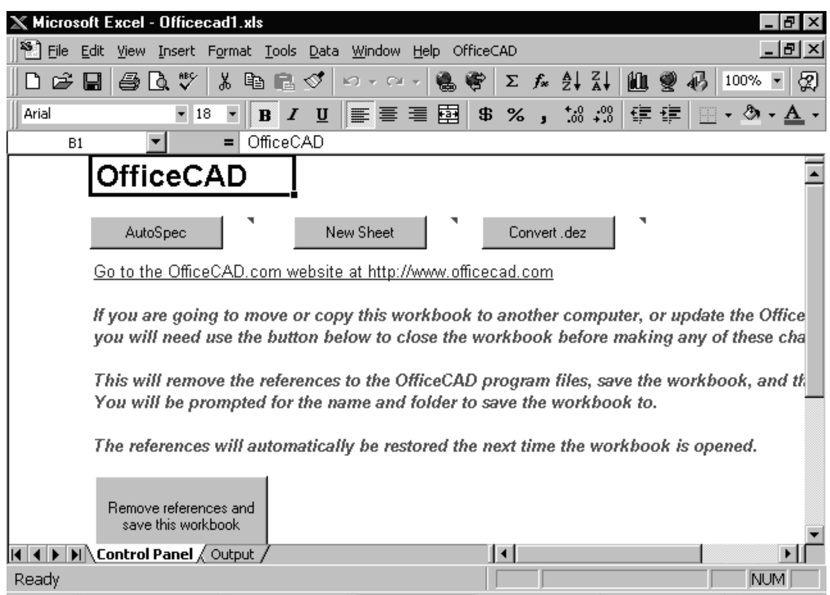


Fig. 9-2 OFFICECAD.XLT CONTROL PANEL

OfficeCAD with Excel uses a five-column worksheet layout. Three columns are used for variables, measurement (numeric) and text expressions. Two columns are used for displaying the output from the numeric and/or text expressions.

VARIABLE contains the variables from your master drawing and variables local to the SpecSheet.

MEASUREMENT contains numeric expressions used for generating new numeric values for your variables. The adjacent column displays the current value generated by a formula, prompt, import, etc.

TEXT column generates string values for your variables. The adjacent column displays the current value which is generated.

Typical steps to creating your Excel-based automated drafting solution with OfficeCAD are as follows:

1. Create a master drawing in AutoCAD or Vdraft using OfficeCAD variable dimensions.
2. Autospec the master drawing to automatically extract the variables into a new OfficeCAD worksheet.
3. If you choose, connect the OfficeCAD worksheet variables to the appropriate cells in your existing Excel worksheets.
4. Calculate your design and view the drawing.

If you are converting existing Specit SpecSheets to Excel worksheets, you can use the "Convert .dez" button on the control panel worksheet, or choose the command from the OfficeCAD menu (select OfficeCAD, New Sheets, Convert ".DEZ").

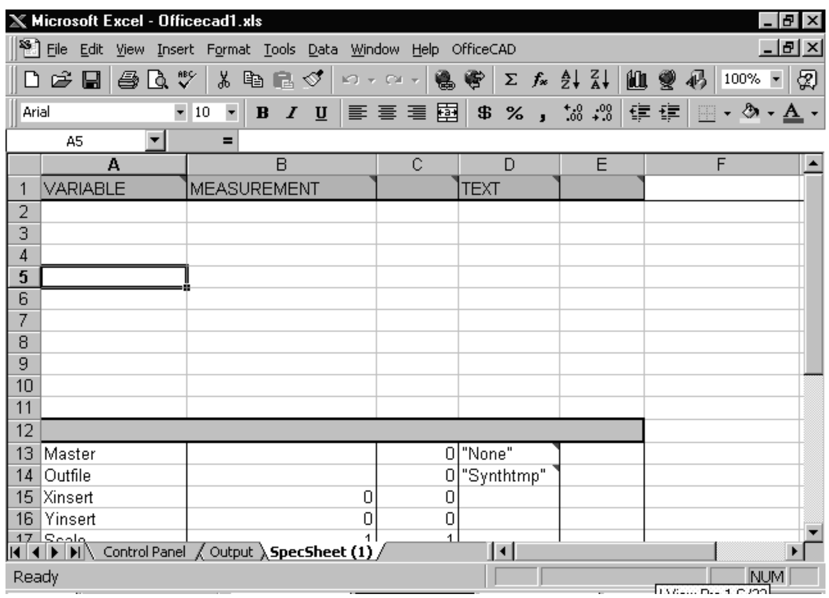


Fig. 9-3 EXCEL/OFFICECAD WORKSHEET

With the OfficeCAD Excel template (OfficeCAD.xlt) you can create new workbooks for use in creating new parametrically controlled drawings. These workbooks will load the OfficeCAD menu add-in (OfficeCAD.xla). A new menu item named "OfficeCAD" will appear when you have an OfficeCAD workbook in view.

To create a new OfficeCAD workbook select from the main Excel menu File, New, select the OfficeCAD tab and double-click on OfficeCAD.xlt.

OfficeCAD Menu in Excel

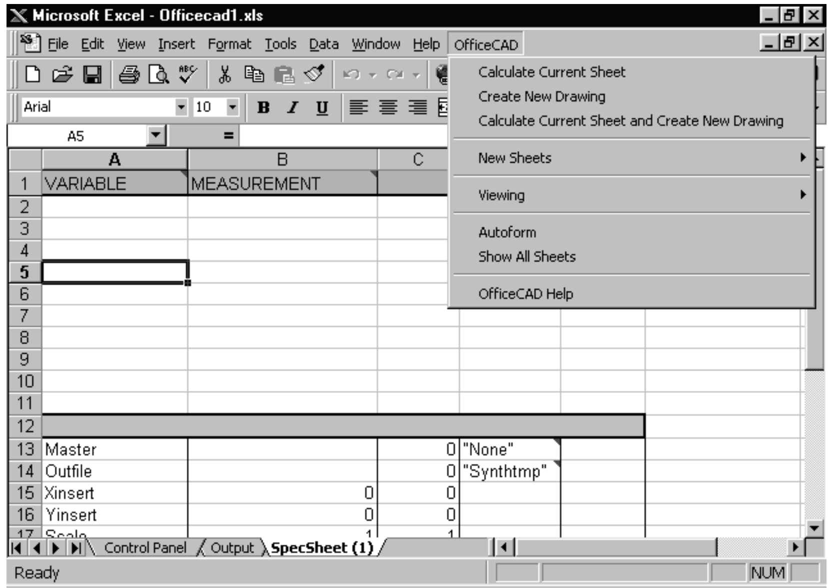


Fig. 9-4 OFFICECAD MENU IN EXCEL

From this menu, you can select Calculate Current Sheet, Create New Drawing, Calculate Current Sheet and Create New Drawing, New Sheets and Viewing. New Sheets and Viewing have submenus.

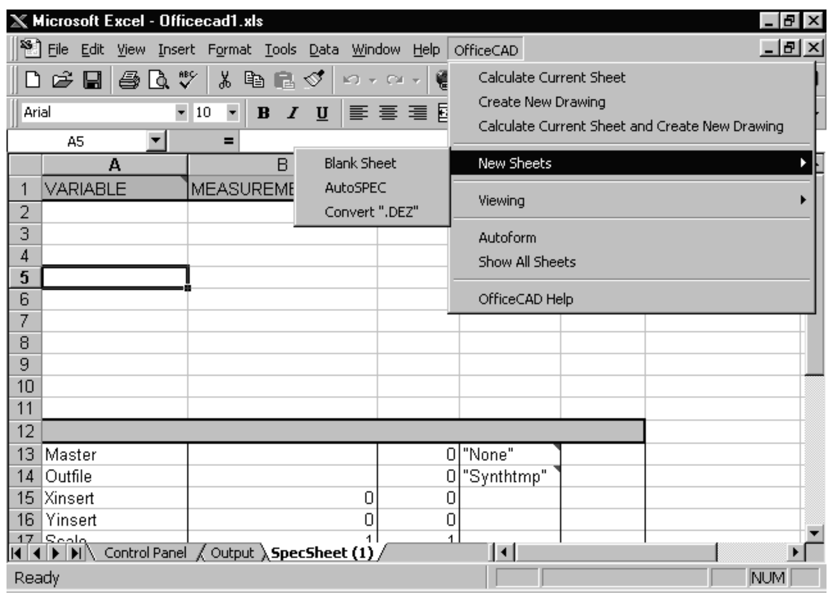


Fig. 9-5 NEW SHEETS MENU

From the New Sheets menu, you can:

1. Create a new empty OfficeCAD worksheet using **Blank Sheet**.
2. **AutoSPEC** an AutoCAD DWG master drawing and extract all variables into a new OfficeCAD worksheet. Optionally, you can extract some or all layers and block names as variables into the worksheet.

3. **Convert ".DEZ"** to convert existing SpecSheet DEZ file(s) into new OfficeCAD worksheets. Convert a single design or entire folder.

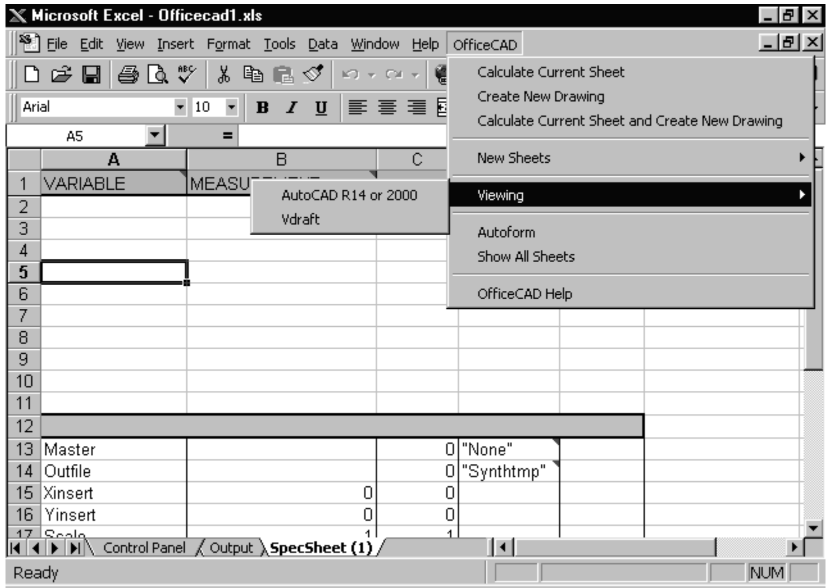


Fig. 9-6 VIEWING MENU

From the Viewing menu, you can select how to view the output drawing. You can choose AutoCAD R14 or 2000, or Vdraft. Microsoft's Office 97 suite of programs included the AutoCAD DXF Import Filter which could be optionally installed along with the Office 97 programs. If you don't have Office 97 you may be able to download the filter from Microsoft's website and then install it. Depending on your computer's configuration, this may or may not work on your system.

Autospec Dimension Text Formats

OfficeCAD supports the dimension text formats shown below (with sample output):

FORMAT	OUTPUT (inches)	OUTPUT (feet)	DESCRIPTION
ARCH(12.1875, 16)	1'0 3/16"	12'-2 1/4"	feet/inch
ARCH_ID(12.1875, 16)	1-0-3/16	12-2-1/4	feet/inch ID
ARCH_ID0(12.1875, 16)	1-0-3/16	12-2-1/4	feet/inch ID, no leading 0
ARCH_ID1(12.1875, 16)	12-3/16	146-1/4	inch ID
ARCH_NZ(12.1875, 16)	1'0 3/16"	12'2 1/4"	feet/inch, no leading 0
ARCH_NZ0(12.1875, 16)	1'0 3/16"	12'2 1/4	feet/inch, no ", no leading 0
ARCH0(12.1875, 16)	1'0 3/16	12'-2 1/4	feet/inch, no "
ARCHI(12.1875, 16)	12 3/16"	146 1/4"	inch
ARCHI_NZ(12.1875, 16)	12 3/16"	146 1/4"	inch, no leading 0
ARCHI_NZ0(12.1875, 16)	12 3/16	146 1/4	inch, no ", no leading 0
ARCHI0(12.1875, 16)	12 3/16	146 1/4	inch, no "
ENG(12.1875, 16)	1'-0.1875"	12'-2.2500"	engineering feet/inch
STR_N(12.1875, 16)	12.188	12.188	decimal string
NONE			

Excel Design: Window_comp Example

The Window_comp Design example uses prompts for width and height. The drawing is geometrically rescaled and can be brought up from inside of Excel. A table of dimensions is used. The following screen shows a sample Excel worksheet using OfficeCAD. Window_comp.xls is included in OfficeCAD in the Demos folder.

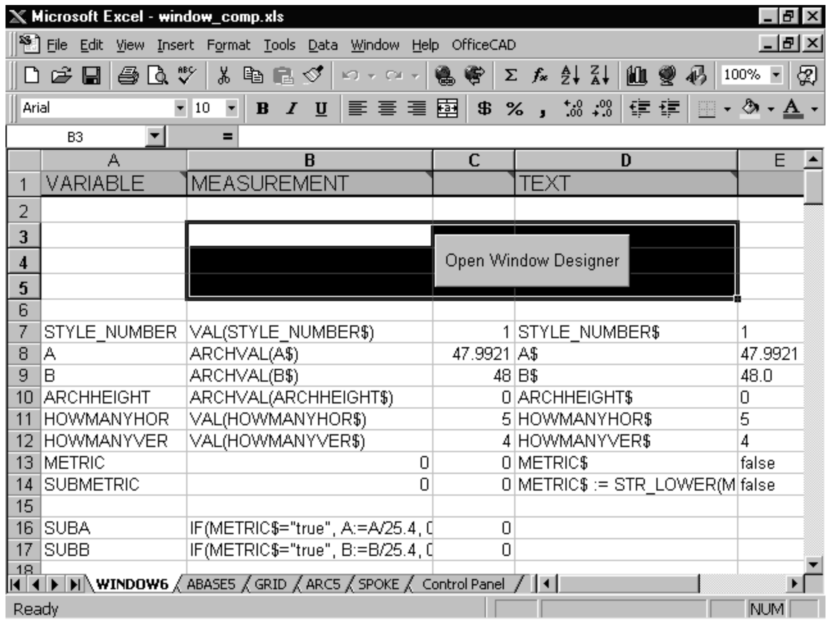


Fig. 9-7 EXCEL WINDOW WORKSHEET

Select Open Window Designer. The following design dialog box will appear:

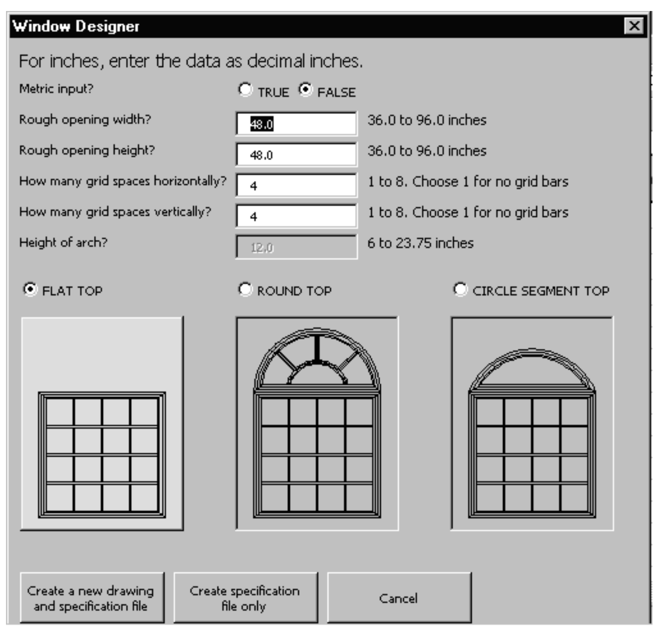


Fig. 9-8 WINDOW DESIGNER USERFORM

This userform prompts for various attributes of a window, including measurements and style. Enter the values for the rough opening width and height, and the number of horizontal and vertical grid spaces. Depending on the style of window, the arch height value may be prompted.

In this example, let's create a drawing for a window that is 4' high by 6' wide, with a grid of 5 x 5 and a round top. Enter the rough opening values in metric inches: set width to 72.0 and height to 48.0. Enter the desired number of grid spaces: set horizontal to 5 and vertical to 5. Now select "round top".

You are ready to create your drawing. On this form, you also have the options of creating a specification (SPC) file without creating the drawing, or canceling the operation. Select the "Create a new drawing and specification file" button.

The values and attributes from the userform are run through the window worksheet, and a window drawing is created.

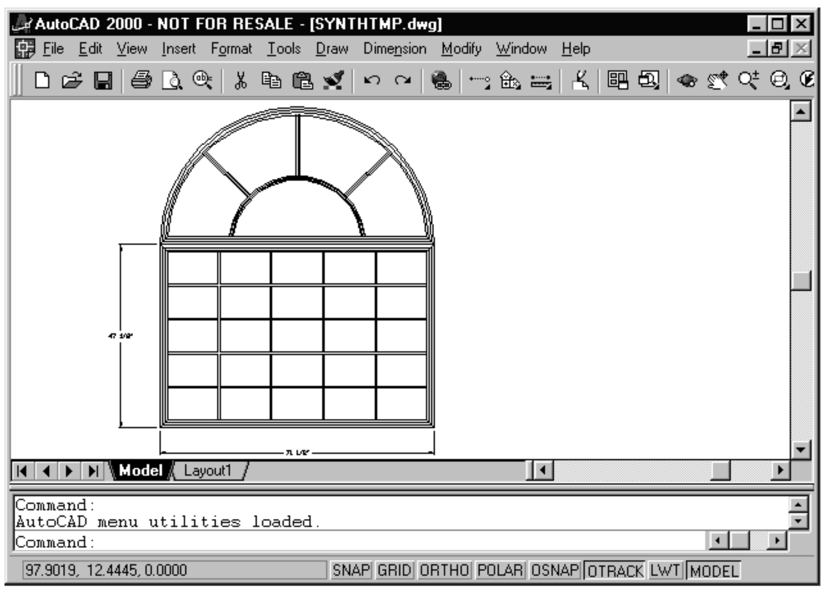


Fig. 9-9 WINDOW_COMP EXAMPLE OUTPUT DRAWING IN AUTOCAD

This drawing was created to the specifications in the Window Designer. The vertical and horizontal measurements have been calculated to be one half inch shorter than the input values, as is the standard.

If you are familiar with Microsoft Visual Basic, you can program your own userforms like this one. View this form in the Visual Basic Editor, if installed.

More Quick Examples

Try a quick example: Window_comp Design (above), Window_simple Design, Cylinder Design, 3D Awning Frame Design, Excel to AutoCAD Table Example.

Window_simple Design

This design uses Excel's graphic user interface to select the size and style for a custom window. In this example, a window drawing will be created using window dimensions entered and the window style selected.

Cylinder Design

This design uses Excel's graphic user interface to select the size and options for a pneumatic cylinder. This example uses variable text and variable layer control to design a cylinder. No geometric rectification is used in this example.

3D Awning Frame Design

This is a 3D rectified drawing with multiple drawing insertions using parametric arrays of entities for the frame.

Additional features for the Excel portion of OfficeCAD may be found at our web site.

Connecting Your Worksheets

In this section, connecting your Excel / OfficeCAD worksheet to your existing worksheets is demonstrated.

NOTE: Any regular Excel functions in the five OfficeCAD columns will cause an error. Excel functions can be used outside of the OfficeCAD columns. You can use the values from those functions by importing the values with either the XL_LOOKUP or XL_LOOKUPRC functions.

With the Excel / OfficeCAD worksheet, you can connect your variables which come from your drawing to any existing Excel worksheets you may have. This is done from the OfficeCAD worksheet using the XL_Lookup function. Parameters are XL_Lookup(cellname, sheetname, [workbook]). Workbook name is optional. OfficeCAD will default to the current workbook if the workbook name is left off.

> Example: XL_Lookup("A23","Sheetname") retrieves the value from cell A23 in the Excel worksheet "sheetname" in the current workbook.

You can also add the third parameter to have OfficeCAD look in a different workbook for the value you wish to use. The workbook must already be open.

In this way, your Excel worksheets can begin to create new AutoCAD drawings automatically in combination with OfficeCAD. Use OfficeCAD to extract your data directly from existing worksheets.

OfficeCAD provides additional functionality for your worksheets. This includes prompting, parametric arrays, text string conversion for dimensions, custom functions, layer control, block control, scaling, t-scaling, rotation, and subordinate master drawing insertion.

OfficeCAD adds approximately 300 functions to Excel. These new functions are specifically useful in parametric drawing applications. Excel further provides excellent calculation tools and a sophisticated graphic user interface for creating complex designs.

Of course, you can use any and all of Excel's other capabilities, such as linking to other Microsoft Office programs, such as Access.

Importing Excel Tables into Your Drawing with OfficeCAD

There are two functions to import a table from an Excel worksheet into your output drawing: CopyTable and CopyTableRC. The difference between these two functions is that CopyTable uses “A1” (cellname) cell reference, and CopyTableRC uses the “R1C1” (row and column) cell reference.

The parameters for each, with optional paramaters in brackets, are:

CopyTable(Worksheet$, UpperLeftCorner$, LowerRightCorner$, BorderStyle$, TextAlignment$, WidthFactor#, HeightFactor#, TextScale#, XInsertion#, YInsertion#, [Outfile$], [Workbook$])

CopyTableRC(Worksheet$, UpperLeftRow#, UpperLeftColumn#, LowerRightRow#, LowerRightColumn#, BorderStyle$, TextAlignment$, WidthFactor#, HeightFactor#, TextScale#, XInsertion#, YInsertion#, [Outfile$], [Workbook$])

**Worksheet$**: The name of the worksheet the table will be copied from. If optional Workbook$ is not used then Worksheet$ is assumed to be a worksheet in the current workbook. Use Workbook$ if Worksheet$ is from a different workbook. You will have to specify Outfile$ if you use Workbook$.

UpperLeftCorner$ (CopyTable): This is the absolute cell reference for the upper left corner cell for the range to be inserted into the output drawing. An example absolute cell reference is “A1”.

LowerRightCorner$ (CopyTable): This is the absolute cell reference for the lower right corner cell for the range to be inserted into the output drawing. An example absolute cell reference is “E14”.

UpperLeftRow# (CopyTableRC): The row number for the upper left corner cell of the table to be copied. R1C1 cell reference.

UpperLeftColumn# (CopyTableRC): The column number for the upper left corner cell of the table to be copied. R1C1 cell reference.

LowerRightRow# (CopyTableRC): The row number for the lower right corner cell of the table to be copied. R1C1 cell reference.

LowerRightColumn# (CopyTableRC): The column number for the lower right corner cell of the table to be copied. R1C1 cell reference.

BorderStyle$: There are four choices for border style in the output drawing: “T” (thin), “K” (thick), “D” (double), and “N” (none). These values are not case sensitive.

TextAlignment$: There are three choices for how the text is aligned in the cells in the output drawing: “L” (left), “R” (right), and “C” (center). These values are not case sensitive.

WidthFactor#, **HeightFactor#**: These multipliers help determine the actual width and height of the cells in the output drawing. Start with 0.08 and see how big the table is in the output drawing, then work up or down from there to adjust the size of the cells in the drawing. 0.08 will roughly correspond to a TextScale of 1 to match the cells with the text size in the output drawing.

TextScale#: First adjust the WidthFactor# and HeightFactor# to make the table fit in your output drawing. Then adjust the TextScale# to make the text fit in the cells.

XInsertion#, **YInsertion#**: These are the X and Y coordinate values for the insertion point (X,Y) of the upper left corner of the table in your output drawing.

**Outfile$**: This is the filename for the output drawing. If it is not specified, the Outfile$ vlue from the current SpecSheet will be used. If the optional Workbook$ value is used, Outfile$ must have a value specified.

**Workbook$**: This is the path and filename for the workbook which contains Worksheet$. If it is not specified, the current workbook will be used.

Example:

COPYTABLE (“sheet1”, “a1”, “c9”, “t”, “c”, .08, .08, 1, 20, 20)

This will copy the range of cells in Sheet1 from A1 to C9, and place this range in the output drawing at the point (20,20).

Chapter 10 – Advanced Methods for Creating Output Drawings

Using Other Programs

The OfficeCAD system of creating output drawings from master drawings is open to customization. There are many methods that can be used to create input to the drawing creation process. These methods will allow you to connect OfficeCAD power to the tools you already use. One of the most powerful is the creation of specification (SPC) files, described later in this chapter.

Tables and Files

The OfficeCAD tools for Microsoft Excel that are available in OfficeCAD.xlt and OfficeCAD.xla include many tools to read and write files and tables. If you have files or tables associated with your drawings, you may find these tools helpful in using or including that data in drawing creation. Likewise, if you want the information entered or calculated by the design to be saved into a file or a table, these tools are also available. To learn more about these tools, open Excel and create a workbook from the OfficeCAD template (OfficeCAD.xlt). From the OfficeCAD pull down menu, select “OfficeCAD help.” The help file contains descriptions of all the tools available, with sections on “File Functions” and “Table Functions.”

Block Libraries

If you have blocks that you already use, these can be utilized by OfficeCAD. An Excel spreadsheet can be directed to use blocks parametrically. This is described in more detail in **Chapter 6**. If you are not using Excel, read the section below on Specification Files and how parametric blocks can be used.

Specification (SPC) Files

The specification (SPC) file specifies the master drawings, output drawings to be created, the replacement values to be used and other drawing controls. This is an ASCII file. Any SPC file of the correct form is valid for use with OfficeCAD master drawings. SPC files can be generated from Excel, a custom program, or written by hand in NotePad or in another text editor. OfficeCAD provides Excel with the ability to easily create SPC files. You may wish to create a SPC file from Excel and compare it to the information below. (See **Chapter 9** for information on using Excel.)

To test your SPC file, from the AutoCAD command prompt enter “OFFCAD,” then select “SpecSheet,” “Choose.” You may then browse and select the SPC file.

From Vdraft, select the “Extensions” pull down menu, then “OfficeCAD,” “Run Design.” Browse and select the SPC file.

The format of the SPC file follows. The variable replacement values are listed in the VARSEC and drawing controls are listed in the MASTSEC. The most basic SPC files will have one VARSEC followed by one MASTSEC, or a series of VARSEC-MASTSEC pairs.

General Format:	Example:
VARSEC	VARSEC
[NAME1]	[X]
value1	10.5
text1	10' - 6"
[NAME2]	[Y]
value2	1.0
text2	1.0
ENDSEC	ENDSEC
MASTSEC	MASTSEC
output_DWG_name	newhouse
input_DWG_name	house
x-insertion-value	0.0
y-insertion-value	0.0
scale	1.0
rotation	0.0
ENDSEC	ENDSEC

In the general format column, the file starts with “VARSEC,” which indicates the “variable section” at the beginning of a VARSEC-MASTSEC pair. (If the master drawing has no variables, the SPC file could start with a MASTSEC.) In this variable section there are only two variables, [NAME1] and [NAME2]. However, there could be none or an unlimited number of variables. After each variable name is a value line and a text line. The value line must be a numeric value or zero. This is true even if this variable is not used in the drawing, or if this variable is used for text replacement only. The text line can contain any text, or can be blank. Note that in the Example column, the [X] variable has a value of 10.5, and the text is entered as 10' - 6". Each variable entry is separated from the next by a blank line. After all the variables are listed, the keyword ENDSEC indicates the end of that section. Next line is the beginning of the MASTSEC, or Master Section. The master section has six lines that must appear (in the following order):

1. Name of the output drawing to be created.
2. Name of the input drawing.
3. *x* insertion point (numeric value).
4. *y* insertion point (numeric value).
5. Annotation scale value (numeric value).
6. Rotation value, in degrees (numeric value).

These six lines are followed by an ENDSEC line. Think of these values like block insertion values. This input drawing can be placed in the output drawing at a specified *x*,*y* point with the desired rotation.

Another VARSEC-MASTSEC pair may follow. Important note: if two or more VARSEC-MASTSEC pairs within a single SPC file have the same output drawing name, the effect will be that of creating a composite output drawing. All drawings that target the same output drawing will be combined. A single SPC file is treated as a single pass through the OfficeCAD drawing generator. If there are multiple VARSEC-MASTSEC pairs, these could create a series of separate output drawings, or could combine input drawings into one or more composite output drawings.

The VARSEC can contain simple variable information, and it can also contain additional instructions, such as parametric layer, parametric block and scaling instructions:

Reserved Variables

Any of these reserved variables (shown in examples that follow) can be used in the VARSEC portion of the SPC file: ZINSERT, DRAWING_VIEW_MASTER, NSCALE, TSCALE, LEGSCALE, DWG_CIRC, DWG_RECT, and DWG_STEP. Variable names beginning with the keywords LAYER_ and BLOCK_ also function as drawing controls. Incorrect use of these variable names may have unpredictable results.

ZINSERT

The variable name ZINSERT is a reserved name. When it appears in the VARSEC, it is treated as the *z* insertion value for the input drawing into the output drawing.

```
[ZINSERT]
5
5
```

Output Drawing View

Since an output drawing may be composed of many input drawings, it may be helpful to specify which of the input drawings should control the default view. You may have an input drawing with viewports, or that defaults to paperspace display, that reflects the type of view you would like to have the output drawing. You can specify which drawing has view control by adding the following to the variable section for that drawing:

```
[DRAWING_VIEW_MASTER]
0
0
```

If no such variable appears in any VARSEC for a given output drawing, the views will be controlled by the first input drawing listed in the SPC file for that output drawing.

Parametric Layers

Instructions can be given in the VARSEC to control layers in output drawings.

- Layers can be turned on, off, frozen or thawed as they would be by AutoCAD commands.
- Layers can be renamed. All entities on the layer being renamed will be moved to the new layer.
- Layers can be erased, and in turn all entities on that layer will be erased as well.

The standard “variable, value, text” fields are utilized. The variable field contains the usual square brackets [ ], and the keyword “LAYER_” precedes the name of the layer to be controlled. The value field contains -1, 0 or 1 depending on the action desired. The text field contains the action keyword or the new layer name.

[LAYER_name] 1 new_layer_name	[LAYER_name] -1 ERASE
[LAYER_name] 0 ON	[LAYER_name] 0 OFF
[LAYER_name] 0 FREEZE	[LAYER_name] 0 THAW

Parametric blocks

Instructions can be given in the VARSEC to control blocks in the output drawings. Blocks can be erased, replaced or inserted. The standard “variable, value, text” fields are utilized. The variable field contains the usual square brackets [ ], and the keyword “BLOCK_” precedes the name of the block to be controlled. The value field contains a -1 to erase, 0 to replace, or 1 to insert. The text field contains the keyword action, or the insertion data. Replacement blocks and blocks to be inserted must be defined in the master drawing, or be in the default path, or given a full path along with the block name (e.g., c:\myblock\newblock).

Block control format:

[BLOCK_name]
-1
ERASE

[BLOCK_name]
0
replacement_block_name

[BLOCK_name]
1
x-insert, y-insert, x-scale, y-scale, rotation, layer-name

Block insertion sample:

[BLOCK_MYBLK]
1
100,30,1,1,45,mylayer

Scaling Options

The scaling options available in the SPC file can affect all the entities in the output drawing, or only the “notational” entities. The entities that are “notational” are text, arrowheads, tick marks, etc.

- SCALE - the fifth item in the MASTSEC. This value becomes the new DIMSCALE value of the output drawing. All notational items will be changed according to this formula:

    new_height = old_height * SCALE / old_DIMSCALE

- NSCALE - placed in the VARSEC. This value is a multiplier for the sizes of all notational entities:

    new_height = old_height * NSCALE

- TSCALE - placed in the VARSEC. This value is a multiplier of all entities in the output drawing. Essentially, the entire drawing is processed, and the TSCALE is applied as a scale in the *x*, *y* and *z* directions from the base point.

- LEGSCALE - placed in the VARSEC. Controls the dimension leg lengths. This is useful when the output drawing is much larger or smaller than the master drawing.

Examples of the scale controls in the VARSEC section:

```
[NSCALE]
2.0
2.0

[TSCALE]
10.0
10.0

[LEGSCALE]
1.5
1.5
```

Multiple Drawing Insert

Normally, one VARSEC-MASTSEC pair creates one modfied version of the input drawing placed in the output drawing. When it is useful to have many copies of the same modifed drawing in the output drawing, you can either repeat the VARSEC-MASTSEC or apply one of the methods listed here. This provides a quick way to insert multiple copies of the modified drawing (unlike multiple block insert, which inserts multiple copies of the unmodified block). Options are provided for a circular, rectangular or stepped array. The entries appear as follows. Note that the value field is different for each type of array:

```
[DWG_CIRC]
0
radius, start-angle, angle-increment, number-of-items
```

The center of the circular array is the *x*-insert, *y*-insert point listed in the master section. The first copy of the modfied input drawing will be placed at specified radius and start-angle away from the *x*-insert, *y*-insert point. The next will be at the specified radius and (start-angle + angle-increment) away from the *x*-insert, *y*-insert point. It will continue adding angle-increment each time until number-of-items have been placed.

```
[DWG_RECT]
1
x-increment, y-increment, number-of-columns, number-of-rows
```

The first copy of the modfied input drawing of the rectangular array will be placed at the *x*-insert, *y*-insert point listed in the master section. It will then use the *x*-increment and *y*-increment values to create the number-of-rows and number-of-columns specified.

[DWG_STEP]
2
x-endpoint, y-endpoint, number-of-items

The first copy of the modfied input drawing of the rectangular array will be placed at the *x*-insert, *y*-insert point listed in the master section. It will then determine the *x* and *y* offset needed to place number-of-items between *x*-insert, *y*-insert and *x*-endpoint and *y*-endpoint.

Child Drawings

Another way to use master drawings:

It is possible to “embed” one or more master drawings into a single master drawing. With OfficeCAD, blocks within a drawing can be treated as separate master drawings. This method allows the source drawings of a design to be grouped together in a single drawing (handy for archiving and file transport). Use the usual AutoCAD or Vdraft methods to insert the “child” drawings as blocks into the parent drawing. To utilize the child drawings, in the Master Section of the design, enter the master drawing name as the name of the parent drawing plus “|” and the name of the child drawing.

C:\pathname\parent|child.dwg

When the design is run, the “child” block will be treated as if it were a separate master drawing.

Exporting Files

The SPC file can be used to specify other types of file output. A DWG file is always created, but additional files may be created also. The name of the file created is the output_DWG_name from the master section, but with the file extension specified.

Valid file types are:

DXF	AutoCAD drawing exchange format
BMP	Windows bitmap
JPG	JPEG - JFIF compliant
PCX	ZSoft Paintbrush
PCT	Mac Pict
PNG	Portable Network Graphics
TGA	Truevision Targa
TIF	TIFF using JPEG

A DXF file does not require additional info. All other types can have height, width and bits-per-pixel specified. When they are not present, default values are used.

variable = [FILE_EXPORT]
value = any value, zero is used in these examples
text = file type OR filetype,height,width,bits-per-pixel

Valid bits-per-pixel values:

1	Monochrome
2	4-Color
4	16-Color
8	256-Color
16	HiColor
24	TrueColor
32	TrueColor

Examples:

```
[FILE_EXPORT]
0
dxf
```

```
[FILE_EXPORT]
0
jpg,640,480,24
```

Chapter 11 – Complete OfficeCAD Function Listing

This section describes the OfficeCAD functions available in Excel. In cases where there is an abbreviated function name, examples may be found at the full function name entry.

Complete OfficeCAD Function Listing

CM_INCH(c)
Converts a number from “cm” to “inch” unit.

Example:
CM_INCH(5.0) returns 1.968503937

FEET_METER(f)
Computes a number in feet to metric meters.

Example:
FEET_METER(5.0) returns 1.524

INCH_CM(i)
Converts a number in “inch” to “cm” unit.

Example:
INCH_CM(5.0) returns 12.7

METER_FEET(m)
Converts a number from meters to feet.

Example:
METER_FEET(5.0) returns 16.40419947

KM_MILE(k)
Converts a number from kilometers to miles.

Note: Mile referred to in U.S. miles.

Example:
KM_MILE(5.0) returns 3.106855961

MILE_KM(m)
Converts a number from miles to kilometers.

Note: Mile referred to in U.S. miles.

Example:
MILE_KM(5.0) returns 8.04672

DELAY(thousands_of_a_seconds)
Waits for the specified time in 1/1000 seconds before continuing execution.

Example:

Variable	Measurement	Text
D1SEC	DELAY(1000)	

Delays design execution by 1 second.

THE_DATE$()
Returns a current date and year from reading the computer clock.

Example:
THE_DATE$() returns “Jan 08, 1991”

THE_TIME$()
Returns the current time in 12 hour format with the appropriate AM/PM indicator.

Example:
THE_TIME$() returns “8:16 PM”

PROMPT(prompt$)
Prompts for a number.

Example:

Variable	Measurement	Text
HEIGHT	PROMPT(" What is the height of door ?")	

Prompts for a numeric value of HEIGHT.

PROMPTD(prompt$, default)
Prompts for a number with a default value.

Example:

Variable	Measurement	Text
HEIGHT	PROMPTD(“ What is the height of the door ?”, 6.5)	

Prompts for the height of the door. If [RETURN] is selected, 6.5 will be assigned as the height of door.

PROMPTD_CHOICE$(prompt$, choice$, var_name$, default$)
Prompts for text from a valid list of selections. If the response is [RETURN] then it will use the default$. It will also assign the response to var_name$.

Example:

Variable	Measurement	Text
PRINT$("A. Steel")		
PRINT$("B. Wood")		
PRINT$("C. Aluminum")		
PRINT$("")		
MATERIAL		PROMPTD_CHOICE$(" Select a material ?", "steel, wood, plastic", "MATERIAL$", "wood")

It will prompt for “steel”, “wood” or “plastic”; if the response is [RETURN] then it will use “wood” as the default.

PROMPTD_STR$(prompt$, default$, var_name$)
Prompts for a text string with default$ then assigns results to var_name$.

Example:

Variable	Measurement	Text
MATERIAL		PROMPTD_STR$("Material to be used ?", "wood", "MATERIAL$")

It will prompt for the type of material. If [RETURN] is selected, then “wood” will be assigned to MATERIAL$.

PROMPTM_NUM$(val_name_list$)
Prompts for numbers to be assigned to the val_name_list$.

The name of the variable that appears in val_name_list$ must exist in the design in which this function was used. Each variable name in the val_name_list$ string must be separated by a comma.

Example:

Variable	Measurement	Text
PROMPTM_NUM$("LENGTH,WIDTH,DEPTH,RADIUS")		
LENGTH		
WIDTH		
DEPTH		
RADIUS		

This function is the same as prompting for four numbers as in the measurement column below.

LENGTH	PROMPT("==> What is the value of LENGTH ?: ")
WIDTH	PROMPT("==> What is the value of WIDTH ?: ")
DEPTH	PROMPT("==> What is the value of DEPTH ?: ")
RADIUS	PROMPT("==> What is the value of RADIUS ?: ")

PROMPTM_STR$(text_name_list$)
Prompts for text strings to be assigned to the text_name_list$.

The name of the variable that appears in text_name_list$ must exist in the design in which this function was used. Each variable name in the text_name_list$ string must be separated by a comma.

Example:

Variable	Measurement	Text
PROMPTM_STR$(“NAME$,JOB_NO$,ID$,MODEL$”)		
NAME		
JOB_NO		
ID		
MODEL		

This function is the same as prompting for four text strings, as in the text column below.

NAME		PROMPTSTR$("==> What is the NAME ?: ")
JOB_NO		PROMPTSTR$("==> What is the JOB_NO ?: ")
ID		PROMPTSTR$("==> What is the ID ?: ")
MODEL		PROMPTSTR$("==> What is the MODEL ?: ")

PROMPTR(prompt$, min, max)
Prompts for a number that is within the range of min and max numbers.

Note: There is an optional default value that can be specified, PROMPTR(prompt$, min, max, Optional default).

If default is specified it will be the choice given when the prompt is displayed.

Example:

Variable	Measurement	Text
HEIGHT	PROMPTR(" What is the height of door ?", 5.5, 7.5)	

Prompts for the height of the door, if the response is out of range (not between 5.5 to 7.5) , an error message will be displayed, and it will re-prompt for a valid response.

PROMPTRD_STR$(prompt$, var_name$, min, max, default, n_decimal)
Prompts for a text string within the range of min and max numbers. If [RETURN] is selected, it will use default value and then assign results to var_name$ with n_decimal.

This function is similar to PROMPTRD only it works with string values and allows control of the number of decimal places to be displayed.

Example:

Variable	Measurement	Text
L		PROMPTRD_STR$("Length of pipe ?", "LENGTH$",30.0, 120.0, 60, 3)
LENGTH		

It will prompt for the length of pipe within the range 30.0 and 120.0, with a default value of 60.0. If the response is “40”, then the number “40.000” will be assigned to LENGTH$.

PROMPT_ARCHSTR$(prompt$)
Prompts for a text string in architectural units.

If the response is not a valid architectural unit of the form [<feet>'-<inches> <fraction>"], then the response is rejected and an error message printed. The function will re-prompt the user until a valid response is obtained.

Example:

Variable	Measurement	Text
HEIGHT	VAL(HEIGHT$)	PROMPT_ARCHSTR$(" What is the height ?")

PROMPTR_ARCHSTR$(prompt$, min$, max$)
Prompts for text an architectural units with min$ and max$ range.

Note: If the min or max is an inch value and the inch mark is being used it must be entered double quoted.
Example, for input between 10 inches and 14 inches:
PROMPTR_ARCHSTR$("What is the length?", "10""", "14""")

The response must be a valid architectural unit of the form
[<feet>'-<inches> <fraction>"] within min$ and max$ range.

It will re-prompt the user until a valid response is obtained.

Example:

Variable	Measurement	Text
LENGTH	VAL(LENGTH$)	PROMPTR_ARCHSTR$(" What is the LENGTH ?", "10'","20'")

PROMPTSTR$(prompt$)
Prompts for a text string.

Example:

Variable	Measurement	Text
PART_NAME		PROMPTSTR$(" What is the name of part ?")

Prompts text value of PART_NAME$.

PROMPT_ARCH(prompt$)
Prompts for an architectural unit, then converts to a decimal number.

Result of conversion is dependent on the design configuration whether feet other units are used.

Example:

Variable	Measurement	Text
H		PROMPT_ARCH(" What is the height of door ?")

If the response is 6'-3" and design unit is in feet then the result will be 6.25. If the design unit is in other units then the result will be 75.00.

PROMPT_CHOICE(prompt$, choice$)
Prompts for one of the valid responses from the list of selections. It returns a number based on the selection.

The first choice in the list returns 1, second returns 2, and so on.

This function only accepts (as valid responses) the choices given in the choice$ list.

Note: Each choice in the choice$ string must be separated by a comma only.

Example:

Variable	Measurement	Text
MAT_NO	PROMPT_CHOICE(" Select material ?", "wood, plastic")	

It will prompt for either “wood” or “plastic” for the material to be used. If “wood” is entered as the response then it will return value 1.0, if “plastic” is entered then 2.0 will be returned.

PROMPTSTR_CHOICE(prompt$, choice$)
Prompts for a text string from a list of list of valid selections. Each choice in the choice$ string must be separated by a comma. Any spaces may be taken as a literal part of the choices.

Example:

Variable	Measurement	Text
MATERIAL		PROMPTSTR_CHOICE(" Select one:" & CHR(10) & "A. Steel" & CHR(10) & "B. Wood" & CHR(10) & "C. Aluminum", "A,B,C")

It will prompt for a valid list of selections from the choices “A”, “B”, or “C”. If the response is invalid, it will display a warning message, then re-prompt for a valid selection.

CHR(10) is the ASCII code for a line feed. It will cause the prompt$ to show up as a list in the prompt dialog.

PROMPT_DMS(prompt$)
Prompts for text in the form of degrees, minutes, and seconds then converts to a decimal.

Response must be separated by a blank space between numbers and must be valid degree-minutes-second format.

Example:

Variable	Measurement	Text
ANG		PROMPT_DMS(" What is the angle ?")

It will prompt for the angle; if the response is 45 15' 30", then the result will be 45.25833.

PROMPT_TOL(prompt$, text_name$, n_decimal, tol_val)
Prompts for a number, then converts to text that is in Vdraft’s special plus-minus (tolerance symbol) text format (i.e. 3.558%%P.005).

It assigns the result text string to text_name$ and also returns response in a decimal number.

Example:

Variable	Measurement	Text
LENGTH	PROMPT_TOL("Metal bar length: ", "LENGTH$", 3, 0.005)	

It will prompt for Metal bar length, if the response is 4.558 then it will assign “4.558%%p.005” to LENGTH$ and 4.558 to LENGTH.

TABLE$(file$, key$, col)
Performs table look-up function by searching text file (file$) for the row that starts with the string key$. If this row is found, the item in the column specified by col is returned.

Parameters:
file$ ....the table file to be searched
key$ ....the keyword of the line being searched for
col ....the column of line to be returned

Example:

Variable	Measurement	Text
PART_NO		PROMPTSTR$(" Part number?")
LENGTH1	VAL(LENGTH1$)	TABLE$("part2.tab",PART_NO$, 1)
WIDTH	VAL(WIDTH$)	TABLE$("part2.tab",PART_NO$, 2)
DIA	VAL(DIA$)	TABLE$("part2.tab",PART_NO$, 3)

Table file “PART2.TAB” looks like:

PART#	L	WD	DIA	<line 1>
12-21	4.0	2.0	0.25	<line 2>
12-34	2.0	3.5	0.75	<line 3>
12-55	3.0	2.5	0.50	<line 4>
<Key	Col1	Col2	Col3>	

It will prompt PART_NO then look-up LENGTH1, WIDTH and DIA values from table file “PART2.TAB”. The table file uses “STANDARD” (three or more spaces) as the delimiter. Assuming PART_NO$ = “12-55”, then the values for LENGTH1, WIDTH and DIA are 3.0, 2.5, and 0.50.

TABLE_D$(file$, key$, col, delimiter$)
Performs table look-up function by searching text file (file$) for the row that starts with the string key$. If this row is found, the item in the column specified by col is returned.

Note: Valid delimiters are any single alphabetic character, single punctuation characters such as comma or semi-colon. Tabs can be used by using the ASCII character code, Chr(9). The standard delimiter of 3 or more spaces is used by entering the reserved word “STANDARD”. Some examples are:

Chr(9)	ASCII code for a Tab
“,”	comma, or any other single character
“;”	semi-colon, or any other single character
“a”	any single character
“STANDARD”	3 or more spaces

Parameters:

file$	....the table file to be searched
key$	....the keyword of the line being searched for
col	....the column of line to be returned
delimiter$	....the column delimiter used in the file

Example:

Variable	Measurement	Text
PART_NO		PROMPTSTR$(" Part number?")
LENGTH1	VAL(LENGTH1$)	TABLE_D$("part1.tab",PART_NO$, 1,";")
WIDTH	VAL(WIDTH$)	TABLE_D$("part1.tab",PART_NO$, 2,";")
DIA	VAL(DIA$)	TABLE_D$("part1.tab",PART_NO$, 3,";")

Table file “PART1.TAB” looks like:

```
PART#;L;W;D              <line 1>
12-21;4.0;2.0;0.25       <line 2>
12-34;2.0;3.5;0.75       <line 3>
12-55;3.0;2.5;0.50       <line 4>
<Key;Col1;Col2;Col3>
```

It will prompt PART_NO then look up LENGTH1, WIDTH and DIA values from table file “PART1.TAB”. The table file uses “;” as the delimiter. Assuming PART_NO$ = “12-55”, then the values for LENGTH1, WIDTH and DIA are 3.0, 2.5, and 0.50.

FILE_DELETE$(file$)
Deletes the file$ if the file exists.

Note: NO WARNING is given before deleting the file(s).

Note: Wildcards can be used. Returns “True” if successful, “False” if not.

Example:

Variable	Measurement	Text
DEL_TAB		FILE_DELETE$("part3.tab")

Deletes “PART3.TAB” if exists.

FILE_FIND_STRING(file$, key$, col, delimiter$)
Locates line specified by the keyword key$ in a particular column in a file$.

Note: Returns line number if found, otherwise returns 0.

The col is located differently from all TABLE functions, it includes the first column as column one rather than treating it as a “key” column.

Parameters:

file$	....the file to be searched
key$	....the keyword to be matched
col	....the column to be checked
delimiter$	....delimiter between columns

Example:

Variable	Measurement	Text
FIND_S1	FILE_FIND_STRING("part3.tab", "0.85", 4, ",")	
FIND_S2	FILE_FIND_STRING("part3.tab", "0.75", 4, ",")	
FIND_S3	FILE_FIND_STRING("part3.tab", "0.50", 4, ",")	

Table file “PART3.TAB” looks like (COMMA delimited):

PART#,L,W,D	<line 1>
12-21,4.0,2.0,0.25	<line 2>
12-34,2.0,3.5,0.75	<line 3>
12-55,3.0,2.5,0.50	<line 4>
12-57,4.0,2.5,0.50	<line 5>

It will return 0, 3 and 4 to FIND_S1, FIND_S2 and FIND_C3.

FILE_LENGTH(file$)
Returns the total number of lines (records) in a file.

Example:

Variable	Measurement	Text
NO_ROWS	FILE_LENGTH("part3.tab")	

Table file “PART3.TAB” looks like (COMMA delimited):

PART#,L,W,D	<line 1>
12-21,4.0,2.0,0.25	<line 2>
12-34,2.0,3.5,0.75	<line 3>
12-55,3.0,2.5,0.50	<line 4>
12-57,4.0,2.5,0.50	<line 5>

It will return 5 to NO_ROWS.

FILE_REWRITE$(file$)
Creates a new file or erases the contents of existing file.

Notes:
Returns “1” if successful, “0” if not.

Use FILE_REREAD$ to reset to the beginning of the file before starts reading writing a file.

The FILE_READ$ and FILE_WRITE$ command after this command will read write from to the first line of the file.

Example:

Variable	Measurement	Text
FILE_REWRITE$("part3.tab")		
FILE_REREAD$("part3.tab")		
PART_NO		PROMPTSTR$(" Enter part#:")
NAME		PROMPTSTR$(" Name:")
NEW_PART		FILE_WRITE$("part3.tab",PART_NO$+" "+NAME$)

Creates the file "PART3.TAB" then resets to the beginning of the file, prompts user for PART_NO$ and NAME$ then writes them to the "PART3.TAB" file.

FILE_WRITE$(file$, text$)
Writes a line of text$ to the file file$.

Example:

Variable	Measurement	Text
PART_NO		PROMPTSTR$(" Enter part#:")
NAME		PROMPTSTR$(" Name:")
NEW_PART		FILE_WRITE$("part3.tab",PART_NO$+" "+NAME$)

Prompts user for PART_NO$ and NAME$ then writes them to "PART3.TAB".

SP_MULTI_CALC$(sheet$, n_call)
Calculates a subordinate design n_call times.

Example:

Variable	Measurement	Text
NO_ORDER	PROMPT(" TOTAL NUMBER OF ORDER ? ")	
SP_MULTI_CALC$("box", NO_ORDER)		

The "box" design looks like:

Variable	Measurement	Text
PRINT$(" ——— NEW ORDER ——")		
WIDTH	PROMPT(" Width ?")	
HEIGHT	PROMPT(" Height ?")	
DEPTH	PROMPT(" Depth ?")	
CUSTOMER		PROMPTSTR$(" Name of Customer ?")

Main design calculates subordinate design "box" NO_ORDER number of times. Each time "box" design is calculated it will request WIDTH, HEIGHT, DEPTH and CUSTOMER information.

SP_MULTI_CALC_WHILE$(sheet$, counter$, counter, n_call)
Calculates a subordinate design while counter is less than or equal to n_call.

Note: The subordinate design "sheet$" must have variable "counter$".

Example:

Variable	Measurement	Text
COUNTER	PROMPT(" Starting number ?")	
LAST	PROMPTR(" Ending number ? ",COUNTER, 100)	
SP_MULTI_CALC_WHILE$("box","COUNTER",COUNTER, LAST)		

The "box" design looks like:

Variable	Measurement	Text
COUNTER	1	

PRINT$(" ——— ORDER NUMBER: "+ STR$(COUNTER,0)+" ——")

WIDTH	PROMPT(" Width ?")	
HEIGHT	PROMPT(" Height ?")	
DEPTH	PROMPT(" Depth ")	
CUSTOMER		PROMPTSTR$(" Name of Customer ?")

Main design calculates subordinate design “box” while value of COUNTER is less than equal to the value of LAST. Each time “box” design is calculated it will request WIDTH, HEIGHT, DEPTH and CUSTOMER information.

SP_CALC$(sheet$)
Calculates all measurement and text column shells in the subordinate design.

Note: No output from the sheets is written to the .SPC file.

If next level subordinate design(s) are being called by subordinate design (sheet$) then they are also calculated.

Example:

Variable	Measurement	Text

SP_CALC$("box")

If the “box” design looks like:

Variable	Measurement	Text

IF (WIDTH = LENGTH, SP_CALC$("cube"), SP_CALC$("rectangl"))

Computes all cells in “box” design, if WIDTH is equal to LENGTH, then “cube” design is computed, otherwise “rectangl” design is computed.

SP_CALC_OUTPUT$(sheet$, outfile$)
Calculates all measurement and text column shells in the subordinate design (sheet$) then creates an output drawing specification file (*.SPC file) “outfile$”.

Note: If next level subordinate design(s) are being called by subordinate design (sheet$) then they are also calculated.

Example:

Variable	Measurement	Text
NEWBOX		PROMPTSTR$(" What is the name of the box?")

SP_CALC_OUTPUT$("BOX", newbox$)

Calculate all cells of “box” design then creates an output drawing specification file, the name of file is obtained by the PROMPTSTR$ command.

SP_CLEAR$(sheet$)
Restores the values in subordinate design (sheet$) to values before EXPORT function was called.

Note: If next level subordinate design(s) are being called by subordinate designs then their values are also restored.

Example:

Variable	Measurement	Text
SP_CLEAR$("box")		

If the “box” design looks like:

Variable	Measurement	Text
WIDTH	PROMPT(" Enter width: ")	
LENGTH	PROMPT(" Enter length: ")	
IF (WIDTH = LENGTH, SP_CLEAR$("cube"), SP_CLEAR$("rectangl"))		
IF (WIDTH = LENGTH, SP_CALC$("cube"), SP_CALC$("rectangl"))		

Restores all the values in subordinate design “box”, if WIDTH equal to LENGTH, then “cube” design is restored, otherwise “rectangl” design is restored to the values before EXPORT function called. It then calculates the design that has been cleared.

SP_EXPORT$(sheet$, sval_name$, value)
Exports a number to the measurement column of sval_name$ in subordinate design (sheet$) from the current design.

Example:

Variable	Measurement	Text
SIZE_W	PROMPT(" Enter size of width: ")	
SP_EXPORT$("box", "WIDTH", SIZE_W)		

Exports SIZE_W to the “box” design’s measurement column variable named “WIDTH”. This command replaces the WIDTH value in the “box” design by current design’s SIZE_W measurement value.

SP_EXPORTSTR$(sheet$, stext_name$, text$)
Exports a text value (text$) to the text column of stext_name$ in subordinate design (sheet$) from the current design.

Note: Abbreviated Function Name: “SP_XT$”

Example:

Variable	Measurement	Text
DOOR_W	PROMPT(" Width ?")	ARCHI$(DOOR_W)
SP_EXPORTSTR$("house", "WIDTH1$", DOOR_W$)		

Export DOOR_W$ to the text column of variable named “WIDTH1$” in design “house”.

SP_INSERT$(sheet$, x_ins, y_ins, rotation)
Exports (Inserts) the current design’s Master Region information to subordinate design (sheet$).

Note: Values of x_ins, y_ins and rotation will be automatically computed from relative position and rotation angle to absolute position in the Vdraft coordinate system.

This function is the same as executing the following commands:
SP_EXPORTSTR$(sheet$, "OUTFILE$", OUTFILE$)
SP_EXPORT$(sheet$, "SCALE", SCALE)
SP_EXPORT$(sheet$, "XINSERT", <absolute coordinates of x_ins>)
SP_EXPORT$(sheet$, "YINSERT", <absolute coordinates of y_ins>)
SP_EXPORT$(sheet$, "ROTATE",<absolute rotation>)

Example:
SP_INSERT$("box", 0.0, 10.0, 45)

Exports the current design's Master Region values (OUTFILE#, SCALE#) to subordinate design "box" and set XINSERT# = 0.0, YINSERT# = 10.0 ROTATE# = 45.0 that are relative to the current design's base point.

SP_INSERT_ABS$(sheet$, outfile$, x_ins, y_ins, scale, rotation)
Exports (Inserts) the current design's Master Region information to subordinate design (sheet$) in the Vdraft coordinate system.

Note: Values of x_ins, y_ins and rotation are absolute position in the Vdraft coordinate system.

This function is the same as executing the following commands:
SP_EXPORTSTR$(sheet$, "OUTFILE$", outfile$)
SP_EXPORT$(sheet$, "SCALE#", scale)
SP_EXPORT$(sheet$, "XINSERT#", x_ins)
SP_EXPORT$(sheet$, "YINSERT#", y_ins)
SP_EXPORT$(sheet$, "ROTATE#", rotation)

Example:
SP_INSERT_ABS$("box","longbox", 0.0, 10.0, 3.0, 45)

Exports the current design's Master Region values to subordinate design "box" with OUTFILE$ = "longbox", XINSERT# = 0,0, YINSERT# = 10.0, SCALE = 3.0 and ROTATE# = 45.0. The insertion location and angle of rotation is in the Vdraft coordinate system.

SP_INSERT_CALC$(sheet$, x_ins, y_ins, rotation)
Exports the current design's Master Region drawing information to subordinate designs, then computes all the cells of subordinate designs.

Note: This function is the same as executing the following commands:
SP_INSERT$(sheet$, x_ins, y_ins, rotation)
SP_CALC$(sheet$)

Example:
SP_INSERT_CALC$("box", 0.0, 10.0, 45.0)

Exports the current design's Master Region values (OUTFILE#, SCALE#) to subordinate design "box" and set XINSERT# = 0,0, YINSERT# = 10.0 and ROTATE# = 45.0 that are relative to the current design base point, then it will calculate all cells in the design "box".

SP_INSERT_CALC_OUTPUT$(sheet$, x_ins, y_ins, rotation)
Exports (Inserts) the current design's Master Region information to subordinate designs, computes Subordinate design then creates output drawing specification file (*.SPC file) using OUTFILE$ in the current design.

Note: This function is the same as executing the following commands:
SP_INSERT$(sheet$, x_ins, y_ins, rotation)
SP_CALC$(sheet$)
SP_OUTPUT$(sheet$)

Example:
SP_INSERT_CALC_OUTPUT$("box", 0.0, 10.0, 45.0)

Calculates all cells of "box" design then creates an output drawing specification file (*.SPC file) using current design OUTFILE$. The drawing is at location (0, 10), offset from current design base point with rotation angle of 45 degrees.

SP_IMPORT(sheet$, sval_name$)
Imports a numeric value from subordinate design's variable named sval_name$ to the current measurement column cell.

Example:

Variable	Measurement	Text
HEIGHT	PROMPT("Enter the height.")	
AREA1	SP_IMPORT("box", "WIDTH") * HEIGHT	

Compute value of AREA by importing the measurement value of “WIDTH” from subordinate design (“box”) then multiply it with HEIGHT.

SP_IMPORTNUM$(sheet$, sval_name$, val_name$)
Imports a numeric value from subordinate’s variable named sval_name$ to the current design’s variable named val_name$.

Example:

Variable	Measurement	Text
SP_IMPORTNUM$("box", "WIDTH", "BOX_SIZE_W")		
BOX_SIZE_W	4.0	

The value of BOX_SIZE_W will be replaced by the measurement value of WIDTH in the subordinate design (“box”).

SP_IMPORTSTR$(sheet$, stext_name$)
Imports a text value from subordinate’s variable named stext_name$ to the current text column cell.

Example:

Variable	Measurement	Text
SIZE1	VAL(SIZE1$)	SP_IMPORTSTR$("box", "WIDTH$")

Imports text value of WIDTH$ from design “box” to the current text cell(SIZE1$).

SP_INSERT_OUTPUT$(sheet$, x_ins, y_ins, rotation)
Exports (Inserts) the current design’s Master Region information to subordinate design then creates an output drawing specification file (*.SPC file) using OUTFILE$ in the current design.

This function is the same as executing the following commands:
SP_INSERT$(sheet$, x_ins, y_ins, rotation)
SP_OUTPUT$(sheet$)

Example:
SP_INSERT_OUTPUT$("box", 0.0, 10.0, 45.0)

Creates output drawing specification file (*.SPC file) using current design OUTFILE$. The drawing is at location (0, 10), offset from current design base point with rotation angle of 45 degrees.

SP_OUTPUT$(sheet$)
Creates an output drawing specification file (*.SPC file) of subordinate design using text value OUTFILE$ of the current design.

If OUTFILE$ MASTER$ is “none” the drawing will not be created, i.e. no output will be written to the output drawing specification file (*.SPC file).

Example:

Variable	Measurement	Text
DRAW_NAME SP_OUTPUT$(DRAW_NAME$)		PROMPT_CHOICE$("Choose:", "bolt, screw")

SHEET$	MASTER$	OUTPUT$
tool	tool	synthtmp

Creates an output drawing specification file (*.SPC file) "synthtmp" using DRAW_NAME$ design.

SP_OUTPUT_MASTER$(sheet$)
Creates an output drawing specification file (*.SPC file) of subordinate design that is identical to the last one that has been output.

Note: This function is commonly used in situations where an identical output drawing is being inserted at different locations (i.e. ARRAY functions).

Example:

Variable	Measurement	Text
SP_INSERT_CALC_OUTPUT$("window", 1, 3, 0) SP_INSERT$("window", 1, 10, 0) SP_OUTPUT_MASTER$("window")		

Insert "window" drawing at (1,3) then insert the same window drawing at (1,10) location with 0.0 rotation angle.

ARCH$(n)
Converts a number to an architectural feet/inch unit.

Example: Converted values:	Design Configuration other (inches)	 feet
ARCH$(0.500)	0 1/2"	6"
ARCH$(12.250)	1'-0 1/4"	12'- 3"
ARCH$(13.95)	1'-1 15/16"	13'-11 3/8"

ARCH0$(n)
Converts a number to an architectural feet/inch unit without inch(") mark.

Example: Converted values:	Design Configuration other (inches)	 feet
ARCH0$(0.500)	0 1/2	6
ARCH0$(12.250)	1'-0 1/4	12'-3
ARCH0$(13.95)	1'-1 15/16	13'-11 3/8"

ARCHI$(n)
Converts a number to an architectural "inch" unit.

Example: Converted values:	Design Configuration other (inches)	 feet
ARCHI$(0.500)	0 1/2"	6"
ARCHI$(12.250)	12 1/4"	147"
ARCHI$(13.95)	13 15/16"	13'-11 3/8"

ARCHI0$(n)
Converts a number to an architectural “inch” unit without inch (") mark.

Example:	Design Configuration	
Converted values:	other (inches)	feet
ARCHI0$(0.500)	0 1/2	6
ARCHI0$(12.250)	12 1/4	147
ARCHI0$(13.95)	13 15/16	167 3/8

ARCHI_NZ$(n)
Converts a number to an architectural “inch” unit without leading zero.

Example:	Design Configuration	
Converted values:	other (inches)	feet
ARCHI_NZ$(0.500)	1/2"	6"
ARCHI_NZ$(12.250)	12 1/4"	147"
ARCHi_NZ$(13.95)	13 15/16"	167 3/8"

ARCHI_NZ0$(n)
Converts a number to an architectural “inch” unit without inch (") mark and without leading zero.

Example:	Design Configuration	
Converted values:	other (inches)	feet
ARCHI_NZ0$(0.500)	1/2	6
ARCHI_NZ0$(12.250)	12 1/4	147
ARCHI_NZ0$(13.95)	13 15/16	167 3/8

ARCHVAL(text$)
Converts an architectural feet/inch unit to a number.

Example:	Design Configuration	
Converted values:	other (inches)	feet
ARCHVAL("12'-6")	150.00	12.5
ARCHVAL("12'-1/2")	147.00	12.08
ARCHVAL("1'-0 1/4")	12.25	1.02

ARCH_ID$(n)
Converts a number to an architectural feet/inch unit in the ID format (i.e. 3-5-1/4).

Note: Unit of *n* value must be in decimal inches.

Example:	Design Configuration	
Converted values:	other (inches)	feet
ARCH_ID$(41.250)	3-5-1/4	41-3
ARCH_ID$(12.250)	1-0-1/4	12-3
ARCH_ID$(0.500)	0-0-1/2	0-6

ARCH_NZ$(n)
Converts a number to an architectural “feet/inch” unit without leading zero.

Example:	Design Configuration	
Converted values:	other (inches)	feet
ARCH_NZ$(0.500)	1/2"	6"
ARCH_NZ$(12.250)	1'-1/4"	12'-3"
ARCH_NZ$(13.95)	1'-1 15/16"	13'-11 3/8"

ARCH_NZ0$(n)
Converts a number to an architectural “feet/inch” unit without inch (") mark and without leading zero.

Example: Design Configuration

Converted values:	other (inches)	feet
ARCHI_NZ0$(0.500)	1/2	6
ARCHI_NZ0$(12.250)	1'-1/4	12'-3
ARCHI_NZ0$(13.95)	1'-1 15/16	13'-11 3/8

IS_ARCH(text$)
Checks whether text$ is in an architectural “feet/inch” unit format.

Note: If text$ is a valid format, it returns -1 or “-1”, otherwise it returns 0 or “0”.

Example:
IS_ARCH("12'-1/2") returns -1.
IS_ARCH("this") returns 0.

ENG$(n)
Converts a number to an engineering architectural feet/inch unit.

Example: Design Configuration

Converted values:	other (inches)	feet
ENG$(0.500)	0.50"	6.00"
ENG$(4.125)	4.13"	4'-1.50"
ENG$(12.250)	1'-0.25"	12'-3.00"

SP_AUTO_STEP$(sheet$, x_start, y_start, x_end, y_end, n_call)
Generates “stair-stepping” arrayed subordinate design output drawings by using starting and ending points.

Note: The starting (x_start, y_start), ending (x_end, y_end) locations and the number of steps (n_call) are specified.

The function computes x and y interval distances needed to space each drawing evenly.

Parameters:
sheet$ ....Name of the drawing to be arrayed
x_start ....X starting location
y_start ....Y starting location
x_end ....X ending location
y_end ....Y ending location
n_call ....Total number of drawings to be arrayed

Example:
SP_AUTO_STEP$("stair", 5, 5, 100, 200, 50)

Generates 50 sets of “stair” drawings with starting location of (5,5) ending location (100,200).

SP_CALC_CIRC_ARRAY$(sheet$, x_cent, y_cent, radius, ang_start, ang_inc, n_call)
Calculates and generates circularly arrayed subordinate design output drawings.

Note: This function places total of n_call subordinate drawings around the center location (x_cent, y_cent) with specified radius and array from ang_start to (ang_start + ang_inc).It also calculates subordinate design each time it is called.

Use this function to do composite array of drawings.

The first object will start at ang_start. The last object will be at angle of (ang_start + ang_inc*n_call).

Parameters:
sheet$ ....Name of the drawing to be arrayed
x_cent ....X center of radius location
y_cent ....Y center of radius location
radius ....Radius of circular array
ang_start ....Starting angle of array
ang_inc ....Total number of degrees to be arrayed
n_call ....Total number of drawing to be arrayed

Example:
SP_CALC_CIRC_ARRAY$("bolthole", 5, 5, 7.5, 45, 360, 12)

The “bolthole” design looks like:

Variable	Measurement	Text
BOLT_SIZE	PROMPT(" Bolt size ? ")	

Generates 12 “bolthole” drawings at every 20 degrees (360/12) with center location of (5,5), radius of 7.5 and starting angle of 45 degrees. The subordinate design “bolthole” will prompt the user for the size of BOLT_SIZE each time it is arrayed.

SP_CALC_RECT_ARRAY$(sheet$, x_start, y_start, x_inc, y_inc, column, row)
Calculates and generates a rectangular array of subordinate design output drawings.

Note: This function places the subordinate drawing in (column x row) array with starting location of (x_start, y_start).

The x and y distance increments between each drawing are (x_inc and y_inc).

It calculates subordinate designs each time the drawing is arrayed.

Use this function to do composite arrayed drawings.

Parameters:
sheet$ ....Name of the drawing to be arrayed
x_start ....X starting location
y_start ....Y starting location
x_inc ....X increment distance
y_inc ....Y increment distance
column ....Total number of columns to be arrayed
row ....Total number of rows to be arrayed

Example:
SP_CALC_RECT_ARRAY$("bolthole", 5, 5, 10, 15, 9, 12)

The “bolthole” design looks like:

Variable	Measurement	Text
BOLT_SIZE	PROMPT(" Bolt size ? ")	

Generates 9 columns by 12 rows “bolthole” drawings with starting location of (5,5), x increment distance is 10, y increment distance is 15. It will prompt the value of BOLT_SIZE each time the drawing is arrayed.

SP_CIRC_ARRAY$(sheet$, x_cent, y_cent, radius, ang_start, ang_inc, n_call)
Generates circularly arrayed subordinate design output drawings.

Note: Subordinate designs are calculated once before the array is created.

Note: It places total of n_call subordinate drawings around the center location (x_cent, y_cent) with specified radius and arrays from ang_start to (ang_start+ang_inc).

The first object will start at ang_start. The last object willbe at angle of (ang_start + ang_inc*n_call).

Parameters:
sheet$ ....Name of the drawing to be arrayed
x_cent ....X center of radius location
y_cent ....Y center of radius location
radius ....Radius of circular array
ang_start ....Starting angle of array
ang_inc ....Total number of degrees to be arrayed
n_call ....Total number of drawings to be arrayed

Example:
SP_CIRC_ARRAY$("bolthole", 5, 5, 7.5, 45, 360, 12)

Generates 12 “bolthole” drawings at every 20 degrees (360/12) with center location of (5,5), radius of 7.5 and starting angle of 45 degrees.

SP_RECT_ARRAY$(sheet$, x_start, y_start, x_inc, y_inc, column, row)
Generates a rectangular array of subordinate design output drawings.

Note: Subordinate designs are calculated once before the array is created.

Note: This function places the subordinate drawing in (column x row)array with starting location of (x_start, y_start).

The x and y distance increments between each drawing are (x_inc and y_inc).

Parameters:
sheet$ ....Name of the drawing to be arrayed
x_start ....X starting location
y_start ....Y starting location
x_inc ....X increment distance
y_inc ....Y increment distance
column ....Total number of columns to be arrayed
row ....Total number of rows to be arrayed

Example:
SP_RECT_ARRAY$("bolthole", 5, 5, 10, 15, 9, 12)

Generates 9 columns by 12 rows “bolthole” drawings with starting location of (5,5), x increment distance is 10, y increment distance is 15.

SP_SINE_ARRAY$(sheet$, x_start, y_start, x_inc, cycles, height, width)
Generates a sine wave pattern of subordinate design output drawings.

Note: Subordinate designs are calculated once before the array is created.

Parameters:
sheet$ ....Name of design to be arrayed
x_start ....X position of starting point
y_start ....Y position of starting point
x_inc ....X increment
cycles ....Number of cycles for the sine wave
height ....Height of sine wave (peak to trough)
width ....Width of resulting figure

Example:
SP_SINE_ARRAY$("bolt", 0, 0, 0.25, 1, 5, 10)

Places one cycle (0 to 360 degrees) “bolt” output drawings at every x increment of 0.25", starting at (0,0), height of 5" and width of 10".

SP_STEP_ARRAY$(sheet$, x_start, y_start, x_inc, y_inc, n_call)
Generates a stair-stepped array of subordinate design output drawings using specified X,Y increment distances.

Note: The starting (x_start, y_start) increment distances (x_inc,y_inc) and the number of steps (n_call) are specified.

Parameters:
sheet$ ....Name of the drawing to be arrayed
x_start ....X starting location
y_start ....Y starting location
x_inc ....X increment distance
y_inc ....Y increment distance
n_call ....Total number of drawings to be arrayed

Example:
SP_STEP_ARRAY$("stair", 5, 5, 50, 10, 20)

Generates 20 sets of “stair” drawings with starting location of (5,5) and x increment distance of 50, y increment distance 10.

IF(condition, true_response, false_response)
Evaluates logical statement (condition); if it is true then returns the value true_response, else it returns the value of false_response.

Note: The results of this function are numeric, so use this IF function in the measurement column.

Example:

Variable	Measurement	Text
TEST	IF(LENGTH > 20, 2.0*A, 0.5*B)	

The value of TEST will be 2.0*A if LENGTH is greater than 20, else it will be 0.5*B.

CONSTANTS()

Predefined constants.
PI = 3.14159265358979323846264333;
TRUE = 1;
FALSE = 0;
LOG_E = 2.718281828459045;

PI()
Returns the value of pi.

Note: The value of pi: 3.14159265358979323846264333

Example:

Variable	Measurement	Text
R	PROMPT("Radius ?")	
AREA	PI()*R*R	

The value of AREA will be pi*R*R.

ABS(n)
Returns the absolute of value *n*.

Example:

Variable	Measurement	Text
A1	ABS(B)	

The value of A1 will be the absolute value of B.

ACOS(n)
Returns the arc cosine of value *n*.

Note: Degrees, or radians, depending on design configuration.

Example:

Variable	Measurement	Text
A1	ACOS(B)	

The value of A1 will be the arc cosine of B.

ASIN(n)
Returns the arc sine of value *n*.

Note: Degrees, or radians, depending on design configuration.

Example:

Variable	Measurement	Text
A1	ASIN(B)	

The value of A1 will be the arc cosine of B.

ATAN(n)
Returns the arc tangent of value *n*.

Note: Degrees, or radians, depending on design configuration.

Example:

Variable	Measurement	Text
A1	ATAN(B)	

The value of A1 will be the arc tangent of B.

ATAN2(n1, n2)
Returns the arc tangent of *n1*/*n2*.

Note: Degrees, or radians, depending on design configuration.

Example:

Variable	Measurement	Text
A1	ATAN2(B1, B2)	

The value of A1 will be the arc tangent of B1/B2.

COS(n)
Returns the cosine of value *n*.

Note: Degrees, or radians, depending on design configuration.

Example:

Variable	Measurement	Text
A1	COS(B)	

The value of A1 will be the cosine of B.

COSH(n)
Returns the hyperbolic cosine of value *n*.

Note: Degrees, or radians, depending on design configuration.

Example:

Variable	Measurement	Text
A1	COSH(B)	

The value of A1 will be the hyperbolic cosine of B.

EXP(n)
Returns the value e (approximately 2.718282) raised to the power of *n*.

Example:

Variable	Measurement	Text
A1	EXP(7)	

The value of A1 will be e^7.

FACT(n)
Computes the factorial of the number *n*.

Note: Factorial formula: *n* * FACT(*n* - 1)

Example:

Variable	Measurement	Text
A1	FACT(3)	

The value of A1 will be 3*2*1=6.

FIB(n)
Computes the fibonacci of the number *n*.

Note: Use this function if *n* is less than or equal to 12.

Example:

Variable	Measurement	Text
A1	FIB(3)	

The value of A1 will be the fibonacci of 3.

FRAC(n)
Returns the fractional decimal part of value *n*.

Example:

Variable	Measurement	Text
A1	FRAC(4.445)	

The value of A1 will be 0.455.

INT(n)
Returns the integer part of value *n*.

Example:

Variable	Measurement	Text
A1	INT(3.446)	

The value of A1 will be 3.

LN(n)
Returns the natural log value of *n*.

Note: The value of *n* must be greater than 0.

Example:

Variable	Measurement	Text
A1	LN(B)	

The value of A1 will be the natural log of B.

LOG(n)
Returns the log base 10 of *n*.

Note: The value of *n* must be greater than 0.

Example:

Variable	Measurement	Text
A1	LOG(B)	

The value of A1 will be the log base 10 of B.

MIN(n1, n2)
Returns the smallest of the two numbers *n1* and *n2*.

Example:

Variable	Measurement	Text
A1	MIN(2, -3.8)	

The value of A1 will be -3.8.

MAX(n1, n2)
Returns the largest of the two numbers *n1* and *n2*.

Example:

Variable	Measurement	Text
A1	MAX(2, -3.8)	

The value of A1 will be 2.

POW(n1, n2)
Returns the *n1* raised to the power of *n2*.

Example:

Variable	Measurement	Text
A1	POW(2, 3)	

The value of A1 will be 8.

RANDOMIZE(n)
Initializes the Excel RND function's random number generator, using the value of *n*.

Notes:
If *n* is not supplied, the RND function's seed value is returned by the system timer.

If the RANDOMIZE function is not used, the RND function (with no arguments) uses the same initial seed value, then uses the last generated number as a seed value.

To repeat random number sequences, use the RND function with a negative argument prior to using the RANDOMIZE function with a numeric argument.

RND(n)
Returns a less than 1 and greater than or equal to zero.

Notes: How this number is generated depends on the value of *n*.

For	Rnd returns
n < 0	The same number every time, using *n* as the seed.
n > 0 or *n* not supplied	The next random number in the sequence.
n = 0	The most recently generated number.

If the RND function is used without using the RANDOMIZE function, the same number sequence is generated. Use the RANDOMIZE function (without arguments) to initialize the random number generator based on the system timer.

The Rnd function returns a value less than 1 but greater than or equal to zero. The value of number determines how Rnd generates a random number: For any given initial seed, the same number sequence is generated because each successive call to the Rnd function uses the previous number as a seed for the next number in the sequence. Before calling Rnd, use the Randomize statement without an argument to initialize the random-number generator with a seed based on the system timer.

To produce random integers in a given range, use this formula:

Int((highnum - lownum + 1) * Rnd + lownum)

Here, highnum is the highest and lownum is the lowest number in the range.

Variable	Measurement	Text
CHOICE	INT(10 * RND + 1)	

Returns an integer between 1 and 10 as the value of CHOICE.

To repeat random number sequences, use the RND function with a negative argument prior to using the RANDOMIZE function with a numeric argument.

ROUND(n)
Returns the *n* rounded up to the closest integer.

Example:

Variable	Measurement	Text
A1	ROUND(2.775)	
A2	ROUND(0.455)	

The value of A1 and A2 will be 3 and 0.

SIN(n)
Returns the sine of value *n*.

Note: Degrees, or radians, depending on design configuration.

Example:

Variable	Measurement	Text
A1	SIN(B)	

The value of A1 will be the sine of B.

SINH(n)
Returns the hyperbolic sine of value *n*.

Note: Degrees, or radians, depending on design configuration.

Example:

Variable	Measurement	Text
A1	SINH(B)	

The value of A1 will be the hyperbolic sine of B.

SQUARE(n)
Returns the value of *n* * *n*.

Example:

Variable	Measurement	Text
A1	SQUARE(7)	

The value of A1 will be 49.

SQRT(n)
Returns the square root of *n*.

Example:

Variable	Measurement	Text
A1	SQRT(2)	

The value of A1 will be 1.414.

TAN(n)
Returns the tangent of value *n*.

Note: Degrees, or radians, depending on design configuration.

Example:

Variable	Measurement	Text
A1	TAN(B)	

The value of A1 will be tangent of B.

TANH(n)
Returns the hyperbolic tangent of value *n*.

Note: Degrees, or radians, depending on design configuration.

Example:

Variable	Measurement	Text
A1	TANH(B)	

The value of A1 will be the hyperbolic tangent of B.

STRTOK$(text$, col, delimiter$)
Returns col-th substring of text$ which has been delimited into sub-string(s) by delimiter$.

Note: If delimiter$ is “STANDARD”, the delimiters are three spaces or a tab. If there is no string in the col-th location, it will return null string (“”).

Example:

STRTOK$("PART, D, L, W", 3, ",")	returns “ L”
STRTOK$("W=30.25cm", 2,"=")	returns “30.25cm”
STRTOK$("W=30.25cm", 2,"STANDARD")	returns “”

STR_ACAD$(n, n_decimal, text$)
Formats a number to Vdraft’s special %% text string.

Note: Text$ must one of “o”, “u”, “d”, “p”, “c”, “%”, “nnn” characters.

Example:

STR_ACAD$(4.6,2, "o")	returns “4.60%%o”	....Overscore mode
STR_ACAD$(0.4588, 3, "u")	returns “0.459%%u”	....Underscore mode
STR_ACAD$(4.60,1, "d")	returns “4.6%%d”	....Degree symbol
STR_ACAD$(4.556, 3, "p")	returns “4.556%%p”	....Tolerance symbol
STR_ACAD$(2.0,0, "c")	returns “2%%c”	....Circle symbol
STR_ACAD$(2.0,2, "%")	returns “2.00%%%”	....Single % sign
STR_ACAD$(2.0, 0, "105")	returns “2%%105”	....ASCII character

STR_CREATE$(text$, n_length)
Creates a text string of n_length characters using the character (s) of text$ to fill the string.

The text will usually be one character long, thus producing a string of all the same characters. However if text is longer than one character, that text is repeated over and over until the string is of the length needed.

If the length of text$ is longer than length, then the string is truncated using the LEFT$() function.

Example:
STR_CREATE$("?-", 10) returns “?-?-?-?-?-”
STR_CREATE$("This is too long.", 4) returns “This”
STR_CREATE$("== ", 4) returns “== =”

STR_DEC_DEGREES$(degrees)
Formats a number into a string Degree-Minutes-Seconds format
<Degrees>%%d<Minutes>'<Seconds>".

Example:
STR_DEC_DEGREES$(30.355) returns (30%%d21'18")

FORMAT("string", Optional "format")
Returns “string” formatted according to the optional “format” given.

Examples of named date and time formats:

General Date

Format("1:30", "General Date")	= 1:30:00 AM
Format("1:30 AM", "General Date")	= 1:30:00 AM
Format("1:30 PM", "General Date")	= 1:30:00 PM
Format("3/21/99", "General Date")	= 03/21/99

Long Date

Format("3/21/99", "Long Date")	= Sunday, March 21, 1999

Medium Date

Format("3/21/99", "Medium Date")	= 21-Mar-99

Short Date

Format("3/21/99", "Short Date")	= 03/21/99

Long Time

Format("1:30:45", "Long Time")	= 1:30:45 AM
Format("1:30:45 AM", "Long Time")	= 1:30:45 AM
Format("1:30:45 PM", "Long Time")	= 1:30:45 PM

Medium Time

Format("1:30:45", "Medium Time")	= 1:30 AM
Format("1:30:45 AM", "Medium Time")	= 1:30 AM
Format("1:30:45 PM", "Medium Time")	= 1:30 PM

Short Time

Format("1:30:45", "Short Time")	= 01:30
Format("1:30:45 AM", "Short Time")	= 01:30
Format("1:30:45 PM", "Short Time")	= 13:30

Examples of named numeric formats:

General Number (displays a number with no thousand separator).

format(2.2, "general number")	= 2.2
format(20.25, "general number")	= 20.25
format(.75, "general number")	= 0.75
format(20256.75, "general number")	= 20256.75
format(2.2)	= 2.2
format(20.25)	= 20.25
format(.75)	= 0.75
format(20256.75)	= 20256.75

Currency (displays the number with thousand separator, if appropriate; displays two digits to the right of the decimal separator; output is based on the local system settings).

format(2.2, "currency")	= $2.20
format(20.25, "currency")	= $20.25
format(.75, "currency")	= $0.75
format(20256.75, "currency")	= $20,256.75

Fixed (displays at least one digit to the left and two digits to the right of the decimal separator).

format(2.2, "fixed")	= 2.20
format(20.25, "fixed")	= 20.25
format(.75, "fixed")	= 0.75
format(20256.75, "fixed")	= 20256.75

Standard (displays the number with the thousand separator, at least one digit to the left and two digits to the right of the decimal separator).

format(2.2, "standard")	= 2.20
format(20.25, "standard")	= 20.25
format(.75, "standard")	= 0.75
format(20256.75, "standard")	= 20,256.75

Percent (displays a number multiplied by 100 with a percent sign (%) appended to the right; always displays two digits to the right of the decimal separator).

format(.022, "percent")	= 2.20%
format(.2025, "percent")	= 20.25%
format(.75, "percent")	= 75.00%
format(.2025675, "percent")	= 20.26%

Scientific (uses standard scientific notation).

format(2.2, "scientific")	= 2.20E+00
format(20.25, "scientific")	= 2.03E+01
format(.75, "scientific")	= 7.50E-01
format(20256.75, "scientific")	= 2.03E+04

Yes/No (displays No if number is 0; otherwise, displays Yes).

format(2.2, "yes/no")	= Yes
format(20.25, "yes/no")	= Yes
format(-0.75, "yes/no")	= Yes
format(0, "yes/no")	= No

True/False (displays False if number is 0; otherwise, displays True).

format(2.2, "true/false")	= True
format(20.25, "true/false")	= True
format(-0.75, "true/false")	= True
format(0, "true/false")	= False

On/Off (displays Off if number is 0; otherwise, displays On).

format(2.2, "on/off")	= On
format(20.25, "on/off")	= On
format(-0.75, "on/off")	= On
format(0, "on/off")	= Off

Examples of user-defined formats:

Format("3/21/99", "mmm dd, yyyy")	= Mar 21, 1999
Format("3/21/99", "mm-dd-yy")	= 03-21-99
Format("1:30:45 AM", "hh:mm")	= 01:30
Format("1:30:45 PM", "hh:mm")	= 13:30

Format("2.25", "$##.00")	= $2.25
Format("20.25", "$##.00")	= $20.25
Format("0.25", "$##.00")	= $.25
Format("0.25", "$#0.00")	= $0.25

Format("this is a string", ">")	= THIS IS A STRING
Format("THIS IS A STRING", "<")	= this is a string

STR_FORMAT$(text$, n_length)
Formats a text string into a right or left justified text string and changes the text string length to n_length number of characters.

Note: If n_length is positive then the text will be right justified. If n_length is negative then the text will be left justified. Blank spaces in text$ are treated as characters.

Examples:

STR_FORMAT$("$20.00", 8)	returns " $20.00"
STR_FORMAT$("PART#", 10)	returns " PART#"
STR_FORMAT$("PART#", -10)	returns "PART# "

STR_LOWER$(text$)
Returns text$ converted to lower-case.

Note: No change in special characters.

Example:

STR_LOWER$("Bad Format")	returns "bad format"
STR_LOWER$("ABC &*%$")	returns "abc &*%$"
STR_LOWER$(PROMPTSTR$("Part#?"))	returns lower case response

STR_NOBLANK$(text$)
Deletes leading spaces from text$.

Note: If both leading and trailing spaces need to be deleted use STR_NOBLANK2$.

Example:

STR_NOBLANK$(" 12-34")	returns "12-34"
STR_NOBLANK$(" 12-34 ")	returns "12-34 "

STR_NOBLANK2$(text$)
Deletes leading and trailing spaces from text$.

Example:

STR_NOBLANK$(" 12-34 ")	returns "12-34"

STR_N(n, n_decimal)
Converts a number to a text with n_decimal decimal places.

Note: n_decimal is optional. If it is not specified the Decimal configuration for the worksheet is used.

Example:

STR_N(0.5544, 3)	returns "0.554".
STR_N(0.5547, 3)	returns "0.555".

STR_NZ$(n, n_decimal)
Converts a number to a text with n_decimal decimal places without leading zero.

Example:

STR_NZ$(0.5544, 3)	returns ".554".
STR_NZ$(0.5547, 3)	returns ".555".

STR_UPPER$(text$)
Returns the string converted to upper-case.

Note: No change in special characters.

Example:

STR_UPPER$("Bad Format")	returns "BAD FORMAT"
STR_UPPER$("abc&*%$")	returns "ABC&*%$"
STR_UPPER$(PROMPTSTR$(" Part#?"))	returns upper case response

YESNO(prompt$)

Notes:
Displays prompt$ with "Yes", "No", or "Cancel" buttons.

If response is "Yes" button then it will return -1 (if used in Measurement column),or "True" (if used in Text column).

If response is "No" button then it will return 0 (if used in the Measurement column),or "False" (if used in the Text column).

If response is "Cancel" button, the calculation is halted.

Example:

Variable	Measurement	Text

IF (YESNO(" Correct Specification(Y/N) ?") = 1,SP_CALC$("cylinder"), SP_CALC$("cyl_data"))

It will prompt to check specifications of "cylinder" design data. If "Y" is selected, it will calculate "cylinder" design, otherwise it will execute "cyl_data" for re-entering the data.

YESNOSTR(prompt$)
Displays prompt$ with "Yes", "No", or "Cancel" buttons.

Notes: If response is "Yes" button then it will return "Y". If response is "No" button then it will return "N". If response is "Cancel" button, the calculation is halted.

Example:

Variable	Measurement	Text
OK_SPEC		IF(YESNOSTR(" Correct Specification(Y/N) ?") = "Y", SP_CALC$("cylinder"), SP_CALC$("cyl_data"))

It will prompt to check specifications of "cylinder" design data. If "Yes" is selected, it will calculate "cylinder" design, otherwise it will execute "cyl_data" for re-entering the data.

YESNOD(prompt$, default$)
Prompts for a response that can only be "Y" "N" with default value.

Notes:
Displays prompt$ with "Yes", "No", and "Cancel" buttons.
"Yes" returns -1 (if used in the Measurement column),or "True" (if used in the Text column).
"No" returns 0 (if used in the Measurement column), or "False" (if used in the Text column).
"Cancel" halts the calculation.
If default$ is "y", or "yes" (case does not matter), the default button will be "Yes". If it is anything else or omitted "No" will be the default button.

Example:

Variable	Measurement	Text
ANSWER		YESNOD("The Design values correct(Y,N) ?",1)

It will prompt to confirm response either "Y" "N", if [RETURN] is selected, the result of the function will be 1.

MESSAGE(text$)
Displays message text$ on the bottom of the design screen as an additional message to the user while running a design.

Note: Displays text$ in a messagebox dialog with an OK button. No text or numeric values are returned.

Example:

Variable	Measurement	Text
TYPE		PROMPTSTR$(" Enter Model Type: ")
SHOW	IF(TYPE$ = "flat", MESSAGE("DESIGNING FOR FLAT MODEL"), MESSAGE("DESIGNING FOR ROUND MODEL"))	

It displays different messages depending upon user input TYPE$.

WARNING(text$)
Displays warning message text$ to design screen then waits for any key input to continue execution of design.

Example:

Variable	Measurement	Text
SIZE	VAL(SIZE$)	PROMPTSTR$(" Enter size: ")
TEST_SIZE	IF(SIZE > 100.0, WARNING("Warning - Size is more than 100.",0)	

It displays warning message then waits for user input to continue execution of the design if the value of SIZE becomes greater than 100.0, otherwise assigns 0 to TEST_SIZE.

WARNING$(text$)
Displays warning message text$ at the bottom of the design screen.

Note: This function is the same as WARNING except the function is in text mode. Returns warning message.

Example:

Variable	Measurement	Text
SIZE	VAL(SIZE$)	PROMPTSTR$(" Enter size: ")
TEST_SIZE		IF(SIZE > 100.0, WARNING$("Warning - Size is more than 100.","ok")

It displays a warning message then waits for user input to continue execution of the design if the value of SIZE becomes greater than 100.0, otherwise assigns “ok” to TEST_SIZE$.

ASC(text$)
Returns ASCII value of the first character of text$.

Example:
ASC("A") returns 65.

CHR$(n)
Returns ASCII text string of 1 character where n (0 to 255) value matches its text ASCII value.

Example:
CHR$(65) returns “A”.

DELETE$(text$, start_pos, n_length)
Deletes characters specified by n_length out of a text$ string starting at start_pos (0 being the first).

Note: Returns what is left of text$ after deletion.

Example:
DELETE$("The big bad dog.", 7,4) ...returns "The big dog."
DELETE$("The big bad dog.", 0,4) ...returns "big bad dog."

INSERT$(text$, start_pos, text_insert$)
Inserts text_insert$ into text$ starting at the start_pos position.

Note: The first character in text$ is position 1.

Example:
INSERT$("The big dog.", 5, "great ") ...returns "The great big dog."
INSERT$("The big dog.", 12, " is great") ...returns "The big dog is great."

INSTR(start_pos, text$, key$)
Searches for the string key$ within text$ starting from start_pos (1 being the first). If key$ string is found in text$, it returns the location that key$ matched, otherwise returns 0.

Example:
InStr(1, "abcd", "a") = 1
InStr(6, "abcd", "b") = 0
InStr(5, "abcdefghij", "efg") = 5
InStr(6, "abcdefghij", "efg") = 0

LEFT$(text$, length)
Returns a substring of text$. The substring consists of characters from text$ starting with the left-most characters and containing length number of characters.

Example:
LEFT$("The big dog.", 7) ...returns "The big"
LEFT$("2.5000%%P.005", 6) ...returns "2.5000"
LEFT$("The big dog.", 17) ...returns "The big dog."

LEN(text$)
Returns the total number of characters in text$.

Example:
LENGTH("The big dog.") ...returns 12
LENGTH("") ...returns 0 (null string)
LENGTH("   The big dog.") ...returns 15 (blanks are counted)

MID$(text$, start_pos, length)
Returns a substring of text$. The substring consists of characters from text$ starting from start_pos to start_pos+length position.

Note: The position of the first character is 1.

Example:
MID$("The big dog.", 3, 3) ...returns "big"
MID$("2.5000%%P.005",1, 6) ...returns "2.5000"

REPLACE$(text$, start_pos, length, text_replace$)
Replaces total of length characters with text_replace$ string starting at the start_pos position in text$.

Note: The position of the first character is 1.

Example:
REPLACE$("The big dog.", 5, 3, "little") ...returns "The little dog."
REPLACE$("The big dog.", 1, 3, "This is a") ...returns "This is a big dog."

RIGHT$(text$, length)
Returns a substring of text$. The substring consists of characters from text$ starting with the right-most characters and containing length number of characters.

Example:
RIGHT$("The big dog.", 8) ...returns “big dog.”
RIGHT$("2.5000%%P.005", 7) ...returns “%%p.005”

VAL(text$)
Converts text$ to a number.

Note: If text$ starts with numeric text, leading space(s) and numeric text then the first set of numeric text will be converted to a number.

Note: If any spaces are encountered after the first set of numeric text and the first characters after the space(s) are numeric, they will be concatenated to the end of first set of numeric text.

Excel example:

VAL("12.25")	...returns 12.25.
VAL("13.00cm")	...returns 13.00
VAL(" 4'-2")	...returns 4.00
VAL(" 123 456.25")	...returns 123456.25

FEET_CFG()
Returns default design unit (feet other) configuration.

Note: If the worksheet is configured for feet, returns -1 (if used in the Measurement column), “True” (if used in the Text column), otherwise returns 0 (if used in the Measurement column), “False” (if used in the Text column).

Default is other (inches).

Example:

Variable	Measurement	Text
F_IN	FEET_CFG()	

Returns the current design’s unit configuration to F_IN.

DECIMAL_CFG()
Returns default design’s decimal configuration.

Note: Decimal configuration can be 0,1,2,3,max (11).
Can be used in either the Measurement or Text column.
Default is 3 decimals.

Example:

Variable	Measurement	Text
NO_DEC	DECIMAL_CFG()	

Returns the current design’s number of decimal configuration to NO_DEC.

DEGREE_CFG()
Returns default design degree configuration.

Note: If the worksheet is configured for degrees, returns -1 (if used in the Measurement column), “True” (if used in the Text column). Otherwise it is configured for radians and returns 0 (if used in the Measurement column), “False” (if used in the Text column).

Default is in degrees.

Example:

Variable	Measurement	Text
DEG	DEGREE_CFG()	

Returns the current design's degree configuration to DEG.

FRACTION_CFG()
Returns default design fraction denominator configuration.

Note: The fraction can be 1/2, 1/4, 1/8, 1/16, 1/32 and 1/64.
Can be used in either the Measurement or Text column.
The fraction can be any valid fraction denominator value.
Default is 1/32.

Example:

Variable	Measurement	Text
NO_FRAC	FRACTION_CFG()	

Returns the current design's fraction denominator configuration to NO_FRAC.

TSCALE
This variable TSCALE is placed in the VARIABLE column of your design. A measurement field value is given. No text field value is needed.

The TSCALE variable SCALES Up or Down the size of your entire output drawing.

Normally the SCALE factor (in the Master region) only affects the ANNOTATIVE portions of the drawing, such as dimension legs, dimension text, drawing text, arrow heads tick marks.

The TSCALE variable scales up EVERYTHING. Use it when:
A. Trying to place details on a large assembly drawing.
B. Trying to scale up a border to fit a detail contained within it. Use the border as a separate Master Drawing. The new size of the border (TSCALE) is the ratio of the new detail length or height / original detail length or height in the Master Drawing.

If TSCALE is used in any design, it must be used in all designs. Set TSCALE() = 1 for those drawings that no scaling is wanted.

Example:

Variable	Measurement	Text
TSCALE	1	

Zinsert
The ZINSERT variable allows you to place your details into 3D space at a different elevation along with XINSERT, YINSERT, ROTATE and TROTATE variables.

The ZINSERT variable operates in a similar fashion to the XINSERT and YINSERT variables. The only difference is that you must put the ZINSERT variable into the Variable column.

Place a value into the Measurement column to specify the value for the Z insertion point.

TROTATE
The TROTATE variable rotates all entities within a drawing about the 0,0 axis.

This is different than normal ROTATE variable. The ROTATE will not affect horizontal vertical dimension text. In some cases you may wish to override this normal ROTATE and use TROTATE.

Place TROTATE into the Variable column. Place a value in the Measurement column for the TROTATE value. The Text column may be left blank.

Chapter 12 – Creating Your Own VBA Function Library for Use with OfficeCAD

OfficeCAD Add Function Library

OfficeCAD has a function library called OffCAD.dll. OfficeCAD includes tools for adding functions using Visual Basic. These additional functions are contained in a function library called OffCADAd.dll.

OffCADAd.vbp is a Visual Basic project file included on the OfficeCAD CD. It contains the class module OffCADaddIn, and a prototype module MoreFuncs1.

The class module OffCADaddIn contains a function called EvalFunction. If a function is unknown to OffCAD.dll, EvalFunction is called. (Note: When a function is added, it can not have the same name as an already-defined OfficeCAD function. The existing function will be called.)

Case statements for new functions must be entered in EvalFunction in the location labeled “Your Case Statements Here.” Functions can be stored in this project or a different project. If new functions are stored in this project, OffCADAd.dll will need to be rebuilt. If functions are stored in a different project, a reference must be placed in this project, and OffCADAd.dll, as well as any other DLL files referenced, must be rebuilt.

Guidelines for Adding Functions to OfficeCAD

All routines added must be Functions. Subs will not work. To make additional functions available:

1. Write a new Function in the MoreFuncs1(xfuncs.bas) module. If you want to create new functions in another Visual Basic Project you will have to make the correct project references and settings for both projects.

2. Add a case to the Select ... Case statement in EvalFunction. EvalFunction makes additional functions available to the expression evaluater in OffCAD.dll, which calls this function when it finds an unknown function.

   Example of a one-argument function named NewFunc1:

   ```
   Case “NEWFUNC1”
   ```

   Notice that above statement contains the function name in all-caps. This is necessary.

   ```
   EvalFunction = NewFunc1(Arguments(0))
   ```

   Since NewFunc1 has one argument, Arguments should have only one element, element 0.

   Example of a three-argument function named NewFunc2:

Case “NEWFUNC2”

EvalFunction = NewFunc2(Arguments(0),Arguments(1),Arguments(2))

This time, Arguments has three elements, 0 through 2.

Example of a function with an optional argument named NewFunc3, which will have either 1 or 2 arguments:

Case “NEWFUNC3”

Select Case UBound(Arguments)

UBound(Arguments) returns the largest available subscript for the Arguments array, hence telling you how many arguments have been included. Remember that since the subscripts start at 0, Ubound will return a number one less than the number of arguments. With one argument:

Case 0

EvalFunction = NewFunc3(Arguments(0))

With two arguments:

Case 1

EvalFunction = NewFunc3(Arguments(0),Arguments(1))

End Select

3. Re-build this project (OffCADAd.DLL) and any others you may have referenced.

Prototype Module Xfuncs.bas

Xfuncs.bas is a module for functions which extend the functionality of OfficeCAD. Guidelines for additional functions:

1. All procedures must be Functions. Subs are not allowed.
2. All arguments must be passed by Value (ByVal).
3. Argument and return value types must be string or numeric, boolean or variant. Value types can not be user-defined or object types.
4. Don’t use names already defined in OfficeCAD. The already-defined functions take precedence and your function will not be called.

If you wish to store extra functions in a different project (hence, a different DLL), you will still need to add to the case statement in EvalFunction in the OffCADaddIn class module. You may need to set the references in a new project if you wish to use existing OfficeCAD functions.

Chapter 13 – Using OfficeCAD on the Internet

Creating Drawings on Your Web Server with OfficeCAD Professional

OfficeCAD provides the ability to create new drawings from the Internet via your web server. A server version of OfficeCAD is available which can create your new CAD drawings as DWG or DXF files along with a bitmap image which can be instantly displayed on your web pages.

A special high capacity server version of OfficeCAD is required to run drawings over your web site. However, a test version can be set up on a single workstation for development and debugging.

Requirements for web-based automated CAD with OfficeCAD

Server hardware and software.

Windows NT 4.0 or Windows 95/98.

OfficeCAD Server Version.

DXF / DWG to raster converter software to generate instant bitmap images from your output drawing for display on your web browser.

CGI program:

> For DXF output files, the CGI program may be written in any language as long as it can call an external DOS application.
>
> For DWG output files, the CGI program must be written in Visual Basic 5.0, 6.0 or C++ or must call another small external program written in Visual Basic 5.0, 6.0 or C++ that will call Vdraft via automation.
>
> No matter what language is used, the CGI program must be capable of writing a text file to the server's hard drive.

Vdraft Server Version (if DWG output is required in addition to DXF output drawings).

The best place to begin your web-based automated drawing is with a setup where you can run tests on your combination of web page data input, drawing generation and drawing display.

Distributing Drawings with Vdraft's Drawing Viewing Plug-in

Many plug-in programs exist for viewing drawings over the Internet. One example is Vdraft's drawing viewing plug-in. Visit the Vdraft website (http://www.vdraft.com/) for information about acquiring and using this plug-in.

CGI Scripts to Create and Deliver Drawings Over the Internet

To run OfficeCAD over the Internet, you need to create a CGI program. OfficeCAD needs to be called from either Visual Basic or C++. If the CGI program is written in Perl or another language, it will still need to have a small Visual Basic or C++ application that the CGI program calls in order to run OfficeCAD.

The CGI program is to write an SPC file (explained elsewhere in this manual) to the server, then call OfficeCAD to create the new drawing(s). The new drawings can be downloaded or displayed using a drawing viewer plug-in (such as Vdraft's, described above). The CGI program also can be used to call programs that produce a rastor image for output drawing display on your website.

Appendix A –
Using OfficeCAD with Vdraft

Vdraft Interface

This manual assumes you are using the OfficeCAD interface within AutoCAD. This appendix is to explain using the OfficeCAD interface within Vdraft.

Installing

Follow the CD instructions on how to install Vdraft (version 2.2 or later) for master drawing creation. AutoCAD does not need to be installed.

Running OfficeCAD in Vdraft

1. Be sure OfficeCAD is installed.
2. Start Vdraft.
3. In Vdraft, from the Extensions menu, select OfficeCAD. The OfficeCAD dialog box will appear. If it does not, re-install OfficeCAD.

It is possible to perform all OfficeCAD operations from inside of Vdraft. Once Vdraft and OfficeCAD are properly installed, start Vdraft. Vdraft will automatically load OfficeCAD. To access the OfficeCAD dialog boxes, select the “Extensions” pull down, then select “OfficeCAD.” The dialog boxes that are displayed using this method are nearly identical to those that would be displayed in AutoCAD. You may follow the examples in the manual, selecting Extensions, OfficeCAD from the menu to display the dialog boxes as needed. It is recommended that you spend some time becoming familiar with Vdraft if you have not used it previously.

Main Menu Items

Most of these items work identically to the AutoCAD options, even if they are located within different OfficeCAD dialog boxes.

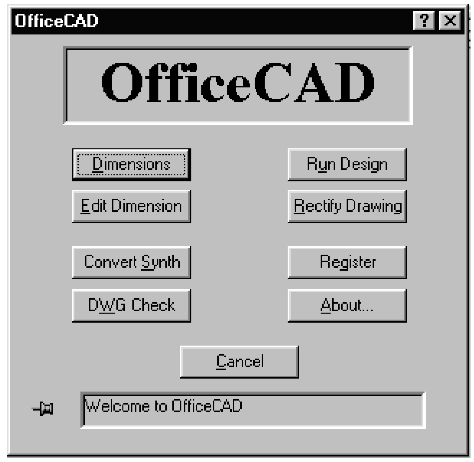


Fig. A-1 THE OFFICECAD MAIN MENU IN VDRAFT

The “SpecSheet” option is missing, and with it the sub dialog options “AutoSpec,” “Edit SpecSheet” and “Choose.” These missing options relate to the use of Excel spreadsheets to perform operations. It is possible to access Excel directly to perform these options. Under Vdraft, the “Run Design” option only allows for the selection of a specification (SPC) file. The SPC file selected can be created from an Excel spreadsheet (XLS) just as in the AutoCAD version.

“Polyline Width” is not present on the main menu. The creation of polyline width controls can be done via the “Dimensions” dialog box. The options to “View,” “Remove” or “Copy” are not offered as dialog options. To view the data, select the polyline and left click, selecting “Properties” from the mini-menu. In the Properties dialog box, select the “Extended Data” tab. To copy a polyline with width control, use the “Edit Dimension,” “Copy Dims” option.

“Convert Synth” appears in the main dialog box instead of the “Edit Dimension” dialog box.

An additional option, “DWG Check,” will check the current drawing for errors.

Dimension Creation

The creation of dimensions is somewhat different when the Vdraft interface is used. In Vdraft, the dimensions are always created with default text (and require editing to change).

When creating horizontal or vertical dimensions for OfficeCAD, you may select the end points as usual, or right click the mouse for a menu of options including “Select Entity.” After selecting the dimension (text) line position, a mini-menu will appear. From this menu select “Done Dimension.” Next you will be prompted for “Depth Point.” You may select a point for the depth, or you can right click and select “Cancel select depth point” which results in the default (infinite) depth. Next you will be prompted for “Text reference point.” Again, you can either select a point, or right click and select “Cancel select text reference point.”

When creating z-tical dimensions, the drawing view is switched automatically to the YZ plane.

For radial or diameter dimensions, it is recommended that you right click and pick "Select Entity" before choosing the arc or circle. Next, select "Done Dimension" from the mini-menu that pops up. You will then be prompted to "Select entity for radial control." This allows you to select additional items to be controlled by that dimension. When selections are complete, right click and pick "Cancel select entities for radial control."

When creating angular dimensions, it is recommended that object snap (accessible from the right click menu) be used to snap to the points of the polyline to be controlled.

Dimension Editing

When editing dimensions, any of the options Edit Layer, Edit Depth, Edit Text, Edit Symmetry, or Activate Dim will allow you to select dimension after dimension to edit. To end, right click and select "Cancel select dimension to edit." When using "Copy Dims," you must first select the items to copy, then right click and select "Start copy" from the mini-menu. You will then be prompted to select a base point and an offset point.

Other Differences between Vdraft and AutoCAD

There are some entity types supported by AutoCAD that Vdraft does not allow you to create. However, Vdraft can load and display AutoCAD drawings that contain these entities. OfficeCAD can correctly process drawings generated from either source.

These entity types include:

- Surfaces generated through the AutoCAD, Draw, Surfaces pulldown method (polymesh type).
- Solids generated through the AutoCAD, Draw, Solids pulldown method. Note that OfficeCAD ignores solids. Solids are not resized and are not relocated.
- Mline.
- Boundary.
- Region.
- Tolerance.
- Hatches.

Master Drawing Creation in Vdraft

The Quad tutorial appears in **Chapter 2**.

Drawing Setup

Master drawing setup is done automatically upon using your first OfficeCAD command. Enter Vdraft and create a new drawing using the default drawing (DWG) template. Select OfficeCAD from the Extensions menu to bring up the OfficeCAD dialog box.

Creating the "Quad" Drawing

Draw a quad to resemble the one in the diagram below. Set SNAP on and draw the box shape with straight, vertical sides, a base that's 90 degrees in relation to the sides, and a tilted top. From OfficeCAD, select "Dimensions", "Horizontal". Pick the two bottom corners of the quad as the end points, and select the text position below the quad. Right click and pick "Cancel select depth point" when depth point is requested. Right click and pick "Cancel select text reference point" for text reference point. From OfficeCAD, change the text to a variable by selecting Edit Dimension, Edit Text. For the text, enter **[W1]** (be sure to enclose variable names in brackets). Now place vertical dimensions on either side of the quad, giving them the texts of **[H1]** and **[H2]** respectively. Next, create an aligned dimension on the top corners of the quad using the standard (not OfficeCAD) drafting tools.

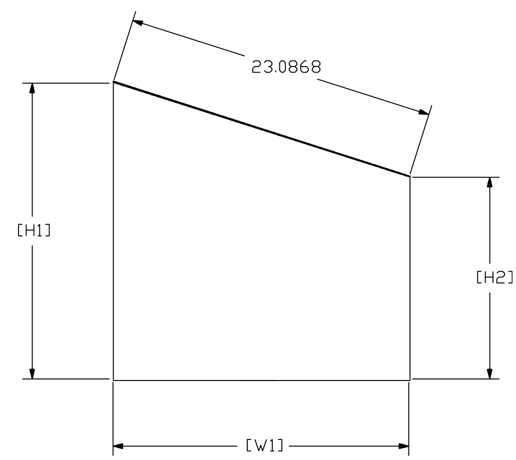


Fig. A-2 QUAD DRAWING

Next, insert a base point from OfficeCAD, Dimensions, "Base Point". In this example, position the base point at the lower left corner of quad. The base point determines the insertion point and center of rotation for the drawing.

Variable Text

Finally, put some text under the figure that will show the correct area of the quad when the parametric design is run. To do this, use Vdraft to create centered text, select the location, and enter the following:

AREA = [A]

The master drawing is now complete. It should resemble the following diagram:

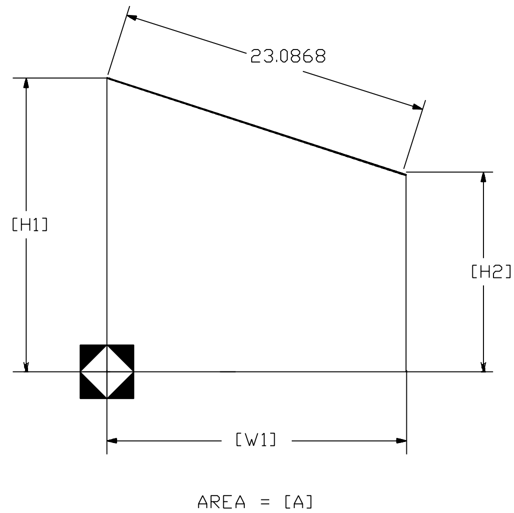


Fig A-3 THE QUAD MASTER DRAWING

The master drawing now contains four variables: [H1], [H2], [W1] and [A]. The first three will be user-defined parameters, the last one will be calculated from the first three. Save your finished quad drawing.

The next section describes the main components of the SpecSheet and shows you how to create a SpecSheet to define the roles of the quad dimensions.

Appendix B – Error Messages

This appendix lists errors that may be encountered when running **OfficeCAD**. Errors are listed by their error number. Each error is followed by a brief explanation and some ideas for fixing the problem.

There are two basic symbols used in the error messages listed here. Your error messages will fill in the variable % as follows:

%d = number
%s = text

General Error Messages

100: Wrong parameters, function terminated.

<spcfile>	Creates DWG file(s) as specified in the SPC file.
-b <spcfile>	Explodes drawing blocks.
-g <spcfile>	Deletes hidden dimension types, such as dimfixhide and dimprohide layers.
-i <spcfile>	Informative mode writes output and master drawing names to Officcad.log file.
-n <spcfile>	Explodes all blocks and moves all entities to layer 0.
-q <spcfile>	Suppresses "Done" message.
-r <dwgfile>	Rectifies a DWG file.
-s <spcfile>	Silent mode. Non-fatal errors do not display on screen.
-t*n* <spcfile>	Sets the tolerance to the numeric value of *n*.

Flags may be combined, such as:
-byt0.01 <spcfile>

101: Program interrupted.
A <CONTROL-C> or other interrupt has been entered.

102: Out of space.
Computer is out of memory.

103: Implausible Internal Error.
You have uncovered a OfficeCAD bug. Please make a note of information provided by the error message, and contact OfficeCAD.

104: Duplicate %s dimension. Endpoint at %s.
Part of your Master Drawing is dimensioned more than once. Re-edit it so as to delete the extra dimension. If you don't see a duplicate dimension you may have one dimension super-imposed on another.

105: Warning: both ends of a %s dimension match. Endpoint at %s.
Your drawing is ambiguously dimensioned. Horizontal or vertical dimensions control the same area twice.

106: Bad dimension type %s.
Your master drawing may be corrupt. You may be able to use AutoCAD or Vdraft to correct the problem. If you can DXFIN and then DXFOUT your master drawing, OfficeCAD should then be able to use it. If that fails, it may be possible to use a word processor to edit out the bad lines, that are referenced in the error message.

107: Bad dimension layer %s in %s - dimension placed on layer %s.
Dimension information should be contained on a layer whose name begins with the characters “DIM”. Delete the bad dimensions from your drawing.

108: Dimension must be in WCS to have depth.
A horizontal or vertical dimension with depth is on a UCS.

109: Negative radial dimension (%s).
Negative radial dimensions are not allowed. You may wish to check your SpecSheet formulas.

110: Angle defined by parallel lines; results unpredictable.
Fixed and moving sides of the dimension may be confused.

111: %s at %s selected by more than one radial dimension.
Your drawing is ambiguously dimensioned. Re-edit your master drawing to delete the extra radial dimension.

112: No active drawing.
The chosen option is valid only when a drawing is open in Vdraft.

File Input/Output Error Messages

113: Can’t open %s.
OfficeCAD cannot find the specified drawing file (DWG) or specification file (SPC). The file name must be a non-number name.

114: Can’t find embedded drawing %s.
Cannot find the block named in the drawing.

115: Unexpected end of file on %s.
If file specified is a SPC file, try running the design session again.

DXF Error Message

116: Tolerance error: DIMASZ and DIMTSZ cannot both be 0
OfficeCAD uses the size of one of these dimension components as an aid in judging what it reads in the rest of the drawing. You must set one or the other of these variables to a reasonable value, so that its size is in keeping with the rest of the drawing. They cannot both be set to zero. For information on how to set these variables please see the AutoCAD or Vdraft manual.

SPC Error Messages

117: More than %d variables in %s
Your design contains more than the allowed number of variables. Try breaking up your design into a multi-SpecSheet design, so as to have additional design files, each with fewer variables.

118: Bad variable at line %d in %s
The specification file (SPC), used to pass information within OfficeCAD, is corrupt. You can re-run the design to produce a new file; or, you can use a word processor to edit the file.

119: Bad rescale value at line %d in %s
The specification file (SPC), used to pass information within OfficeCAD, is corrupt. You can re-run the design to produce a new file; or, you can use a word processor to edit the file.

120: Bad drawing specification line near line %d in %s
The specification file (SPC), used to pass information within OfficeCAD, is corrupt. You can re-run the design to produce a new file; or, you can use a word processor to edit the file.

121: Undefined line type at line %d in %s
The specification file (SPC), used to pass information within OfficeCAD, is corrupt. You can re-run the design to produce a new file; or, you can use a word processor to edit the file.

122: Specification File jumps back and forth between output files too much
The specification file (SPC), used to pass information within OfficeCAD, violates the allowed format. If OUTFILE$ are not grouped, they will jump back and forth too much. To reduce the number of jumps, the file could be sorted. If this error occurs, please make note of the message and contact OfficeCAD.

Polyline Error Messages

123: More than %d %s dimensions affecting a single polyline
You have exceeded the limit of angular or radial dimensions for a single polyline.

124: Consecutive radially-dimensioned arcs in a polyline, arcs join at %s
Consecutive radial dimensions on arcs in polyline is mathematically ambiguous. Separate arcs by straight segment or break polyline.

125: Incompatible angular dimensions on a polyline
Two angular dimensions share a moving side; or, the fixed side of one dimension is the moving side of the second dimension.

126: Angular dimension not matching polyline points, dimension ignored
Angular dimension is not compatible with polyline. Use the OfficeCAD dialog box to remove angular dimension and recreate it.

127: Radial and angular dimensions on consecutive segments in a polyline conflict.
Point of polyline is in contention for control by both a radial and an angular dimension.

128: Due to Previous error, radial dimension on polyline ignored.
Conflicting radial dimension is ignored in the processing of the polyline.

129: Due to previous error, angular dimension on polyline ignored.
Conflicting angular dimension is ignored in the processing of the polyline.

130: Unable to draw tangent, endpoint inside arc radius.
Relocation of a polyline point has placed the endpoint of a tangent line within the circle inscribing the arc. Tangency from that point to the arc is mathematically impossible.

131: Unable to draw cross tangent in polyline, arcs overlap.
Tangency cannot be maintained due to position of arcs.

132: Unable to draw outside tangent in polyline, one arc inside other.
Tangency cannot be maintained due to position of arcs.

Miscellaneous Error Messages

133: Hatching will be removed.
Hatching patterns will be removed from drawing.

134: Hardware Lock not Installed.
The program may require a hardware lock to be properly installed in your parallel port.

135: Entity on User Coordinate System. Unpredictable results possible.
OfficeCAD does not fully support entities on user coordinate systems. They may be relocated incorrectly.

136: Too many control lines on point %s.
Three or more control lines intersect an entities control point. Only the first two will be used to decide the relocation of the control point.

137: Too many control circles on point %s.
Three or more control circles intersect an entities control point. Only the first two will be used to decide the relocation of the control point.

138: Point %s is overdimensioned.
Either two or more control circles and a control line intersect an entities control point, or two or more control lines and a control circle intersect an entities control point. If there are two control circles, the control point will be relocate with respect to them. Otherwise the point will be relocated by the first control line and the control circle.

139: Control circle and control line on %s no longer intersect.
In the master drawing, the control point of an entity was at the intersection of a control line and a control circle. The relocated control line and control circle no longer intersect. The control point will be relocated according to its relationship to the control circle.

140: Control lines intersecting at %s are now parallel.
In the master drawing, the control point of an entity was at the intersection of two control lines. The relocated control lines are now parallel. The control point will be relocated with respect to the first control line.

141: Control circles meeting at point %s are now concentric.
In the master drawing, the control point of an entity was at the intersection of two control circles. The relocated control circles are now concentric. The control point will be relocated with respect to the first control circle.

142: Control circles meeting at point %s no longer intersect.
In the master drawing, the control point of an entity was at the intersection of two control circles. The relocated control circles no longer intersect. The control point will be relocated with respect to the first control circle.

143: Unable to write authorization code to file %s.
OfficeCAD needs to store the authorization code in a file. The directory protection may be preventing this file from being written.

144: Can't replace layer %s with %s. Layer %s is undefined.
The specification file (SPC) contains a layer command to replace a layer with another layer. The definition of the new layer was not found.

145: Insert block information not complete for %s, skipping insert.
The specification file (SPC) contains a block command to insert a named block. The insertion information is incorrect or incomplete.

146: Layer %s not found, will not be modified.
The specification file (SPC) contains a layer command for a layer that it not in the drawing.

147: Unable to insert %s, block not found.
The specification file (SPC) contains a block command to insert a named block. A block by that name could not be found.

148: Failure while creating OfficeCAD dimension or control entity.
Attempt to create an OfficeCAD dimension, control line or control circle has failed.

149: Incomplete parameters to create drawing array.
The specification file (SPC) contains an array command, DWG_CIRC, DWG_RECT or DWG_STEP, with incorrect or incomplete information.

150: Horizontal and Vertical dimensions must be in the XY plane.
Active OfficeCAD horizontal and vertical dimensions must be in WCS, and therefore lie in the XY plane. Different Z elevations are allowed.

151: Z-tical dimensions must be in the YZ plane.
Active OfficeCAD z-tical dimensions must lie in the YZ Plane.

152: Unmatched parentheses in equation: %s.
The master drawing has variable text in the form of an equation. In the equation, the parentheses do not match, such as [3 * (A + B].

153: Unable to evaluate equation: %s.
The master drawing has variable text in the form of an equation. The program is unable to evaluate the equation.

154: Equation contains divide by zero: %s.
The master drawing has variable text in the form of an equation. The equation is attempting to divide a number by zero (possibly from a zero replacement value for a variable).

155: Can't replace value %s in equation %s.
The master drawing has variable text in the form of an equation. A replacement value can not be found for one or more of the variables in the equation.

156: Unable to export type %s.
The specification file (SPC) contains a command to export the file to an unknown type.